DeJuan Anderson May 8, 2014

6.046 Problem 5-3

Collaborators: *Anthony Adams, Michelle Johnson, Kang Zhang* Recitation: Kang Zhang, 12:00

1 State the decision version of this problem and explain why it is NP-complete. You need only describe the problem that you would reduce from and how the reduction works. This should take two sentences. The if and only if part of the proof will be quite obvious and you dont need to write it here.

We will reduce from Partition to Load Balancing.

The reduction is simple; we simply set m = 2. In the case where M = 2, Load Balancing becomes exactly the partition problem.

In other words, Load Balancing is a more general version of the Partition problem.

2 Prove that the simple polynomial time algorithm given is a 2-approximation algorithm for Load Balancing.

For the sake of clarity in explanation, I will assume in this section that the weights for a given task indicate the amount of time the task takes. A more generic example could be formed by taking any instances of "length" and replacing them with "weight".

Let us first define two proxies that will be useful in proving our claim.

First, let p_1 be equal to the length of the longest task. Concretely:

$$p_1 = max(task\ lengths) \tag{1}$$

Next, let p_2 be equal to the average task length. To be concrete:

$$p_2 = \frac{\sum_i^n weight(task_i)}{m} \tag{2}$$

Let our approximation algorithm's result be known as Approx. Likewise, let the optimal solution be known as OPT.

We want to show that Approx $\leq$ OPT.

We can do this in a series of steps.

1) Show that:

$$\begin{aligned} p_1 + p_2 &\leq OPT + OPT \\ &= 2 * OPT \end{aligned} \tag{3}$$

2) Next, define two quantities, x_1 and x_2.

x_1 is the *minimum load on an employee at the time just before the final task is assigned.* That is, x_1 is the amount of time the employee with the least load has worked before the final task is assigned.

x_2 is the amount of time taken after the last task is assigned until completion of all tasks.

These two quantities, summed together, are equal to our approximation's time consumption. In other words,

$$x_1 + x_2 = Approx \tag{4}$$

Given these definitions, we want to show:

$$x_1 + x_2 \leq p_1 + p_2 \tag{5}$$

3) If we show steps 1 and 2, we can leverage these inequalities to get the desired result:

$$x_1 + x_2 = Approx \leq p_1 + p_2 \leq 2 * OPT. \tag{6}$$

2.1 Showing Step 1

Recall that Step 1 was given in the previous section as Equation (3).

In order to show this, let us prove seperately that each proxy is less than the optimal solution, then combine these results.

p_1 is $\leq$ OPT because in the most simple case, there is only one task that must be assigned, and therefore, OPT $= p_1$; the amount of time consumed by the optimal solution is precisely the amount of time it takes to complete the only task.

Now assume there are more tasks. The minimum amount of time that could possibly be taken is in fact equal to that of the longest task; if the other tasks are sufficiently short and/or there are a sufficient number of employees, the other tasks could all be completed before the longest task is done.

In the case where any given worker's tasks take more time than p_1, then OPT $\geq p_1$ by necessity, as OPT requires all tasks to be completed.

Thus, p_1 is $\leq$ OPT.

Now, we examine p_2. Recall that p_2 is average task length and OPT is at least equal to the longest task length.

Therefore, if there are at least two tasks, the average task duration will be at most equal to the longest task length if the tasks are of equal duration. Otherwise, the average will be lower than the longest task length.

In the case of only one task, then again, $p_2 =$ longest task length $=$ OPT.

Therefore, p_2 is $\leq$ OPT.

Since $p_1 \leq OPT$ and $p_2 \leq OPT$, it naturally follows that $p_1 + p_2 \leq OPT + OPT$, and so we have proven that $p_1 + p_2 \leq 2 * OPT$.

2.2 Showing Step 2

Recall that Step 2 is showing the result given in Equation (5). For convenience, Step 2 is given again below:

We want to prove that

$$x_1 + x_2 \leq p_1 + p_2$$

.

We will show this by proving that $x_2 \leq p_1$ and that $x_1 \leq p_2$.

$x_2 \leq p_1$ because if x_2 is the amount of time consumed by the task after the final task is assigned, the absolute worst case would be that we have left the longest task for the final one, in which case we assign the longest task and $x_2 = p_1$.

Otherwise, there will be less time remaining just after the last task is issued, and so $x_2 < p_1$ for these cases.

Thus, we have shown that $x_2 \leq p_1$.

As for proving that $x_1 \leq p_2$, this is more complicated.

Recall that x_1 is the minimum amount of time spent by an employee before the final task is assigned. It is the amount of time that the employee with the shortest tasks has worked.

This must be less than the average task length since the other workloads took equal or greater time, and summing a small number with numbers at least as large can never decrease the average below that of the small number. In fact, the average can either stay equal to the small number or increase above it.

Thus, $x_1 \leq p_2$.

With these results, we have just shown that $x_1 \leq p_2$ and $x_2 \leq p_1$, so naturally, $x_1 + x_2 \leq p_2 + p_1$.

We can now leverage the equalities as desired:

$$\begin{aligned} x_1 + x_2 \leq p_1 + p_2 \leq 2 * OPT \\ x_1 + x_2 = Approx. \quad Sub\ in; \\ Approx \leq p_1 + p_2 \leq 2 * OPT \\ Therefore, \\ Approx \leq 2 * OPT \end{aligned}$$

This completes the proof.