A problem is strongly NP-hard if it does not admit pseudo-polynomial algorithms.

PTAS: \((1+\epsilon)\)-approx in \(f(n, \frac{1}{\epsilon})\) time where \(f\) is poly-

FPTAS: Special case of PTAS where \(f = \text{poly}(\|I\|)\cdot\text{poly}(\frac{1}{\epsilon})\) \(\forall \epsilon \in \mathbb{R}^+\).

Knapsack
- \(n\) items with weight \(w_i\), profit \(p_i\) for the \(i\)th item
- Knapsack of capacity \(W\)
- Find subset of items of maximum value whose total weight is \(\leq W\)

- One of Karp’s 21 NP-hard problems

\((3\text{-SAT} \rightarrow \text{Knapsack})\)

- Also called 0-1 knapsack; fractional version where fractions of items can be picked from an exact greedy algorithm

- Pseudo-poly algo: Dynamic program
  
  \[ A(i, p) = \begin{cases} 
  \text{min weight combination among first } i \text{ items that has overall profit of } p, \\
  \min(A(i, p), A(i - 1, p - p_i) + w_i) & \text{o.w.} 
  \end{cases} \]

# of entries: \(O(n^2 P)\) where \(P = \max_i p_i\)

How to convert to an FPTAS:

\[ K = \frac{\epsilon P}{n}, \ \text{round all profits to multiples of } K \]

\[ \text{Error} \leq \frac{\epsilon P}{n}, n = \frac{\epsilon P}{\epsilon}, \quad (1-\epsilon) - \text{approx} \]
\[ \text{Error} \leq \frac{3^p}{n}, \quad n = 3^p \int \left( -3 \right)^{-1} \text{-approx} \]

\[ \text{OPT} \geq p \]

\# of entries in DP table: \( O \left( \frac{n^2 P}{k} \right) = O \left( \frac{n^2}{\epsilon} \right) \)

**Bin packing and Load Balancing**

Bin packing: Items of sizes \( a_1, a_2, \ldots, a_n \) to be packed in minimum number of bins of unit size.

If \( a_i \leq 1 \) feasable, else infeasable.

First Fit: Place items in arbitrary order into existing bins; if not possible, start a new bin.

Analysis: At most one bin is less than \( \frac{1}{2} \) full.

Better Algo:

If there are \( K \) item types and each item is of size \( \geq \frac{\epsilon}{K} \), then the number of configurations of a bin is \( K^{\epsilon} \). Since there are at most \( n \) bins, number of configurations overall is \( n K^{\epsilon} \). To ensure this, remove all items of size \( < \frac{\epsilon}{K} \) and round all other items into equi-sized groups containing \( n^{\epsilon/2} \) items each. Then, round the sizes of all items to the max size its group.

Lemma: OPT changes by a factor of \( 1 + \epsilon \).

Proof: In original OPT, replace each item of class \( i \) by a rounded item of class \( (i-1) \). Place \( n^{\epsilon/2} \) items of highest class to private bins. Finally note that OPT \( \geq n^{\epsilon/2} \).

To complete the algo, use FIRSTFIT for the items of size \( \leq \frac{\epsilon}{K} \).
If new bins are used, all except one bin must be full up to a volume of \((1 - \varepsilon)\).

Running time: \(n \left( \frac{1}{\varepsilon^2} \right)^{1/\varepsilon}\)