3.3 The bin-packing problem

In the *bin-packing problem*, we are given n pieces (or items) with specified sizes $a_1, a_2, \ldots, a_n$, such that

$$1 > a_1 \geq a_2 \geq \cdots \geq a_n > 0;$$

we wish to pack the pieces into bins, where each bin can hold any subset of pieces of total size at most 1, so as to minimize the number of bins used.

The bin-packing problem is related to a decision problem called the *partition problem*. In the partition problem, we are given n positive integers $b_1, \ldots, b_n$ whose sum $B = \sum_{i=1}^{n} b_i$ is even, and we wish to know if we can partition the set of indices $\{1, \ldots, n\}$ into sets S and T such that $\sum_{i \in S} b_i = \sum_{i \in T} b_i$. The partition problem is well known to be NP-complete. Notice that we can reduce this problem to a bin-packing problem by setting $a_i = 2b_i/B$ and checking whether we can pack all the pieces into two bins or not. This gives the following theorem.

Theorem 3.8: *Unless* P = NP*, there cannot exist a ρ-approximation algorithm for the bin-packing problem for any $\rho < 3/2$.*

However, consider the First-Fit-Decreasing algorithm, where the pieces are packed in order of non-increasing size, and the next piece is always packed into the first bin in which it fits; that is, we first open bin 1, and we start bin $k+1$ only when the current piece does not fit into any of the bins $1, \ldots, k$. If $\mathrm{FFD}(I)$ denotes the number of bins used by this algorithm on input I, and $\mathrm{OPT}(I)$ denotes the number of bins used in the optimal packing, then a celebrated classic result shows that that $\mathrm{FFD}(I) \leq (11/9)\,\mathrm{OPT}(I) + 4$ for any input I.

Thus, significantly stronger results can be obtained by relaxing the notion of the performance guarantee to allow for small additive terms. In fact, it is completely consistent with our current understanding of complexity theory that there is an algorithm that always produces a packing with at most $\mathrm{OPT}(I) + 1$ bins.

Why is it that we have bothered to mention hardness results of the form "there does not exist a ρ-approximation algorithm unless P = NP" if such a result can be so easily circumvented? The reason is that for all of the weighted problems that we have discussed, any distinction between the two types of guarantees disappears; any algorithm guaranteed to produce a solution of value at most $\rho\,\mathrm{OPT} + c$ can be converted to a ρ-approximation algorithm. Each of these problems has a natural *rescaling property*: for any input I and any value κ, we can construct an essentially identical instance I' such that the objective function value of any feasible solution is rescaled by κ. For example, for the scheduling problem of Section 3.2, if one simply multiplies each processing time p_j by κ, one can accomplish this rescaling, or for a combinatorial problem such as the unweighted vertex cover problem, one can consider an input with κ disjoint copies of the original input graph. Such rescaling makes it possible to blunt the effect of any small additive term $c < \kappa$ in the guarantee, and make it effectively 0. Observe that the bin-packing problem does not have this rescaling property; there is no obvious way to "multiply" the instance in a way that does not blur the combinatorial structure of the original input. (Think about what happens if you construct a new input that contains two copies of each piece of I!) Thus, whenever we consider designing approximation algorithms for a new combinatorial optimization problem, it is important to consider first whether the problem does have the rescaling property, since that will indicate what sort of performance guarantee one might hope for.

Although we will not prove the performance guarantee for the First-Fit-Decreasing algorithm for the bin-packing problem, we shall show that an exceedingly simple algorithm does perform reasonably well. Consider the First-Fit algorithm, which works exactly as First-Fit-Decreasing, except that we don't first sort the pieces in non-increasing size order. We can analyze its