Linear and Quadratic Knapsack Problems

Professor M Ali, Wits

The Knapsack Problem (KP) is considered to be a combinatorial optimization problem. A Knapsack model serves as an abstract model with broad spectrum applications such as: Resource allocation problems, Portfolio optimization, Cargo-loading problems and Cutting stock problems. Two problems are considered: Linear KP and the Quadratic KP (QKP) problem. In linear KP the objective function and constraint(s) are linear. QKP has quadratic objective function and it is an extension of the linear Knapsack problem where there are additional terms in the objective function that describes extra profit gained from choosing a particular combination of items.

Problem 1: Linear Knapsack Problem: Consider the following pairs
$(v_i, w_i) = \{(2, 7), (6, 3), (8, 3), (7, 5), (3, 4), (4, 7), (6, 5),$
$(5, 4), (10, 15), (9, 10), (8, 17), (11, 3), (12, 6), (15, 11), (6, 6),$
$(8, 14), (13, 4), (14, 8), (15, 9), (16, 10), (13, 14), (14, 17), (15, 9), (26, 24),$
$(13, 11), (9, 17), (25, 12), (26, 14)\}$ with total capacity $W = 30$.

Use the following two algorithm to solve the above problem:
Greedy Algorithm:

1. Identify the available items with their weights and values and take note of the maximum capacity of the bag.
2. Use of a score or efficiency function, i.e. the profit to weight ratio: $\frac{v_i}{w_i}$
3. Sort the items non-increasingly according to the efficiency function.
4. Add into knapsack the items with the highest score, taking note of their accumulative weights until no item can be added.
5. Return the set of items that satisfies the weight limit and yields maximum profit.

Polynomial Time Approximation Algorithm (PTAS):

1. Consider all sets of up to at most k items

$$S = \{F \subset \{1, 2, \cdots, 28\} : |F| = k, w(F) < W\} \tag{1}$$

2. For all F in S
   - Pack F into the knapsack
   - Greedily fill the remaining capacity
   - End
3. Return highest valued item combination set

Note $|F|$ denotes the number of item in the set F; $w(F)$ denotes the total weights of the items in F; Use $k = 10$; when creating S do not take subsets F with less than 3 items.

Problem 2: Consider a QKP problem with $n = 15$ items with
$v_i = 7, 6, 13, 16, 5, 10, 9, 23, 18, 12, 9, 22, 17, 32, 8;$
$w_i = 13, 14, 14, 15, 15, 9, 26, 24, 13, 11, 9, 12, 25, 12, 26$, $W = 50$
and $p_{ij} = \{(12, 7, 6, 13, 8, 11, 7, 15, 23, 14, 15, 17, 9, 15, 15,$
$13, 10, 15, 9, 10, 8, 17, 11, 13, 12, 16, 15, 11, 16, 6,$
$8, 14, 13, 4, 14, 8, 15, 9, 16, 10, 13, 14, 14, 17, 15, 14, 6, 24, 13, 4, 9, 7, 25, 12, 6,$
$6, 16, 10, 15, 14, 2, 13, 12, 16, 9, 11, 23, 10, 21, 8$
$18, 4, 13, 14, 14, 17, 15, 9, 16, 12, 3, 14, 14, 27, 15, 16, 13, 14, 7, 17, 28, 5, 19, 6, 18, 13, 4, 13, 16, 11,$
$19, 13, 15, 12, 16)\}$

Use the following greedy algorithm and get the results
Greedy algorithm for QKP:

1. Sample k (say $k = 7$) items from the set of 15 items.
2. Obtain a set of all pairs from the k items.
3. . Sort the items non-increasingly according to the efficiency function $\frac{p_{ij}}{w_i+w_j}$
4. Add into knapsack the pair of items with the highest score, ensuring that the accumulated weight does not exceed the maximum capacity.
5. Repeat steps 1 through 4 until pairs can no longer be added.
6. Fill remaining capacity with singleton items, using the previous greedy approach for Problem 1.