FPTAS and pseudopoly time exact algorithm

For an instance I, let I_u denote an encoding of I where all the numbers are written in unary.

$5 \to 11111$ unary encoding

Pseudopolynomial time algorithm: runs in time $poly(|I_u|)$.

Theorem 1: Let $p()$ be a polynomial & let Π be a minimization problem such that OBJ_Π is integer-valued and on any instance I, $OBJ_\Pi(I) \le p(|I_u|)$. If Π admits a FPTAS, then Π also admits a pseudopolynomial time exact algorithm.

Proof: Suppose there is an FPTAS for Π which runs in time $q(|I|, \frac{1}{\varepsilon})$ on an instance I (q is polynomial), and returns a sol^n S s.t.

$$OBJ(S) \le (1+\varepsilon) \cdot OPT.$$

$$\varepsilon := \frac{1}{2p(|I_u|)}.$$

Runtime: $q(|I|, p(|I_u|)) \Rightarrow$ Pseudopolynomial time algorithm.

$$
\begin{aligned}
OBJ(S) &\le (1+\varepsilon)\cdot OPT \\
&\le OPT + \tfrac{1}{2} \quad \left[\because\ OPT \le p(|I_u|)\ \&\ \varepsilon = \frac{1}{2p(|I_u|)}\right]
\end{aligned}
$$

$$\Rightarrow OBJ(S) = OPT.$$

$\blacksquare$

<u>Definition</u>: A problem Π is strongly NP-hard if every problem in NP can be polynomially reduced to Π such that the numbers in the reduced instance are written in unary.

<u>Observation I</u>: If Π is strongly NP-hard and $P \neq NP$ then Π does not have a pseudopolynomial time algorithm.

<u>Proof</u>:

SAT

$$|I_u| \le \text{poly}(|\phi|)$$

$$\phi \xrightarrow[\text{poly-time}]{\text{reduction}} I_u \longrightarrow A \longrightarrow$$

(runs in time $\text{poly}(|I_u|) \le \text{poly}(|\phi|)$)

$\blacksquare$

Corollary: If Π is a strongly NP-hard minimization problem satisfying the conditions of Theorem 1, and $P \neq NP$, then Π does not have any FPTAS.

Bin Packing

Input: n items with sizes $a_1, \cdots, a_n \in (0, 1]$.

Task: Find a packing in unit-sized bins that minimizes the number of bins used.

$$\underbrace{\tfrac{1}{2}, \tfrac{1}{3}}_{1}, \underbrace{\tfrac{3}{5}}_{1}, \underbrace{\tfrac{4}{5}}_{1}$$

Algorithm (First-fit):

- Consider items in an arbitrary order. Let the sizes be $a_1, \ldots, a_n$ in the order considered
- for $i = 1$ to n
  - if an existing bin B has residual capacity at least a_i, put i in B. Otherwise, open a new bin and put i in it.

Claim: First-Fit is a two approximation algorithm.

Proof: Let First-Fit use m bins on an instance.

Claim: At least $m-1$ bins are at least half full.

Proof: Towards a contradiction, assume that there are two bins B_1 & B_2 which are less than half-full. Assume that B_2 was created after B_1. The size of the object that led to the creation of B_2 is less than $\frac{1}{2}$. But B_1 had residual capacity more than $\frac{1}{2}$. This is a contradiction.

$$\therefore \sum_{i=1}^{n} a_i > \frac{1}{2}(m-1).$$

$$OPT \geq \sum_{i=1}^{n} a_i$$

$$OPT \geq \frac{1}{2}(m-1) \Rightarrow m-1 < 2 \cdot OPT$$

$$\Rightarrow m \leq 2 \cdot OPT.$$

Theorem: $\forall\, \varepsilon > 0$, there is no approximation algorithm having a guarantee of $\frac{3}{2} - \varepsilon$ for the bin packing problem, assuming $P \neq NP$.

Proof: Subset-Sum: Given $a_1, \dots, a_n \in \mathbb{R}^{>0}$, does there exist a set $S \subseteq \{1, \dots, n\}$ s.t. $\sum_{i \in S} a_i = \sum_{i \notin S} a_i = \frac{1}{2} \underbrace{\sum_{i=1}^{n} a_i}_{K}$

Let $I = \left(\frac{a_1}{K}, \frac{a_2}{K}, \dots, \frac{a_n}{K} \right)$

$$\sum_{i=1}^{n} \frac{a_i}{K} = 2.$$

The answer to the subset-sum instance is "yes" if & only if the instance I can be packed using two unit-sized bins.

Towards a contradiction, assume that there exists a poly-time $(\frac{3}{2} - \varepsilon)$-factor approximation algorithm A for bin packing. Consider the following algorithm for subset-sum: Input: $(a_1, \dots, a_n)$. $K = \frac{1}{2} \sum_{i=1}^{n} a_i$.

- Run $\mathcal{A}$ on $\left(\frac{a_1}{K}, \ldots, \frac{a_n}{K}\right)$

- If the number of bins in the computed sol^n is 2, return "yes". Otherwise return "no".

<u>Correctness</u>: Assume that $(a_1, \ldots, a_n)$ is a 'yes' instance. Then $\left(\frac{a_1}{K}, \ldots, \frac{a_n}{K}\right)$ can be packed using 2 bins. Then, the no. of bins of the computed sol^n is at most $\left(\frac{3}{2} - \varepsilon\right) \cdot 2 = 3 - 2\varepsilon$. Since, no. of bins is integral, it must equal 2. The algo. returns "yes".

If $(a_1, \ldots, a_n)$ is a 'no' instance, then $\left(\frac{a_1}{K}, \ldots, \frac{a_n}{K}\right)$ can only be packed using 3 or more bins. So, algo returns "no".

Thus we have a polytime algo. for subset sum which contradicts $P \neq NP$.