Chapter 8
Approximation Algorithms

Algorithm Theory
WS 2015/16

Fabian Kuhn

Knapsack

- n items $1, \dots, n$, each item has weight $w_i > 0$ and value $v_i > 0$
- Knapsack (bag) of capacity W

- Goal: pack items into knapsack such that total weight is at most W and total value is maximized:

$$\max \sum_{i \in S} v_i$$

$$\text{s.t.} \quad S \subseteq \{1, \dots, n\} \text{ and } \sum_{i \in S} w_i \leq W$$

- E.g.: jobs of length w_i and value v_i, server available for W time units, try to execute a set of jobs that maximizes the total value

Approximation Algorithm

$O(n^2 V)$ if values are int.

- The algorithm has a parameter $\varepsilon > 0$
- We assume that each item alone fits into the knapsack
- We define:

$$V := \max_{1 \le i \le n} v_i, \qquad \forall i: \hat{v}_i := \left\lceil \frac{v_i n}{\varepsilon V} \right\rceil, \qquad \hat{V} := \max_{1 \le i \le n} \hat{v}_i$$

- We solve the problem with integer values $\hat{v}_i$ and weights w_i using dynamic programming in time $O(n^2 \cdot \hat{V})$
- If solution value $< V$, we take item with value V instead

max value

Theorem: The described algorithm runs in time $O(n^3/\varepsilon)$.

Proof:

$$\hat{V} = \max_{1 \le i \le n} \hat{v}_i = \max_{1 \le i \le n} \left\lceil \frac{v_i n}{\varepsilon V} \right\rceil = \left\lceil \frac{Vn}{\varepsilon V} \right\rceil = \left\lceil \frac{n}{\varepsilon} \right\rceil$$

Approximation Algorithm

Theorem: The approximation algorithm computes a feasible solution with approximation ratio at most $1 + \varepsilon$.

$\hat{v}_i = \left\lceil \frac{v_i n}{\varepsilon V} \right\rceil$

Proof:

- Define the set of all feasible solutions (subsets of $[n]$)

$$\mathcal{S} := \left\{ S \subseteq \{1, \dots, n\} : \sum_{i \in S} w_i \leq W \right\}$$

- $v(S)$: value of solution S w.r.t. values $v_1, v_2, \dots$
  $\hat{v}(S)$: value of solution S w.r.t. values $\hat{v}_1, \hat{v}_2, \dots$
- S^*: an optimal solution w.r.t. values $v_1, v_2, \dots$
  $\hat{S}$: an optimal solution w.r.t. values $\hat{v}_1, \hat{v}_2, \dots$
- Weights are not changed at all, hence, $\hat{S}$ is a feasible solution

Approximation Algorithm

Theorem: The approximation algorithm computes a feasible solution with approximation ratio at most $1 + \varepsilon$.

Proof:

- We have

$$v(S^*) = \sum_{i \in S^*} v_i = \max_{S \in \mathcal{S}} \sum_{i \in S} v_i \,,$$

$$\hat{v}(\hat{S}) = \sum_{i \in \hat{S}} \hat{v}_i = \max_{S \in \mathcal{S}} \sum_{S \in \mathcal{S}} \hat{v}_i$$

comp. by the dyn. progr.

- Because every item fits into the knapsack, we have

$$\forall i \in \{1, \dots, n\} \colon\ v_i \leq V \leq \sum_{j \in S^*} v_j$$

$\hat{v}_i \geq \frac{v_i n}{\varepsilon V}$

- Also: $\hat{v}_i = \left\lceil \frac{v_i n}{\varepsilon V} \right\rceil \implies v_i \leq \frac{\varepsilon V}{n} \cdot \hat{v}_i,$ and $\hat{v}_i \leq \frac{v_i n}{\varepsilon V} + 1$

Approximation Algorithm

$\frac{OPT}{ALG} \leq 1+\varepsilon$

Theorem: The approximation algorithm computes a feasible solution with approximation ratio at most $1 + \varepsilon$.

Proof:

$v_i \leq \frac{\varepsilon V}{n} \cdot \hat{v}_i$ $\quad$ $\hat{S}$ opt. w.r.t. $\hat{v}_i$ $\quad$ $\hat{v}_i \leq \frac{v_i n}{\varepsilon V} + 1$

- We have

$$v(S^*) = \sum_{i \in S^*} v_i \leq \frac{\varepsilon V}{n} \cdot \sum_{i \in S^*} \hat{v}_i \leq \frac{\varepsilon V}{n} \cdot \sum_{i \in \hat{S}} \hat{v}_i \leq \frac{\varepsilon V}{n} \cdot \sum_{i \in \hat{S}} \left(1 + \frac{v_i n}{\varepsilon V}\right)$$

- Therefore

$|\hat{S}| \leq n$ $\quad$ $\leq V$

$$v(S^*) = \sum_{i \in S^*} v_i \leq \frac{\varepsilon V}{n} \cdot |\hat{S}| + \sum_{i \in \hat{S}} v_i \leq \varepsilon V + v(\hat{S})$$

$\varepsilon V \leq \varepsilon v(\hat{S})$

- If $v(\hat{S}) \geq V$: $\qquad v(S^*) \leq (1 + \varepsilon) \cdot v(\hat{S})$

$v(\hat{S}) < V$

$\hookrightarrow V$

- Otherwise: algorithm solution value is V and

$$v(S^*) \leq (1 + \varepsilon) \cdot V$$

ALG

Approximation Schemes

- For every parameter $\varepsilon > 0$, the knapsack algorithm computes a $(1+\varepsilon)$-approximation in time $O(n^3/\varepsilon)$.
- For every fixed ε, we therefore get a polynomial time approximation algorithm
- An algorithm that computes an $(1+\varepsilon)$-approximation for every $\varepsilon > 0$ is called an approximation scheme.
- If the running time is polynomial for every fixed ε, we say that the algorithm is a polynomial time approximation scheme (PTAS)
- If the running time is also polynomial in $1/\varepsilon$, the algorithm is a fully polynomial time approximation scheme (FPTAS)
- Thus, the described alg. is an FPTAS for the knapsack problem