Approximationsalgorithmen

Lecture 9: Approximation Schemes and the Knapsack Problem

Joachim Spoerhase       Alexander Wolff

(Steven Chaplick)

[Vazirani : Chapter 8]
[Williamson & Shmoys : Chapter 3.1]
Approximation Scheme

Let Π be an optimization problem. An algorithm A is called polynomial-time approximation scheme (PTAS), if it outputs for every input \((I, \epsilon)\) with \(I \in D_\Pi\) and \(\epsilon > 0\) a solution \(s \in S_\Pi(I)\) such that the following holds:

- \(\text{obj}_\Pi(I, s) \leq (1 + \epsilon) \cdot \text{OPT}\), if Π min problem,
- \(\text{obj}_\Pi(I, s) \geq (1 - \epsilon) \cdot \text{OPT}\), if Π max problem

The runtime of A is polynomial in \(|I|\) for every fixed \(\epsilon > 0\).

A is called fully polynomial-time approximation scheme (FPTAS), if its running time is polynomial in \(|I|\) and \(1/\epsilon\).

Example running times

- \(O(n^{1/\epsilon}) \Rightarrow\) PTAS
- \(O(2^{1/\epsilon} n^4) \Rightarrow\) PTAS
- \(O((1/\epsilon)^2 n^3) \Rightarrow\) FPTAS
Knapsack Problem

We are given a set $S = \{a_1, \ldots, a_n\}$ of Objects. For each object $a_i$, $i = 1, \ldots, n$ a size $\text{size}(a_i) \in \mathbb{N}^+$ and a profit $\text{profit}(a_i) \in \mathbb{N}^+$ is specified. Moreover, we are given a (knapsack) capacity $B \in \mathbb{N}^+$. We are looking for a subset of the objects whose total size is at most $B$ and whose total profit is maximum.

NP-hard
Pseudopolynomial-Time Algorithm

Let Π be an optimization problem whose instances can be represented by objects (such as sets, elements, edges, nodes) and numbers (such as costs, weights, profits). As usual, $|I|$ denotes the size of the instance $I \in D_\Pi$, where all numbers in $I$ are encoded in binary. We use $|I_u|$ to denote the size of $I$ when all numbers in $I$ are encoded in unary.

- The running time of a polynomial-time algorithm for Π is polynomial in $|I|$.

- The running time of a pseudo-polynomial-time algorithm is polynomial in $|I_u|$.

- The running time of a pseudo-polynomial algorithm may not be polynomial in $|I|$.
Pseudo-Polynomial Alg. for Knapsack

- \( P := \max_i \text{profit}(a_i) \Rightarrow \text{OPT} \leq nP \)

- For every \( i = 1, \ldots, n \) and every \( p \in \{1, \ldots, nP\} \) let \( S_{i,p} \) be a subset of \( \{a_1, \ldots, a_i\} \) whose total profit is precisely \( p \) and whose total size is minimum among all subsets with these properties. Such a set may not exist.

- \( A(i, p) \) denotes the total size of \( S_{i,p} \) (set \( A(i, p) = \infty \) if no such set exists).

- If all \( A(i, p) \) are known we can compute \( \text{OPT} \) by
  \[
  \max\{ p \mid A(n, p) \leq B \} 
  \]
Pseudo-Polynomial Alg. for **Knapsack**

- $A(1, p)$ can easily be determined for $p \in \{0, \ldots, nP\}$

- set $A(i, p) := \infty$ for $p < 0$

- $A(i + 1, p) = \min\{A(i, p), \text{size}(a_{i+1}) + A(i, p - \text{profit}(a_{i+1}))\}$

- All values $A(i, p)$ and thus OPT can be determined in $O(n^2 P)$ time

**Knapsack** can be solved to optimality in $O(n^2 P)$ time.
FPTAS for Knapsack by Scaling

Note: $O(n^2 P)$ is polynomial in $n$, when $P$ is polynomial in $n$

- FPTAS-Idea: **Scale** profits to polynomial size (as required by the error parameter $\epsilon$).
FPTAS for Knapsack by Scaling

KnapsackFPTAS($I, \varepsilon$)

\[ K \leftarrow \frac{\varepsilon P}{n}, \quad \text{profit}'(a_i) := \left\lfloor \frac{\text{profit}(a_i)}{K} \right\rfloor \]

compute optimum solution $S'$ for $I$ w.r.t. profit'($\cdot$)

return $S'$

Lemma The solution $S'$ satisfies $\text{profit}(S') \geq (1 - \varepsilon) \cdot \text{OPT}$.

Proof: Let $o_1, \ldots, o_k$ be an optimal solution.

Obs 1: For each $o_i$, $K \cdot \text{profit}'(o_i) \in [\text{profit}(o_i) - K, \text{profit}(o_i)]$

\[ \implies \text{OPT} - K \sum_i \text{profit}'(o_i) \leq nK \iff K \sum_i \text{profit}'(o_i) \geq \text{OPT} - nK \geq \text{OPT} - \varepsilon P \]

Obs 2: $\text{profit}(S') \geq K \sum_i \text{profit}'(o_i)$

\[ \implies \text{profit}(S') \geq \text{OPT} - \varepsilon P \geq \text{OPT} - \varepsilon \text{OPT} = (1 - \varepsilon) \cdot \text{OPT} \]

Thm 1 KnapsackFPTAS is an FPTAS for Knapsack with $O(n^3/\varepsilon)$ running time.
Strong NP-hardness

An optimization problem is called strongly NP-hard, if it remains NP-hard with unary numbers.

Can the Knapsack problem be strongly NP-hard?

Thm 2 A strongly NP-hard problem has no pseudo-polynomial algorithm unless $P = NP$. 
FPTAS and Pseudo-Polynomial Algorithms

**Thm 3** Let $p$ be a polynomial and $\Pi$ be an NP-hard minimization problem with integral objective function and with $\text{OPT}(I) < p(|I_u|)$ for all instances $I$ of $\Pi$. If $\Pi$ has an FPTAS then there is a pseudo-polynomial algorithm for $\Pi$. 
FPTAS and Strong NP-hardness

**Corollary** Let $\Pi$ be an NP-hard optimization problem, that fulfils the requirements of Theorem 3. If $\Pi$ is strongly NP-hard then there is no FPTAS for $\Pi$ unless $P = NP$. 