0-1 Knapsack problem

Knapsack with some capacity, say B
Each of the n objects have a profit \( p_i \) and some weight \( w_i \):

Maximize \( \sum p_i x_i \) \( \text{s.t. } \sum w_i x_i \leq B \)

\( x_i = \begin{cases} 1 & \text{if object } i \text{ is chosen} \\ 0 & \text{otherwise} \end{cases} \)

Consider all possibilities i.e. all the \( 2^n \) possibilities of choosing the objects and choose the most profitable: Brute force/exhaustive

\( O(2^n \cdot n) \sim O(2^n) \)

There exist faster, polynomial time algorithms to solve linear prog (LP) (not simplex)
Choose the most profitable element among the remaining till we can't fit any more

\[ B = 15 \quad x_1, x_2, x_3, x_4 \]

profit 10 10 12 18

weight 2 4 6 9

\[ x_4 (18, 6) \quad x_3 (18+12, 0) \]

\[ \{ x_1, x_2, x_4 \} \]

\[ 38 \]

No efficient polynomial time algorithm is known for 0-1 Knapsack.

Use heuristics: and try to prove some properties about the heuristics even if it doesn't give the best solution.
A very general heuristic

Branch and Bound

\[ \beta = 15 \]
\[ 10 \ 10 \ 12 \ 18 \]
\[ 2 \ 4 \ 6 \ 9 \]
\[ 5 \ 2.5 \ 2 \ 2 \]

\[ x_1 = 0 \rightarrow x_1 = 1 \]
\[ x_2 = 0 \]
\[ x_2 = 1 \]
\[ x_3 = 0 \rightarrow x_3 = 1 \]
\[ x_4 = 0 \rightarrow x_4 = 1 \]

If residual capacity
\[ = c \]
and the map profit/weight
\[ \text{max profit/weight} = m \]
\[ \implies \text{upper bound} \leq m x c \]

(better upper bounds welcomed)

Estimate function: What is the best that we can get if we choose a subset of \( \{ x_1, x_2, \ldots, x_k \} \) (upper bound)

\[ \frac{\text{profit}}{\text{weight}} \]

rail