MPRI
TD 2

Stéphane Vialette

October 9, 2016

1 Nussinov

For a secondary structure S

- the number of *base pairs stackings* is

$$\mathrm{BPS}(S) = |\{(i,j) \in S : (i+1, j-1) \in S\}|, \text{ and}$$

- the number of *stacking base pairs* is

$$\mathrm{SBP}(S) = |\{(i,j) \in S : (i+1, j-1) \in S \text{ or } (i-1, j+1) \in S\}|.$$

- **Design an algorithm for finding a pseudo-knot-free structure that maximizes $\mathrm{BPS}(S)$.**
- **Design an algorithm for finding a pseudo-knot-free structure that maximizes $\mathrm{SBP}(S)$.**

2 Bin packing

Here we consider the classical Bin Packing problem: Given a set of bins $\{S_1, S_2 \ldots\}$ with the same size V and a list of n items with sizes $a_1, a_2, \ldots, a_n$ to pack, find an integer number of bins B and a B-partition $S_1 \cup S_2 \cup \ldots \cup S_B$ of the set $\{1, 2, \ldots, n\}$ such that $\sum_{i \in S_k} a_i \leq V$ for all $k = 1, 2, \ldots, B$. A solution is optimal if it has minimal B. The B-value for an optimal solution is denoted **OPT**.

- **Prove that the Bin Packing problem is NP-complete even for 2 bins.**
- **Prove that there is no ρ-approximation algorithm with $\rho < 3/2$ for Bin Packing unless P $=$ NP.**

An *approximation algorithm* is called a $f(n)$-approximation algorithm for input size n if it can be proven that the solution that the algorithm finds is at most a multiplicative factor of $f(n)$ times worse than the optimal solution. Here, $f(n)$ is called the *approximation ratio*. Problems in APX are those with algorithms for which the approximation ratio $f(n)$ is a constant c. The approximation ratio is conventionally stated greater than 1. In the case of minimization problems, $f(n)$ is the found solution's score divided by the optimum solution's score, while for maximization problems the reverse is the case. For maximization problems, where an inferior solution has a smaller score, $f(n)$ is sometimes stated as less than 1; in such cases, the reciprocal of $f(n)$ is the ratio of the score of the found solution to the score of the optimum solution.

Consider the *first-fit algorithm* for the Bin Packing problem: For each item, it attempts to place the item in the first bin that can accommodate the item. If no bin is found, it opens a new bin and puts the item within the new bin.

- **Prove the number of bins used by the first-fit algorithm is no more than twice the optimal number of bins. In toher words, the first-fit algorithm achieves an approximation factor of 2.**

3 Linear graphs

- **Let $G = (V, E)$ be a linear graph. Prove that finding a $\{<, \between\}$-structured perfect matching in G is NP-complete.**