The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 8, Problem 8.27

The perfect illustration of this theorem is *Knapsack*, where the feasibility of a solution is determined by the set of sizes and the bag size, the cost of a solution is determined by the set of weights (values), the optimal cost of any instance is easily bounded above by the product of the number of items by the value of the most valuable item, and a pseudo-polynomial algorithm exists that runs in time bounded by a polynomial in the number of items and the cost of the optimal solution. We used all these facts in deriving a fully polynomial-time approximation scheme for *Knapsack*, essentially by truncating the numbers representing the weights, solving the resulting problem exactly using the pseudo-polynomial algorithm, and showing that the resulting solution's cost in the original problem could be made arbitrarily close (in ratio) to the optimal by appropriate choice of truncation width.

Assume a maximization problem (minimization is the obvious symmetric) and assume that the cost function is non-negative when the q_is are non-negative. (This rather reasonable assumption in fact implies that the cost function is monotone non-decreasing.) Also assume that the optimal cost is no less than the largest cost-setting parameter, q. Let t be the truncation length; how to set it will soon become apparent. Then let $\hat{q}_i = \lfloor q_i/2^t \rfloor$; we have

$$\frac{q_i}{2^t} \geq \hat{q}_i \geq \frac{q_i}{2^t} - 1.$$

Let f^* be the optimal solution for the problem and let $\hat{f}$ be the optimal solution for the truncated problem. We get

$$c(q_i, f^*) \geq c(q_i, \hat{f}) \geq 2^t \cdot c(\hat{q}_i, \hat{f}) \geq 2^t \cdot c(\hat{q}_i, f^*) \geq c(q_i, f^*) - 2^t \cdot c(1, f^*).$$

The first inequality is due to the optimality of f^* in the original problem; the second derives from the cost function's monotonicity and linearity; the third is due to the optimality of $\hat{f}$ in the truncated problem; the last again comes from the monotonicity and linearity of the cost function. An immediate consequence is that

$$c(q_i, f^*) - c(q_i, \hat{f}) \leq 2^t \cdot c(1, f^*),$$

so that the performance ratio of our algorithm is bounded as follows:

$$\frac{c^* - \hat{c}}{c^*} \leq \frac{2^t \cdot c(1, f^*)}{c^*}.$$

But, by assumption, $c^* \geq q$ and $c(1, f^*) \leq p_1(|I|, 1)$ for some polynomial p_1, so that we get

$$\frac{c^* - \hat{c}}{c^*} \leq \frac{2^t \cdot p_1(|I|, 1)}{q},$$

and this last bound is the desired ε. Thus we set the truncation length to

$$t = \left\lfloor \log \frac{q \cdot \varepsilon}{p_1(|I|, 1)} \right\rfloor$$

Now our pseudo-polynomial algorithm runs in time $p_3(|I|, q)$, so that, on the truncated problem, it will run in time $p_3(|I|, \hat{q})$, that is, $p_3(|I|, \frac{p_1(|I|,1)}{\varepsilon})$; since $p_1(|I|, 1)$ is polynomial in $|I|$, our algorithm runs in time polynomial in $|I|$ and $1/\varepsilon$, as desired, and we are done.