The Fraction Knapsack Problem: A Greedy Example

Given:
A knapsack of capacity \( W \) and \( n \) items with weights \( w_1, w_2, \ldots, w_n \) and values \( v_1, v_2, \ldots, v_n \).

Find:
\( \alpha_1, \alpha_2, \ldots, \alpha_n \) maximizing \( \sum \alpha_i v_i \).

Subject to: \( 0 \leq \alpha_i \leq 1 \), and \( \sum \alpha_i w_i = W \).

[Note: "0-1 Knapsack" same, except \( \alpha_i = 0 \) or \( 1 \).]

Examples

<table>
<thead>
<tr>
<th>Object</th>
<th>Weight</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liqueur-Filled Bon Bons</td>
<td>1</td>
<td>$12</td>
</tr>
<tr>
<td>Dark Chocolate Truffles</td>
<td>2</td>
<td>$18</td>
</tr>
<tr>
<td>Milk Choc. Spring Surprise</td>
<td>3</td>
<td>$24</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccccc}
| \text{Item} | w_i & v_i & \alpha_i w_i & \alpha_i v_i & \alpha_i & \alpha_i v_i |
|-------------|-----|-----|-------------|-----------|-----|-----------|
| BB          | 1   | $12 | 0           | 0         | 1   | $12       |
| T           | 2   | $18 | 1           | 1         | 1   | $18       |
| SS          | 3   | $24 | 1           | 2/3       | 1   | $16       |
| Total       |     | $42 | $45         |           |     | $46       |
\end{array}
\]

Greedy Solution

- Order by decreasing value per unit weight (renumbering as needed)

\[
\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \cdots \geq \frac{v_n}{w_n}
\]

- Take as much 1 as possible, then as much 2 as possible, …

The Greedy Choice Pays

Claim 1: 3 an optimal solution with as much as possible of item 1 in the knapsack, namely \( \alpha_1 = \min(w_1, W) \). Equivalently \( \alpha_1 = \min(w_1, W/w_1) \).

Proof: Among all optimal solutions, let \( \beta_1, \beta_2, \ldots, \beta_n \) be one with maximum \( \beta_1 \), but suppose (for the sake of contradiction) \( \beta_1 < \alpha_1 \). Since \( \beta_1 \) has less of 1 than \( \alpha_1 \), it must have more of something else, say \( j \), i.e. \( \beta_j > \alpha_j \). Form \( \beta' \) from \( \beta \) by carrying a little more 1 and less \( j \), say \( \epsilon = \min(\beta_1 - \alpha_1, w_1 (\alpha_1 - \beta_1)) > 0 \). Then \( \beta'_1 \) will not have a lower value than \( \beta_1 \), since \( v_1/w_1 \), \( v_j/w_j \) \( \geq 0 \), but \( \epsilon (v_1/w_1, v_j/w_j) = 0 \), but \( \beta'_1 > \beta_1 \), contradicting our choice of \( \beta_1 \). QED

Optimal Sub-solutions

Claim 2: The best solution for any given \( \alpha_1 \) has \( \alpha_2, \ldots, \alpha_n \) equal to an optimal solution for the smaller knapsack problem having items 2, 3, …, \( n \) and capacity \( W - \alpha_1 w_1 \).

Proof: If not, we could get a better solution.
Keys to Greedy Algorithms

“Greedy Choice Property”: Making a locally optimal (“greedy”) 1st step cannot prevent reaching a global optimum. [E.g., see Claim 1.]

“Optimal Substructure”: The optimal solution to the problem contains optimal solutions to subproblems. [E.g., see Claim 2. True of Dynamic Programming, too.]

NOTE:
- Greedy algorithms are very natural for optimization problems, but
- they don’t always work
- E.g., if you try greedy approach for 0-1 knapsack on the candy example, it will choose to take all of BB & T, for a total value of $30, well below the optimal $42
- So: Correctness proofs are important!