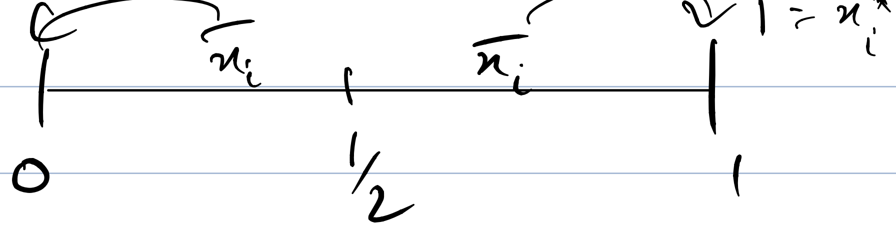

$\Rightarrow$ Since $w \geq 0$,

$$w^T x^* = \sum w_i x_i^* \leq 2 \sum w_i \bar{x}_i$$

$$= 2\, w^T \bar{x}.$$

$\square$.

Approximation algorithm for Knapsack.

Knapsack: items $1, \ldots, n$

(integers) weight $w_1, \ldots, w_n$.

(integers) values $v_1, \ldots, v_n$.

$\underset{S \subseteq [n]}{\text{maximize}} \quad \sum_{i \in S} v_i$ subject to

$$\sum_{i \in S} w_i \leq W.$$

Defn: [Polynomial time approximation sheme (PTAS)]

Approx. algorithm A for some opt. problem O, takes an additional input ε, and it outputs a solution which is within $(1+\varepsilon)$-factor of the optimal solution. A must run in polynomial time (in input instance of O) for any constant $\varepsilon > 0$.

e.g. A can run in time

$$n^{O(1/\varepsilon)} \quad \text{or} \quad \text{poly}(n) \cdot 2^{O(1/\varepsilon)}.$$

PTAS for Knapsack.

Will design the following algorithm A:

A solves Knapsack in time

$$O(n^2 v^*), \quad v^* = \max_{i \in [n]} v_i$$

$$\left([n] = \{1, \dots, n\} \right).$$

(Based on dynamic programming; pseudopoly time).

Idea: → Pick a parameter b,

round the values (v_i's) to

multiples of b.

→ Scale the values down by b, and use $\mathcal{A}$ to solve the instance.

→ Output the solution obtained.

Let's make things precise:

→ Will pick b later.

→ Define $\tilde{v}_i = \lceil \frac{v_i}{b} \rceil \cdot b$

→ $\hat{v}_i = \lceil \frac{v_i}{b} \rceil = \frac{\tilde{v}_i}{b}$.

$$\boxed{v_i \le \tilde{v}_i \le v_i + b.}$$

Approximate-Knapsack(ε , input to knapsack):

(i) Set $b = \left\lfloor \frac{v^*}{\left(\frac{2n}{\varepsilon}\right)} \right\rfloor$;

(recall $v^* = \max_i v_i$)

(ii) Output solution of algorithm A on knapsack using $\hat{v}_i$'. Let S be the output set.

Some assumptions: (i) $1/\varepsilon$ is an integer.

(ii) $\forall i \in [n]$, $w_i \leq W$.

Claim: The above algorithm runs in

$O(n^3/\varepsilon)$ time.

Pf: Run time is $O\left(n^2 \cdot \max_i \hat{v}_i\right)$.

$$\hat{v}_i \le \left\lceil \frac{v^*}{b} \right\rceil \le 1 + \frac{2n}{\varepsilon}$$

$$\Rightarrow O(n^3/\varepsilon).$$

$\square$.

Proof that is a $(1+\varepsilon)$-approx algorithm.

Obs 1 S is indeed a 'valid solution'

i.e $\sum_{i \in S} w_i \le W$.

$\square$.

Obs 2 S is an optimal solution using values $\hat{v}_i$ iff S is also an optimal solution using values $\tilde{v}_i$.

$\square$

Consider any $T \subseteq [n]$ s.t

$$\sum_{i \in T} w_i \leq W .$$

We will prove : $(1+\varepsilon) \sum_{i \in S} v_i \geq \sum_{i \in T} v_i .$

Pf:

$$\sum_{i \in T} v_i \underset{\uparrow}{\leq} \sum_{i \in T} \tilde{v}_i \underset{\uparrow}{\leq} \sum_{i \in S} \tilde{v}_i$$

$\forall i \; v_i \leq \tilde{v}_i$

since S is an opt solution for Knapsack using $\tilde{v}_i$.

$$\underset{\uparrow}{\leq} \sum_{i \in S} v_i + nb ,$$

$\forall i \; \tilde{v}_i \leq v_i + b$

$$\sum_{i \in T} v_i \le \sum_{i \in S} v_i + nb.$$

Thus, enough to show,

$$\frac{nb}{\varepsilon} \le \sum_{i \in S} v_i .$$

Proving the above inequality!

$$\sum_{i \in S} \tilde{v}_i \ge \max_{j \in [n]} \tilde{v}_j$$

$$\Rightarrow \sum_{i \in S} v_i \ge \max_{j \in [n]} \tilde{v}_j - nb.$$

$$\left(\text{since } \tilde{v}_i \le v_i + b\right)$$

Observe that

$$\max_{j \in [n]} \tilde{v}_j = \left\lceil \frac{v^*}{b} \right\rceil \cdot b$$

$$> 2n \quad b$$

$- \quad \frac{\quad}{\varepsilon}$

since $\left\lceil \frac{V^*}{b} \right\rceil \geq \frac{2n}{\varepsilon}$,

by our choice of b.

$$\Rightarrow \sum_{i \in S} v_i \geq \left(\frac{2}{\varepsilon} - 1 \right) nb \geq \frac{nb}{\varepsilon},$$

for $\varepsilon < 1$.

$\square$.