CSC 373 H 1 Y — Summer 2007

University of Toronto — St. George Campus

Lecture Summary for Week 13

This summary is not a replacement for the lecture. If you miss a class, please arrange with a friend to take note for you.

6.3 Bin Packing Problem (Continued)

The First-Fit algorithm for this problem works by going through the items, packing each item into the first avaiable bin and opening new bin if necessary.

For example, suppose that there are 6 items with sizes

$$0.3, 0.8, 0.5, 0.4, 0.2, 0.2$$

then the output of the algorithm is three bins:

$$\{0.3, 0.5, 0.2\}, \quad \{0.8, 0.2\}, \quad \{0.4\}$$

Note that for this example, we need at least 3 bins, since the total size of all items is 2.4.

In spite of its simplicity, we will show that this algorithm has approximation ratio of 2. First, let S be the total size of all items. Then since each bin has size 1, the total number of bins is at least $\lceil S \rceil$, the least integer that is $\geq S$.

Let k be the number of bins output by the First-Fit algorithm. Let OPT be the minimum number of bin required. We need to show that

$$k \leq 2OPT$$

The above observation gives us that $OPT \geq \lceil S \rceil$. So it suffices to show that $k \leq 2\lceil S \rceil$.

Observe that there cannot be 2 bins that are at most half-full (because the second bin would not be opened). So we consider two cases:

Case I: There is no half-full bin. Then the total size of items in every bin is > 0.5. So the total size of all items is $> 0.5k$, i.e.,

$$S > 0.5k, \qquad \text{i.e.,} \qquad k < 2S$$

so $k < 2\lceil S \rceil$.

Case II: There is exactly one bin that is at most half-full. First, if $k = 1$ then k is also the minimum number of bin required, so we are done. Thus suppose that $k \geq 2$.

Without loss of generality, assume that bin 1 is only half-full. Notice that the total size of items in bins 1 and 2 is > 1. Also, the total size of items in bins $3, 4, \ldots, k$ is $> 0.5(k-2)$. Therefore

$$S > 1 + 0.5(k-2) = \frac{1}{2}k$$

Hence $k < 2S \leq 2\lceil S \rceil$. QED.

There are a number of approximation algorithms that use the greedy idea. Best-Fit always packs an item into the bin that can accomodate the item but that has least space left. By the same proof, we can show that Best-Fit also have approximation ratio of 2.

Exercise: Prove that the Best-Fit algorithm has approximation ratio of 2.

The Next-Fit algorithm goes through the items, if an item cannot be packed into the current bin, then the bin is closed permanently and a new bin is opened. For the example above, Next-Fit would output the following packing with 4 bins:

$$\{0.3\}, \quad \{0.8\}, \quad \{0.4, 0.5\}, \quad \{0.2, 0.2\}$$

Again, although it seems wasteful to close a bin permanently when it might still accomodate other items that come after the current item, it can be shown that this algorithm has approximation ratio of 2.

Exercise: Prove that the Next-Fit algorithm has approximation ratio of 2.

The First-Fit approach can be improved by first sorting the items into decreasing order of their sizes. It can be shown that with this modification, the approximation ratio is improved to $\frac{11}{9}$ (the number of bins produced is at most $\frac{11}{9}OPT + \frac{6}{9}$). We will not discuss this improvement.

6.4 Weighted Vertex Cover

This problem is similar to Vertex Cover, but here each vertex is associated with a nonnegative weight, and we want to find a vertex cover with minimum weight.

More precisely, suppose that we are given a graph $G = (V, E)$ where each vertex v has a weight $w(v) \geq 0$. The weight of a subset V' of V is defined to be the total weight of all vertices in V'. The problem is to find a vertex cover of V that has minimum weight.

This problem is a generalization of the Vertex Cover problem; we obtain the latter from the former by setting $w(v) = 1$ for all vertices v.

The simple approximation algorithm that we have for Vertex Cover (Section 6.1) does not have constant ratio for Weighted Vertex Cover. Here we will use Linear Programming to design an approximation algorithm for Weighted Vertex Cover that has approximation ratio of 2.

First, the 0-1 IP formulation of Weighted Vertex Cover is obtained by introducing a variable $x_v \in \{0, 1\}$ for each vertex $v \in V$. The meaning of x_v is that $x_v = 1$ iff v is in the intended minimum weight vertex cover. The objective is to minimize the function

$$\sum_{v \in V} x_v$$

The constraints are

$$x_u + x_v \geq 1 \qquad \text{for each edge } (u, v) \in E$$

Our approximation algorithm works by first "relax" the 0-1 IP to an LP instance. Instead of requiring that $x_v \in \{0, 1\}$ we will add the constraints $0 \leq x_v \leq 1$. The output of an LP algorithm for the LP relaxation might not be a solution to the original IP problem, but we can use it to approximate the Weighted Vertex Cover problem by "rounding" to the nearest integers.

In particular, we consider the LP with the constraints

$$x_u + x_v \geq 1 \qquad \text{for each edge } (u, v) \in E \tag{1}$$

and

$$0 \leq x_v \leq 1 \qquad \text{for each } v \in V$$

The approximation algorithm works as follows:

1. Run a (polytime) LP solver for the above instance;
2. Let $x_v^\star$ be the output of the LP solver;
3. Define a set C as follows:
$$v \in C \text{ iff } x_v^\star \geq 0.5$$
4. Output C.

First we show that the output C is really a vertex cover for G. For this, consider any edge $e = (u, v) \in E$. Then the constraint (1) gaurantees that $x_u^\star + x_v^\star \geq 1$, so at least one of $x_u^\star, x_v^\star$ is ≥ 0.5. Hence at least one of u, v is in C, so C covers e.

Now let OPT be the minimum weight of a vertex cover for G. We will show that

$$w(C) \leq 2OPT$$

where $w(C)$ is the total weight of all vertices in C.

First, since OPT the minimum value of the objective function for the 0-1 IP, and

$$\sum_{v \in V} w(v) \times x_v^\star$$

is the minimum value of the objective function for the LP, we have

$$\sum_{v \in V} w(v) \times x_v^\star \leq OPT$$

(because all solutions to the 0-1 IP constraints are also solutions to the LP constraints).

We have

$$\begin{aligned}
& w(C) = \sum_{v \in C} w(v) \\
\text{so } & w(C) \leq \sum_{v \in C} w(v) \times 2x_v^\star \quad \text{(because } x_v^\star \geq 0.5 \text{ for } v \in C\text{)} \\
\text{so } & w(C) \leq \sum_{v \in V} w(v) \times 2x_v^\star \\
\text{so } & w(C) \leq 2OPT \quad \text{(by the above observation)}
\end{aligned}$$

We have just proved that the above algorithm for Weighted Vertex Cover has approximation ratio of 2.