5. How to solve a NPC problem

Example 1: Bin Packing problem

Suppose we have an unlimited number of bins, each of capacity 1, and \( n \) objects with sizes \( s_1, s_2, \ldots, s_n \), where \( 0 < s_i \leq 1 \).

- **Optimization problem**: Determine the smallest number of bins into which objects can be packed and find an optimal packing.

- **Decision problem**: Do the objects fit in \( k \) bins?

**Theorem.** Bin Packing problem is NPC

Proof. reduced from the Subset Sum – omit
5. How to solve a NP-complete problem

Approximate algorithm for the Bin Packing

▶ First-fit strategy (greedy):

places an object in the first bin into which it fits.

▶ Example:
Objects = \{0.8, 0.5, 0.4, 0.4, 0.3, 0.2, 0.2, 0.2\}

First-fit strategy solution:

\[
\begin{array}{cccc}
B_1 & B_2 & B_3 & B_4 \\
0.2 & 0.4 & 0.3 & 0.2 \\
0.8 & 0.5 & 0.4 & 0.2 \\
\end{array}
\]

▶ Optimal packing:

\[
\begin{array}{ccc}
B_1 & B_2 & B_3 \\
0.2 & 0.3 & 0.4 \\
0.8 & 0.5 & 0.4 \\
\end{array}
\]
5. How to solve a NP-complete problem

**Theorem.** Let $S = \sum_{i=1}^{n} s_i$.

1. The optimal number of bins required is at least $\lceil S \rceil$.
2. The number of bins used by the first-fit strategy is never more than $\lceil 2S \rceil$. 
5. How to solve a NP-complete problem

The vertex-cover problem:

- A vertex-cover of an undirected graph $G = (V, E)$ is a subset set of $V' \subseteq V$ such that if $(u, v) \in E$, then $u \in V'$ or $v \in V'$.

In other words, each vertex “covers” its incident edges, and a vertex cover for $G$ is a set of vertices that covers all edges in $E$.

- The size of a vertex cover is the number of vertices in it.

- Decision problem: determine whether a graph has a vertex cover of a given size $k$.

- Optimization problem: find a vertex cover of minimum size.

- Theorem. The vertex-cover problem is NPC.
5. How to solve a NP-complete problem

The vertex-cover problem:

- An approximate algorithm
  
  \[ C' = \emptyset \]
  \[ E' = E \]
  
  while \( E' \neq \emptyset \)
  
  \[
  \text{let } (u, v) \text{ be an arbitrary edge of } E' \\
  C = C \cup \{u, v\} \\
  \text{remove from } E' \text{ every edge incident on either } u \text{ or } v.
  \]
  
  endwhile
  
  return \( C \)

- **Theorem.** The size of the vertex-cover by the approximate algorithm is no more than twice the size of an optimal vertex cover.