Combinatorial Optimization – Problem Set 12 – Solutions

You can hand in one of the following problems at the start of Tuesday's problem session. Please explain your solution carefully. Don't forget to put your name.

Bin Packing

1. *Show that Next Fit is not a k-approximation algorithm for any $k < 2$.*
   Find an example for which First Fit (without sorting) has an approximation factor 5/3.

   We can force Next Fit to be very inefficient by making it leave bins mostly unfilled. A sequence like the following would do this, for some small $\varepsilon > 0$:

   $$2\varepsilon, 1-\varepsilon, 2\varepsilon, 1-\varepsilon, 2\varepsilon.$$

   Next Fit would give each number its own bin, because no two consecutive bins fit together. But there is a more efficient packing (which First Fit would find): give each $1-\varepsilon$ its own bin, and put all the 2ε together in one bin. For this we have to make ε small enough; for example, for the sequence above, $\varepsilon = 1/6$ would do. Then in this case we would have an approximation factor of 5/3.
   To generalize this, we take a sequence like above with m times 2ε, and we set $\varepsilon = 1/2m$. Then Next Fit will use $2m-1$ bins (m for the 2ε, $m-1$ for the $1-\varepsilon$), but the best packing uses m bins (1 for the 2ε, $m-1$ for the $1-\varepsilon$). So the approximation factor is $2-1/m$, which proves that no approximation factor < 2 is possible.

   This example does what is required (for bins of capacity 24):

   $$7, 7, 7, 4, 4, 4, 13, 13, 13.$$

   First Fit will need 5 bins, but 3 bins is possible.
   Here's how you might come up with a small example like this. After some playing around you could guess at a "symmetric solution", with 3×3 items of sizes α, β, γ filling 3 bins exactly, so $\alpha + \beta + \gamma = 1$. The way to make First Fit do badly is to have $\gamma = 1/2 + \epsilon$, so that the γ items will take up the last 3 bins of 5, and so that the 3 α items end up in the first bin, the 3 β items in the second.
   Now you can deduce what they should be. The two constraints are that a β does not fit in with the 3 α's, and that a γ does not fit in with the 3 β's. So

   $$3\alpha + \beta > 1, \quad 3\beta + \gamma > 1.$$

   The second together with $\gamma = 1/2 + \epsilon$ gives $\beta > 1/6 - \epsilon/3$. We take

   $$\beta = \frac{1}{6} - \epsilon/3 + \delta, \quad \text{so} \quad \alpha = 1 - \beta - \gamma = \frac{1}{3} - \frac{2}{3}\epsilon - \delta.$$

   Then $3\alpha + \beta > 1$ gives $14\epsilon + 12\delta < 1$. Any pick of small ϵ, δ that satisfies this will work, but to get nice integers we pick $\epsilon = 1/24, \delta = 1/72$ (found by playing with the divisibility a little). Then $24\alpha = 7, 24\beta = 4, 24\gamma = 13$.

2. *Show that Next Fit is a 2-approximation algorithm for* Bin Packing.

   Let k be the number of bins used by Next Fit, and k^* the minimal number of bins, so we want to show $k \leq 2k^*$. Write $S = \sum_{i \in I} s_i$, so $k^* \geq S$.
   We use the fact that Next Fit ensures that any two adjacent used bins together contain items of size larger than 1, since otherwise the items in the second bin would have been place in the first. So for every odd j, except maybe the last, we have

$$\sum_{i:a(i)=j \text{ or } j+1} s_i > 1.$$

   We have $\lfloor k/2 \rfloor$ such inequalities. Summing all of them gives $S > \lfloor k/2 \rfloor$, and since these are integers we get

$$S \geq \lfloor k/2 \rfloor + 1 \geq \frac{k-1}{2} + 1,$$

   which implies

$$k \leq 2S - 1 \leq 2S \leq 2k^*.$$

   Alternative proof: Take the first element of bins 2 to k, and add that element to the bin before it. That will give $k-1$ overfilled bins. Hence we have

$$k - 1 < S + \sum (\text{first element of each bin}) \leq 2S \leq 2k^*.$$

   Since these are integers, $k - 1 < 2k^*$ implies $k \leq 2k^*$

3. *We call an algorithm for Bin Packing monotonic if the number of bins it uses for packing a list of items is always larger than for any sublist.*
   Show that Next Fit is monotonic, but First Fit is not.

   If Next Fit were not monotonic, then there would be an instance I such that removing one of its items would lead to more bins. It's pretty easy to see that this is not possible. Let i be the item removed. The items $i' < i$ remain in the same bin, while the items $i'' > i$ can only be moved to an earlier bin. So clearly the number of bins used can only become less.

   The following example (with bin size 10) shows that First Fit is not monotonic:

$$4, 6, 4, 1, 6, 1, 4, 4.$$

   First Fit uses 3 bins, but if we remove the first 1, it will use 4 bins.

4. *Show that instances of* Bin Packing *for which $s_i > 1/3$ for all item sizes s_i can be solved exactly using a polynomial algorithm that we saw earlier in the course.*
Show that this is not that useful, because FFD solves these instances exactly as well.

   Construct a graph with the items for vertices, and an edge between items i and j if $s_i + s_j < 1$. Then a matching in this graph will correspond to a bin packing, by putting any two matched items into a bin together. Because of the minimum size $1/3$, no bin can contain more than 3 items, so every bin packing corresponds to a matching. The number of bins equals the number of matching edges plus the number of unmatched vertices. This number is minimized when the matching has maximum cardinality, so we can use the blossom algorithm for maximum cardinality matchings.

   The first part is not so useful because FFD is much faster.
   Let the *large* items be those with $s_i > 1/2$ and the *medium* items $1/3 < s_i \leq 1/2$.
   In any packing, each large item will get its own bin, some of the medium items will be added to a bin with a large item, and the remaining medium items are either paired off, or get their own bins. In a minimum packing, these remaining medium items will all be paired off arbitrarily, except possibly a single one. This is exactly what FFD will do, and the number of bins used only depends on the number of remaining medium items. It is not hard to see that FFD will maximize the number of the large-medium pairs, so it will minimize the number of remaining medium items, hence also the total number of bins.