Name: Grade:

Quick Test: Approximation Algorithms

Florent Bouchez Tichadou — Gwenaël Delaval

November 20th, 2015

Duration: 30 minutes.
All documents forbidden.

Exercise 1 (Bin packing problem)

We consider the bin packing problem, where we try to put items in bins while minimizing the total number of bins used. A bin is a container of size 1, and we have a set of n items $a_1, \ldots, a_n$ of different sizes $s_1, \ldots, s_n$. The sum of the sizes of the items in a container cannot be more than 1.
A solution to the problem is a partition of the items $\{B_1, B_2, \ldots, B_k\}$, with the following three constraints:

$$\forall j, \sum_{a_i \in B_j} s_i \leqslant 1 \qquad \forall i, \exists j \text{ such that } a_i \in B_j \qquad k \text{ is minimal}$$

Question 1.1 We can suppose that $\forall i \in 1, \ldots, n,\ 0 < s_i \leqslant 1$. Explain why.

Solution to 1.1: A negative size has no sense, and a size zero means we can ignore the item (we can always put it in a bin), so $s_i > 0$.
$s_i \leqslant 1$ or there is **no solution**.
Note: It is possible that items have size $s_i > 1$, but in that case there is no solution no the problem and it is very easy to detect it, so we only consider instances where $s_i \leqslant 1$ for all i.

Next Fit

We first consider the *next fit* algorithm whose principle is the following: we have a currently opened bin, and consider items one by one, trying to fit them in the current bin. If the current item does not fit in the current bin, we close it **definitively** and open a new empty bin to put the item in; This new bin is now the current bin.

Question 1.2 Give a pseudo-code for the next fit algorithm.

Solution to 1.2:

```
1  k ← 1;
2  P ← ∅;
3  B ← ∅;
4  s ← 0;
5  foreach 1 ⩽ i ⩽ n do
6      if s + s_i > 1 then
7          P ← P ∪ B;
8          B ← ∅;
9          s ← 0;
10     B ← B ∪ a_i;
11     s ← s − s_i;
12 return P;
```

Question 1.3 Suppose you have the following items:

a_i	a_1	a_2	a_3	a_4	a_5
s_i	0.7	0.8	0.1	0.2	0.1

Show that the next fit algorithm cannot be an α-approximation, with $\alpha < 3/2$.

Solution to 1.3: Next fit solution : $\{0.7\}, \{0.8, 0.1\}$ and $\{0.2, 0.1\}$, so $C = 3$.
Optimal solution : $\{0.7, 0.2, 0.1\}, \{0.8, 0.1\}$, so $C^* = 2$.
Hence we have an example where $\frac{C}{C^*} = 3/2$, so the next fit cannot be an α-approximation with $\alpha < 3/2$, otherwise, our example would be a counter example.

We want to prove the next fit algorithm is a 2-approximation.

Question 1.4 Suppose the items could be split to that we can put them in multiple bins. Give the formula that would provide the optimal number of bins required in that case. Deduce a lower bound for the optimal of original problem.

Solution to 1.4: If we can split items, it is always possible to completely fill a bin before closing it, so in the next fit solution all bins are full but for the last. The cost of this solution is $L = \lceil \sum_i s_i \rceil$.
L is clearly a lower bound to the original problem as it is not possible to be "more compact" than L.

Question 1.5 Consider successive bins by pairs in the next fit solution (i.e., bins B_{2j} and B_{2j+1}), what can you say about the empty space in the bins of a pair ?

Solution to 1.5: The empty space is < 1, otherwise, next fit would have managed to put everything in only one bin.

Question 1.6 Prove that the next fit algorithm is a 2-approximation.

Solution to 1.6: If the next fit solution has an even number of bins, then $C = 2k$. From the previous question, each pair of bins is at least half-full, so $\sum_i s_i > k$. This means $C = 2k < 2L < 2C^*$.

For an odd number of bins, $C = 2k + 1$. We know the first $2k$ are at most half empty. So, at best, those $2k$ bins would fit in k bins, all completely full. Since the last bin is not empty, optimal requires one more bin: $C^* \geqslant k + 1$, hence $C = 2k < 2C^*$.

We shown that for all cases, $C < 2C^*$, so next fit is a 2-approximation.
Note: Actually, we see that the result is a bit stronger. In the even case also, $C^* \geqslant k + 1$, otherwise it would mean that in next fit the bin are on average *exactly* half empty, which is impossible as this would mean there exists two bins B_{2i} and B_{2i+1} whose sum is $\leqslant 1$.
So we have the stronger result $C < 2C^* - 1$.

We now want to prove that the analysis is tight, i.e., that the next fit algorithm is not an α-approximation for any $\alpha < 2$.

Question 1.7 Find a family of examples where next fit gives a solution as close as you want from a factor 2 of the optimal number of bins.

Solution to 1.7: Note: We want a solution that maximizes the amount of space lost, i.e., something like ϵ, 1, ϵ, 1, $\epsilon \ldots$

I_n, consisting of $2n$ items of size $1/n, 1, 1/n, 1, \ldots$ (repeated n times). As n increases, the amount of space left empty in the bins gets close to $\frac{1}{2}$ of the total space.

Note: The optimal is putting all $1/n$ items in one bin, plus n bins for the rest, so $C^* = n + 1$. The next fit will put one item per bin, so $C = 2n$. By increasing n, the factor $C/C^* = 2n/n+1$ can get as close to 2 as we want. So for any $\alpha < 2$, there is an n such that I_n is a counter example.

First Fit (bonus section)

We can see that the *next fit* algorithm is not very efficient because only one bin is open at a time. To avoid such problems we now consider the *first fit* algorithm where all non-full bins can be considered for storage, i.e., when adding item a_i, we put it in the first bin that has sufficient space.

Question 1.8 What is the worst possible example you can find for *first fit*? Can you conjecture on the approximation ratio?

Solution to 1.8: For instance 0.6, 0.7, 0.3, 0.4.
Optimal is 2 but first fit will group 0.3 with 0.6, and then need another bin for 0.4.
So we are sure first fit is not better than a $3/2$-approximation.

Note: It is actually a $17/10$-approximation, but it is a bit hard to prove...