The Knapsack Problems

Date Generated: April 20, 2006

Background
Reference: Larson, et. al.

The **0-1 knapsack problem** is posed as follows. A thief robbing a store finds n items; the i-th item is worth v_i dollars and weighs w_i pounds, where v_i and w_i are integers. He wants to take as valuable a load as possible, but he can carry at most W pounds in his knapsack for some integer W. Which items should he take? This is called the 0-1 knapsack problem because each item must either be taken or left behind; the thief cannot take a fractional amount of an item or take an item more than once.

In the **fractional knapsack problem**, the setup is the same, but the thief can take fractions of items, rather than having to make a binary (0-1) choice for each item. You can think of an item in the 0-1 knapsack problem as being like a gold ingot, while an item in the fractional knapsack problem is more like gold dust.

Both knapsack problems exhibit the optimal substructure property. For the 0-1 problem, consider the most valuable load that weighs at most W pounds. If we remove item j from this load, the remaining load must be the most valuable load weighing at most $W - w_j$ pound that the thief can take from the $n-1$ original items excluding j. For the comparable fractional problem, consider that if we remove weight w of one item j from the optimal load ($w < w_j$), the remaining load must be the most valuable load weighing at most $W - w$ pounds that the thief can take from the $n-1$ original items plus $w_j - w$ pounds of item j.

Although the problems are similar, the fractional knapsack problem is solvable by a **greedy strategy** whereas the 0-1 problem is not. To solve the fractional problem, we first compute the value per pound, v_i/w_i for each item. The thief begins by taking as much as possible from the item with the greatest value per pound. If the supply of that item is exhausted and he can still carry more, he takes as much as possible of the item with the next greatest value per pound, and so forth until he can't carry any more.

Example: Greedy isn't always optimal

Suppose $W = 50$ and there are three items with values and weights as shown in the table below.

	Item 1	*Item 2*	*Item 3*
Value of Item	\$60	\$100	\$120
Weight of Item	10	20	30
Value / Pound	6	5	4

The optimal solution for the 0-1 problem is to take items 2 and 3. However, if the greedy strategy is used, the thief would select items 1 and 2 and there would not be enough space leftover to include item 3. The optimal solution for the fractional problem is to take 10 lbs of item 1, 20 lbs of item 2 and 20 lbs of item 3.

Example: Capital Budgeting and Knapsacks

Reference: http://mat.gsia.cmu.edu/orclass/integer

Suppose we wish to invest \$14,000. We have identified four investment opportunities. Information about the four investments is presented in the table below.

	Investment 1	*Investment 2*	*Investment 3*	*Investment 4*
Net Present Value	\$8,000	\$11,000	\$6,000	\$4,000
Initial Investment	\$5,000	\$7,000	\$4,000	\$3,000
Value / \$ Invested	1.60	1.57	1.50	1.33

Into which investment should we place our money so as to maximize our total net present value?

We may setup this problem as a binary linear program (BLP). Let $x_j = 1$ if investment j is selected and 0 otherwise (o.w.). The formulation is

$$\begin{array}{rll} \max & 8x_1 + 11x_2 + 6x_3 + 4x_4 & \text{(max total net present value of investment in \$K)} \\ \text{s.t.} & 5x_1 + 7x_3 + 4x_3 + 3x_4 \leq 14 & \text{(total investment may not exceed \$14K)} \\ & x_j \in \{0,1\}, j = 1, \ldots, 4 & \text{each opportunity is either selected or rejected} \end{array}$$

If we **relax** the binary constraints on the variables, we obtain the following linear program.

$$\begin{array}{rll} \max & 8x_1 + 11x_2 + 6x_3 + 4x_4 & \text{(max total net present value of investment in \$K)} \\ \text{s.t.} & 5x_1 + 7x_3 + 4x_3 + 3x_4 \leq 14 & \text{(total investment may not exceed \$14K)} \\ & 0 \leq x_j \leq 1, j = 1, \ldots, 4 & \text{invest no more than the total requirement} \end{array}$$

This capital budgeting problem is a knapsack problem (why?). Thus, the knapsack problem has applications beyond optimizing the activities of thieves. Also observe that the greedy solution (for the fractional problem) is $x_1 = 1$, $x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$. The rounded solution $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$ is NOT optimal for the 0-1 problem. There is a better all-integer solution of $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$.

REMEMBER: the greedy solution is ALWAYS optimal for the fractional knapsack problem. The same cannot be said for the 0-1 knapsack problem.