0/1 knapsack problem using Branch and Bound

The 0/1 knapsack problem

- Positive integer $P_1, P_2, \ldots, P_n$ (profit)

  $W_1, W_2, \ldots, W_n$ (weight)

  M (capacity)

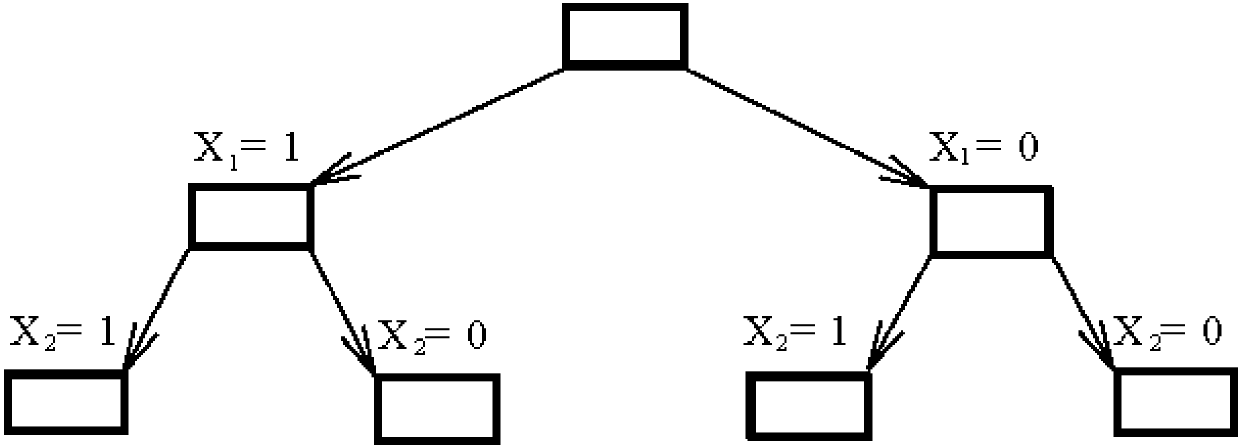
$$\text{maximize} \quad \sum_{i=1}^{n} P_i X_i$$

$$\text{subject to} \quad \sum_{i=1}^{n} W_i X_i \leq M \qquad X_i = 0 \text{ or } 1,\ i = 1, \ldots, n.$$

The problem is modified:

$$\text{minimize} \quad -\sum_{i=1}^{n} P_i X_i$$

The 0/1 knapsack problem



The Branching Mechanism in the Branch-and-Bound Strategy to Solve 0/1 Knapsack Problem.

How to find the upper bound?

- Ans: by quickly finding a feasible solution in a greedy manner: starting from the smallest available i, scanning towards the largest i's until M is exceeded. The upper bound can be calculated.

How to find the ranking Function

Ans: by relaxing our restriction from $X_i = 0$ or 1 to $0 \leq X_i \leq 1$ (knapsack problem)

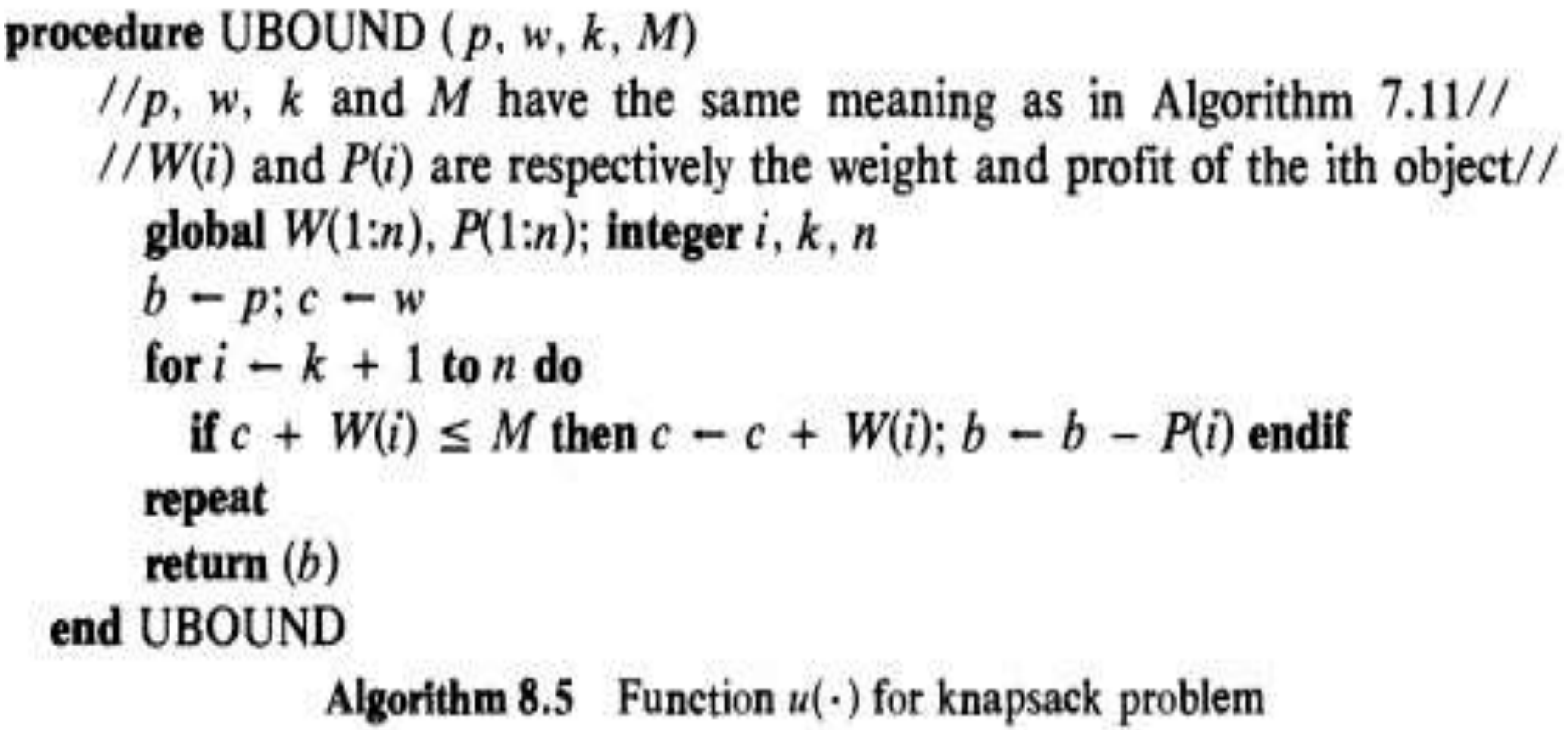
Let $-\sum_{i=1}^{n} P_i X_i$ be an optimal solution for 0/1 knapsack problem and $-\sum_{i=1}^{n} P_i X'_i$ be an optimal solution for fractional knapsack problem. Let $Y = -\sum_{i=1}^{n} P_i X_i$, $Y' = -\sum_{i=1}^{n} P_i X'_i$.

$\Rightarrow Y' \leq Y$

How to expand the tree?

- By the best-first search scheme
- That is, by expanding the node with the best lower bound. If two nodes have the same lower bounds, expand the node with the lower upper bound.

0/1 Knapsack algorithm using BB



```
procedure UBOUND (p, w, k, M)
   //p, w, k and M have the same meaning as in Algorithm 7.11//
   //W(i) and P(i) are respectively the weight and profit of the ith object//
     global W(1:n), P(1:n); integer i, k, n
     b ← p; c ← w
     for i ← k + 1 to n do
        if c + W(i) ≤ M then c ← c + W(i); b ← b − P(i) endif
     repeat
     return (b)
 end UBOUND
```

Algorithm 8.5 Function $u(\cdot)$ for knapsack problem

0/1 Knapsack Example using LCBB (Least Cost)

□ Example (LCBB)

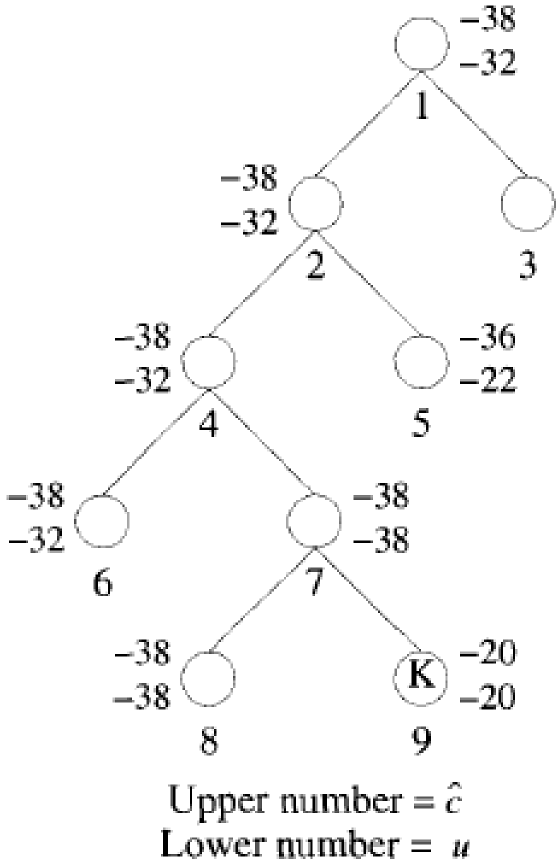
□ Consider the knapsack instance:

□ n = 4;

□(pi, p2, □p3, p4) = (10, 10, 12, 18);

□ (wi. w2, w3, w4) = (2, 4, 6, 9) and □M = 15

0/1 Knapsack State Space tree of Example using LCBB



LC branch-and-bound tree

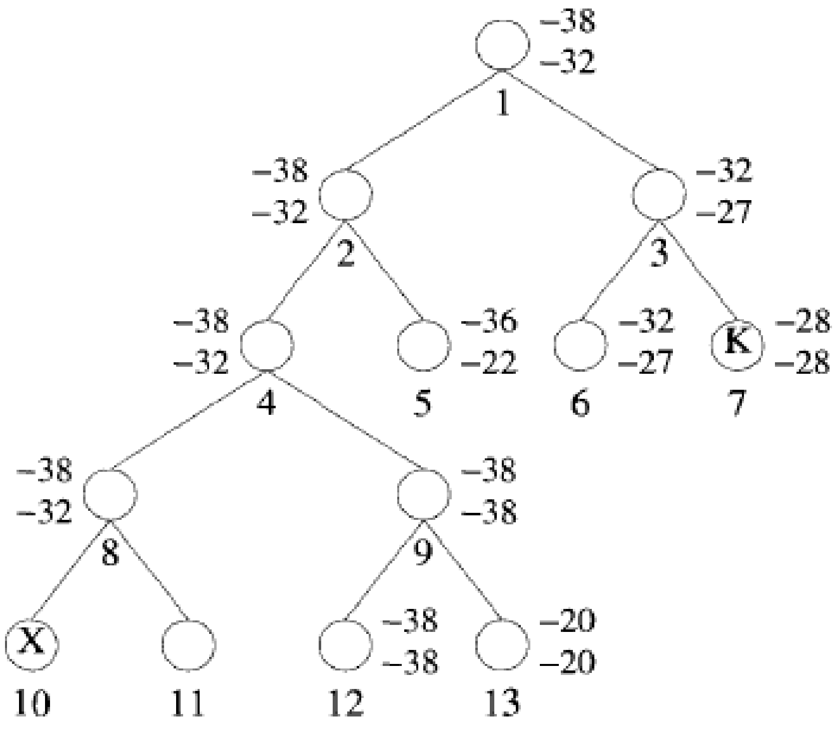
‘u’ without fraction
‘c’ with fractions

(pi, p2,p3, p4) = (10, 10, 12, 18)

(wi. w2, w3, w4) = (2, 4, 6, 9)

M = 15

0/1 Knapsack State Space tree of Example using FIFO BB



upper number = $\hat{c}$
lower number = u

FIFO branch-and-bound tree