ナップサック問題の近似解法

Approximation Algorithm of the Knapsack Problem

問題　real numbers a[1], a[2], …, a[n], w[1], w[2], …, w[n], b given
a[i] are profit of item i, w[i] are size, and b is the size of knapsack. Let I be a subset of {1, 2, …, n}. Find I that gives maximum to

$$\text{Max} \ \sum_{i \in I} a[i] \quad \text{subject to} \ \sum_{i \in I} w[i] \leqq b$$

We maximize the total profit while the total size is limited by b.

アルゴリズム (approximation algorithm)

(1) Sort a[i]/w[i] i=1,…, n in non-increasing order.
Suppose a[i]/w[i] are sorted. That is, item 1 is most profitable, etc.

(2) Pick up item 1, 2, … while the total size is not greater than b, that is, w[1]+ … + w[k] ≦ b for some k. Let G = a[1] + … + a[k]. Greedy solution.

(3) Find the maximum of a[1], …, a[n]. Let the maximum be M. Safe guard.

(4) Let the approximation solution S be max{G, M}.

Theorem. Let R be the real solution. Then S ≧ R/2. That is, S is as bad as half of real solution in the worst case. For random data, S is very close to R.
Proof. In the step (2), we assume we break the size constraint at k+1 for the first time. That is, w[1]+ … +w[k]+w[k+1] > b. a[i]/w[i] is the profit ratio of item i. Observe a[1]+…+a[k]+a[k+1] ≧ R, since we pack items into the knapsack from the most profitable items and the total size exceeds b while the size for the real solution does not exceed b. Also observe M ≧ a[k+1]. Thus

$$R \leqq a[1]+\cdots+a[k]+a[k+1] = G + a[k+1] \leqq G + M \leqq S + S = 2S.$$

Theorem. The time complexity of the algorithm is O(nlog n)
Proof. The break-down of the algorithm is given by
(1) O(nlog n), (2) O(n), (3) O(n), (4) O(1).

Example. a[1]=2, w[1]=1, a[2]=a[3]=b/2, w[2]=w[3]=b/2.
G=2+b/2. M=b/2. S=2+b/2. R=b/2+b/2=b. That is S is close to R/2.