CIS 575. Introduction to Algorithm Analysis
Material for April 15, 2024

A Greedy Solution to the Fractional Knapsack Problem

The topic of this note is mentioned in *Cormen*'s Section 15.2.

1 Fractional Knapsack Problem

We have already encountered the generic knapsack problem: given n items, where each item i has a *weight* w_i and a *value* v_i (both positive), we want to select items such that their total value is as big as possible, while their total weight does not exceed a given capacity W.

We studied the *binary* variant where for each item, we either pick it or omit it. But now *assume* that

1. we may cut each item i in pieces, and put only a **fraction** x_i into the knapsack
2. the weight of that piece will be $x_i \cdot w_i$
3. the value of that piece will be $x_i \cdot v_i$.

Note that these assumptions may not be satified for a given application: the first assumption holds for only certain items, and the third holds for even fewer kinds of items.

As for the binary knapsack problem, our task can thus be formalized as follows: for each $i \in 1 \ldots n$, find x_i ("how much of item i") such that we do not exceed capacity:

$$\sum_{i=1}^{n} x_i \cdot w_i \leq W$$

while we maximize the total value:

$$\sum_{i=1}^{n} x_i \cdot v_i.$$

But whereas in the binary knapsack problem, each x_i must be either zero or one, for the fractional knapsack problem, we shall allow x_i to be any real number between zero and one, both included. Thus, rather than requiring $x_i \in \{0, 1\}$, we now only require $x_i \in [0, 1]$.

Which problem do we expect to be easier? The binary problem involves "only" 2^n possible solutions, whereas the fractional problem involves infinitely many possible solutions. In principle, we can thus solve the binary problem by brute force, whereas there is no way to do this for the fractional problem.

Still, perhaps surprisingly, it turns out that the fractional knapsack problem is easier, in that it allows an efficient greedy algorithm. Intuitively, this is because it allows us to pick "just the right amount" of an item, whereas the binary problem may force us to choose between "taking too much" and "taking too little".

Greedy Strategies We shall now explore various greedy strategies for the fractional knapsack problem, and shall illustrate them by the below example where the capacity W is 100.

i	1	2	3	4	5
w	10	20	30	40	50
v	20	30	66	40	60

Let us first explore the strategy of giving preference to the **most valuable** items: we then first take item 3, and then item 5; next we would like to take item 4 but we have only a capacity of 20 left so we can only take *half* of that item. Formally, this strategy produces the x-values

x	0	0	1	0.5	1

for a total value of 146.

Let us next explore the strategy of giving preference to the **least heavy** items: we can then take all of the first 4 items, but cannot take anything of item 5. Formally, this strategy produces the x-values

x	1	1	1	1	0

for a total value of 156.

We see that for the given example, it is better to prefer the least heavy items than to prefer the most valuable items. But it is obviously possible to construct an example where it is the other way round. And indeed, none of these two strategies will always produce an optimal schedule. For that purpose, we must *combine* them, so as to get as much "bang for the buck" (value for the weight) as possible:

Preferring Precious Items We define the *preciousness* of an item i (written p_i) as $\frac{v_i}{w_i}$. We can augment our example to include information about preciousness:

i	1	2	3	4	5
w	10	20	30	40	50
v	20	30	66	40	60
p	2	1.5	2.2	1	1.2

We now explore the strategy of giving preference to the *most precious* items: we take first item 3, next item 1, next item 2; we would like to next take item 5 but are left with a capacity of only 40 so we can only take four fifth of item 5. Formally, this strategy produces the x-values

x	1	1	1	0	0.8

for a total value of $20 + 30 + 66 + 0.8 \cdot 60 = \mathbf{164}$.

We see that this is better than our first two strategies. And this is not a coincidence; it is **always optimal to pick the most precious items first** as we shall now prove:

Proof: For ease of notation, assume that the items are listed in decreasing order of preciousness:

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \ldots \geq \frac{v_n}{w_n}.$$

Let $X = x_1 \ldots x_n$ be the solution produced by the strategy of prefering the most precious items. Let Y be *any* solution; our goal is to prove that the value from Y cannot be greater than the value from X.

Let k be the first item that cannot completely fit (there must exist such an item as otherwise the problem is trivial). Thus $1 \leq k \leq n$ and $x_i = 1$ for all $i < k$, $x_k < 1$ (with $x_k = 0$ possible), and $x_i = 0$ for all $i > k$.

A key observation is that for all $i \in 1 \ldots n$ we have

$$(x_i - y_i)\frac{v_i}{w_i} \geq (x_i - y_i)\frac{v_k}{w_k}. \tag{1}$$

This follows from a case analysis on i:

- if $i = k$, the claim is trivial
- if $i < k$, then
$$\frac{v_i}{w_i} \geq \frac{v_k}{w_k}$$
and as $x_i - y_i \geq 0$ (since $x_i = 1$) we can multiply by $x_i - y_i$ to get the desired inequality.
- if $i > k$, then
$$\frac{v_i}{w_i} \leq \frac{v_k}{w_k}$$
and as $x_i - y_i \leq 0$ (since $x_i = 0$) we can multiply by $x_i - y_i$ to *reverse* the inequality which is what we desire.

We now have the calculation

$$\begin{array}{rcl}
 & & \text{value of}(X) - \text{value of}(Y) \\
 & = & \sum_{i=1}^{n} x_i \cdot v_i - \sum_{i=1}^{n} y_i \cdot v_i \\
 & = & \sum_{i=1}^{n} (x_i - y_i)\frac{v_i}{w_i} w_i \\
\text{(by Inequality 1)} & \geq & \sum_{i=1}^{n} (x_i - y_i)\frac{v_k}{w_k} w_i \\
 & = & \frac{v_k}{w_k} \sum_{i=1}^{n} (x_i - y_i) w_i \\
 & = & \frac{v_k}{w_k} (\sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i) \\
 & = & \frac{v_k}{w_k} (W - \sum_{i=1}^{n} y_i w_i) \\
\text{(as } Y \text{ does not exceed } W\text{)} & \geq & 0
\end{array}$$

which shows that the value of X, the solution produced by preferring the most precious items, is indeed at least as good as any other solution. □

One may ask: could we apply our greedy strategy also to the *binary* knapsack problem? We certainly can, but there is no guarantee that we get an optimal solution. To see this, consider the example

i	1	2	3
w	6	5	5
v	9	6	4
p	1.5	1.2	0.8

with capacity 10. Then our greedy strategy will pick item 1 (the most precious) but in the binary setting we cannot pick anything else and thus get a total value of only 9. But by picking the less precious items 2 and 3 we can get a total value of 10.

Recall that when W is relatively "small", there exists an efficient dynamic programming algorithm for the binary knapsack problem, but for large W that algorithm may not be feasible.