Fractional knapsack

Vassos Hadzilacos

A thief breaks into a store holding a knapsack that can carry up to a maximum weight $W > 0$. The store contains items $1, 2, \ldots, n$, where item $i$ has value $v_i > 0$ and weight $w_i \geq 0$. The thief can steal some amount $x_i$ of item $i$, where $0 \leq x_i \leq w_i$; the value of this amount of item $i$ is $(x_i/w_i) \cdot v_i$, i.e., the fraction of the item’s weight stolen times the item’s value. The thief must decide what fraction of each item to steal, so as to maximize the total value of the stolen goods, subject to the constraint that their total weight must not exceed the knapsack’s capacity $W$.

More precisely, a knapsack is a vector $S = (x_1, \ldots, x_n)$, where $x_i$ represents the amount (in weight) of item $i$ stolen. The value of $S$, denoted $\mathcal{V}(S)$, is defined as $\mathcal{V}(S) = \sum_{i=1}^{n} (x_i/w_i) \cdot v_i$. $S$ is a feasible knapsack if (a) for each $i$, $1 \leq i \leq n$, $0 \leq x_i \leq w_i$, and (b) $\sum_{i=1}^{n} x_i \leq W$. $S$ is an optimal knapsack if it is feasible knapsack of maximum value; i.e., for all feasible knapsacks $S' = (x'_1, \ldots, x'_n)$, $\mathcal{V}(S) \geq \mathcal{V}(S')$. The fractional knapsack problem can now be specified as follows:

**Input.** $W > 0$, and $v_i > 0, w_i \geq 0$ for each $i$, $1 \leq i \leq n$.

**Output.** An optimal knapsack.$^1$

The 0-1 knapsack problem is the version of this problem where we require that $x_i = 0$ or $x_i = w_i$, for each $i$; in other words, the thief can either steal an item in its entirety ($x_i = w_i$) or not at all ($x_i = 0$); partial items have no value. The 0-1 knapsack problem seems to be computationally very hard. It belongs to the famous class of NP-complete problems, which has the following property: (a) we don’t know how to solve any NP-complete problem efficiently (i.e., in time that is a polynomial in the size of the input), and (b) if any one of these problems has an efficient solution, then they all do! The class of NP-complete problems includes many optimization problems that are important in practice.

In contrast to the 0-1 knapsack problem, the fractional knapsack problem can be solved by means of a simple and efficient greedy algorithm. Informally, the algorithm is as follows: Consider the items in non-increasing value-to-weight ratio. Add items to the knapsack one at a time, in this order, until we reach an item whose addition would cause the knapsack’s capacity $W$ to be exceeded; add the largest fraction of that item that fits into the knapsack, and stop. This is expressed in pseudocode below:

1. sort the items so that $v_1/w_1 \geq v_2/w_2 \geq \ldots \geq v_n/w_n$
2. $s := 0$ # total weight of stolen items in knapsack so far
3. $i := 1$ # next item to be considered
4. while $i \leq n$ and $s + w_i \leq W$ do
5. \hspace{1em} $x_i := w_i$
6. \hspace{1em} $s := s + w_i$
7. \hspace{1em} $i := i + 1$
8. if $i \leq n$ then
9. \hspace{1em} $x_i := W - s$
10. for $j = i + 1$ to $n$ do $x_j := 0$
11. return $(x_1, \ldots, x_n)$

It is straightforward to see that this algorithm can be implemented so that it runs in $O(n \log n)$ time. We now prove that the knapsack returned by this algorithm is optimal.

---

$^1$For simplicity, let’s assume that $W$, and all $v_i, w_i, x_i$ are integers. This makes it obvious that an optimal knapsack exists, since there is only a finite number of feasible knapsacks. An optimal knapsack exists even if these quantities are reals, but proving existence in that case is not as trivial.
Proof of optimality. If $\sum_{i=1}^{n} w_i \leq W$ (i.e., the knapsack can accommodate all items), then clearly the optimal solution is $(w_1, w_2, \ldots, w_n)$, and this is what the algorithm returns in this case. So, in the rest of the proof we assume that $\sum_{i=1}^{n} w_i > W$.

**Fact 1.** Let $T = (y_1, \ldots, y_n)$ and $T' = (y'_1, \ldots, y'_n)$ be knapsacks, and $k, \ell$ be items such that $k < \ell$. If $y_i = y'_i$ for all items $i \notin \{k, \ell\}$, and for some $\Delta > 0$, $y_k = y'_k + \Delta$, and $y_\ell = y'_\ell - \Delta$, then $\mathcal{V}(T) \geq \mathcal{V}(T')$.

**Proof.** From the definition of value of knapsack, we have

$$\mathcal{V}(T) - \mathcal{V}(T') = (\Delta/w_k) \cdot v_k - (\Delta/w_\ell) \cdot v_\ell = \Delta \cdot (v_k/w_k - v_\ell/w_\ell) \geq 0$$

where the last inequality holds because $k < \ell$ and therefore $v_k/w_k \geq v_\ell/w_\ell$. \hfill \Box

Let $S = (x_1, \ldots, x_n)$ be the schedule returned by the greedy algorithm.

**Fact 2.** Knapsack $S$ satisfies the following:

(a) There is some item $t$ such that (i) $0 \leq x_t \leq w_t$, (ii) for all $i < t$, $x_i = w_i$, and (iii) for all $i > t$, $x_i = 0$.

(b) $\sum_{i=1}^{n} x_i = W$.

**Proof.** From the algorithm’s description it is clear that, for some $t$, the algorithm assigns $x_i = w_i$ to the first $t - 1$ items in line 5, assigns part of item $t$’s weight to $x_t$ in line 9, and assigns 0 to all remaining items in line 10. Thus, part (a) of the lemma holds. A straightforward induction shows that, at the end of the $i$-th iteration of the loop in lines 4–7, $s = \sum_{j=1}^{i} w_j$. Since, by assumption, $\sum_{i=1}^{n} w_i > W$, the algorithm exits the while loop with $i \leq n$. So, by the assignments in lines 9 and 10, $\sum_{i=1}^{n} x_i = W$. \hfill \Box

We say that two knapsacks $T = (y_1, \ldots, y_n)$ and $T' = (y'_1, \ldots, y'_n)$ first differ on item $i$ if $y_i \neq y'_i$ and for all $j < i$, $y_j = y'_j$.

Suppose, for contradiction, that $S$ is not optimal. Therefore, every optimal knapsack first differs from $S$ on some item $i$. Let $k$ be the maximum item on which some optimal knapsack first differs from $S$; i.e.,

(*) there is some optimal knapsack that first differs from $S$ on item $k$; and for every optimal knapsack $S'$, if $S'$ first differs from $S$ on item $j$ then $j \leq k$.

Among all optimal knapsacks that first differ from $S$ on item $k$, let $S^* = (x^*_1, \ldots, x^*_n)$ be one with the maximum $k$-th component; i.e.,

(†) $S^*$ first differs from $S$ on $k$; and for every optimal knapsack $\hat{S} = (\hat{x}_1, \ldots, \hat{x}_n)$, if $\hat{S}$ first differs from $S$ on $k$ then $\hat{x}_k \leq x^*_k$.

**Fact 3.** $x_k > x^*_k$.

**Proof.** Suppose, for contradiction, that $x_k < x^*_k$. (It cannot be that $x_k = x^*_k$, since $S, S^*$ first differ on $k$.) Since $x_k < x^*_k \leq w_k$, by Fact 2(a), we must have that, for all $i > k$, $x_i = 0$. We have

$$W = \sum_{i=1}^{n} x_i \quad \text{[by Fact 2(b)]}$$

$$= (\sum_{i=1}^{k-1} x_i) + x_k + (\sum_{i=k+1}^{n} x_i) \quad \text{[since $S, S^*$ first differ on $k$, and $x_i = 0$ for all $i > k$]}$$

$$= (\sum_{i=1}^{k-1} x^*_i) + x_k \quad \text{[since we assumed that $x_k < x^*_k$, and $x^*_i \geq 0$ for all $i$]}$$

$$< (\sum_{i=1}^{k-1} x^*_i) + x^*_k + (\sum_{i=k+1}^{n} x^*_i) \quad \text{[by Fact 2(b)]}$$

$$= \sum_{i=1}^{n} x^*_i$$
This contradicts that $S^*$, being optimal, is feasible. \hfill \Box

**Fact 4.** For some $\ell > k$, $x_\ell < x_\ell^*$.  

**Proof.** Suppose, for contradiction, that for all $\ell > k$, $x_\ell \geq x_\ell^*$. Then, from the fact that $S$ and $S^*$ first differ on $k$, Fact 3, the fact that $v_k > 0$, and the definition of $V(\cdot)$, it follows that $V(S) > V(S^*)$. Since, by Fact 2, $S$ is feasible, this contradicts that $S^*$ is optimal. \hfill \Box

Let $\Delta = \min(\Delta^*, x_k - x_k^*)$. By Facts 3 and 4, $\Delta > 0$. Define now the knapsack $\hat{\mathcal{S}} = (\hat{x}_1, \ldots, \hat{x}_n)$ as follows:

\[
\hat{x}_i = x_i^*, \quad \text{for all items } i \neq k, \ell \\
\hat{x}_k = x_k^* + \Delta \\
\hat{x}_\ell = x_\ell^* - \Delta
\]

We now argue that $\hat{\mathcal{S}}$ is feasible. Intuitively this is because we added to and subtracted from the $k$-th and $\ell$-th components the same amount, so the sum of the amounts in $\hat{\mathcal{S}}$ did not change relative to $S^*$; the amount by which we incremented the $k$-th component did not cause it to exceed $w_k$; and the amount by which we decreased the $\ell$-th component did not cause it to go below zero. More precisely,

- $\sum_{i=1}^n \hat{x}_i = \sum_{i=1}^n x_i^* \leq W$ (since $S^*$ is feasible);
- for all $i \neq k, \ell$, $0 \leq \hat{x}_i \leq w_i$ (since $\hat{x}_i = x_i^*$, and $S^*$ is feasible);
- $\hat{x}_k = x_k^* + \Delta \leq x_k^* + (x_k - x_k^*) = x_k \leq w_k$ (since $\Delta \leq x_k - x_k^*$ and, by Fact 2, $S$ is feasible); and
- $\hat{x}_\ell = x_\ell^* - \Delta \geq x_\ell^* - x_\ell^* = 0$ (since $\Delta \leq x_\ell^*$).

By Fact 1, $V(\hat{\mathcal{S}}) \geq V(S^*)$. Since $\hat{\mathcal{S}}$ is feasible and $S^*$ is optimal, $\hat{\mathcal{S}}$ is also optimal. There are two cases.

**Case 1.** $x_k - x_k^* \leq x_\ell^*$. Then $\Delta = x_k - x_k^*$ and so $\hat{x}_k = x_k$. Since for all $i < k$, $\hat{x}_i = x_i^* = x_i$, it follows that $\hat{\mathcal{S}}$ is an optimal knapsack that first differs from $S$ on an item greater than $k$. This contradicts the definition of $k$ (see (\star)).

**Case 2.** $x_k - x_k^* > x_\ell^*$. Then $\Delta = x_\ell^*$ and so $\hat{x}_k = x_k^* + x_\ell^* < x_k$ (by the assumption of this case). Since for all $i < k$, $\hat{x}_i = x_i^* = x_i$, it follows that $\hat{\mathcal{S}}$ is an optimal knapsack that first differs from $S$ on $k$, but $\hat{x}_k = x_k^* + \Delta > x_k^*$ (since $\Delta > 0$). This contradicts the definition of $S^*$ (see (\dagger)).

Since both cases lead to a contradiction, the assumption that $S$ is not optimal must be false. Therefore, $S$ is optimal.