Institut für Informatik
Prof. Dr. Heiko Röglin
Clemens Rösner

universitätbonn

Probabilistic Analysis
of Algorithms
Summer 2015

Problem Set 2

Problem 1

Let n points be placed uniformly at random on the boundary of a circle of circumference 1. These n points divide the circle into n arcs.

(a) What is the average arc length?

(b) Let x denote an arbitrary fixed point on the circle. What is the expected length of the arc that contains the point x?

Problem 2

Find an algorithm for the knapsack problem that runs in the worst case in time $O(nP)$, where n is the number of items, all profits $p_1, \dots, p_n \in \mathbb{N}$ are natural numbers, and $P := \sum_{i=1}^{n} p_i$. Why does the existence of such an algorithm not prove $P = NP$?

Problem 3

For an instance of the knapsack problem with profits $p \in \mathbb{R}^n_{\geq 0}$, weights $w \in \mathbb{R}^n_{\geq 0}$, and capacity $W \in \mathbb{R}$, we define the *winner gap* Δ to be the difference in profit between the best solution x^* and the second best solution x^{**}. Formally, let $\Delta := p^\mathsf{T} x^* - p^\mathsf{T} x^{**}$, where

$$x^* := \arg\max\{p^\mathsf{T} x \mid x \in \{0,1\}^n \text{ and } w^\mathsf{T} x \leq W\}$$

$$x^{**} := \arg\max\{p^\mathsf{T} x \mid x \in \{0,1\}^n \text{ and } w^\mathsf{T} x \leq W \text{ and } x \neq x^*\}.$$

We assume that there are at least two feasible solutions. Then Δ is well-defined. Let the weights be arbitrary and let the profits be ϕ-perturbed numbers from $[0,1]$, i.e., each profit p_i is chosen independently according some probability density $f_i : [0,1] \to [0,\phi]$ for some fixed $\phi \geq 1$. Show that for any $\epsilon > 0$

$$\Pr[\Delta \leq \varepsilon] \leq n\phi\varepsilon.$$

Problem 4

Give an implementation of the Nemhauser-Ullmann algorithm in Java or C++ with running time $O(\sum_{i=0}^{n-1} |\mathcal{P}_i|)$, where n denotes the number of items and $\mathcal{P}_i$ denotes the Pareto set of the restricted instance that consists only of the first i items.

Use your implementation to generate the Pareto set of instances in which all profits and weights are chosen uniformly at random from $[0,1]$ for $n = 10, 20, 30, \dots$. How does the number of Pareto-optimal solutions and the running time depend on n in your experiments.