

Branch and Bound

Branching

Splitting the problem into number of subproblems.

Bounding

Finding an optimistic estimate of the best solution to the subproblem.

Consider the example -

maximizing $45x_1 + 48x_2 + 35x_3$

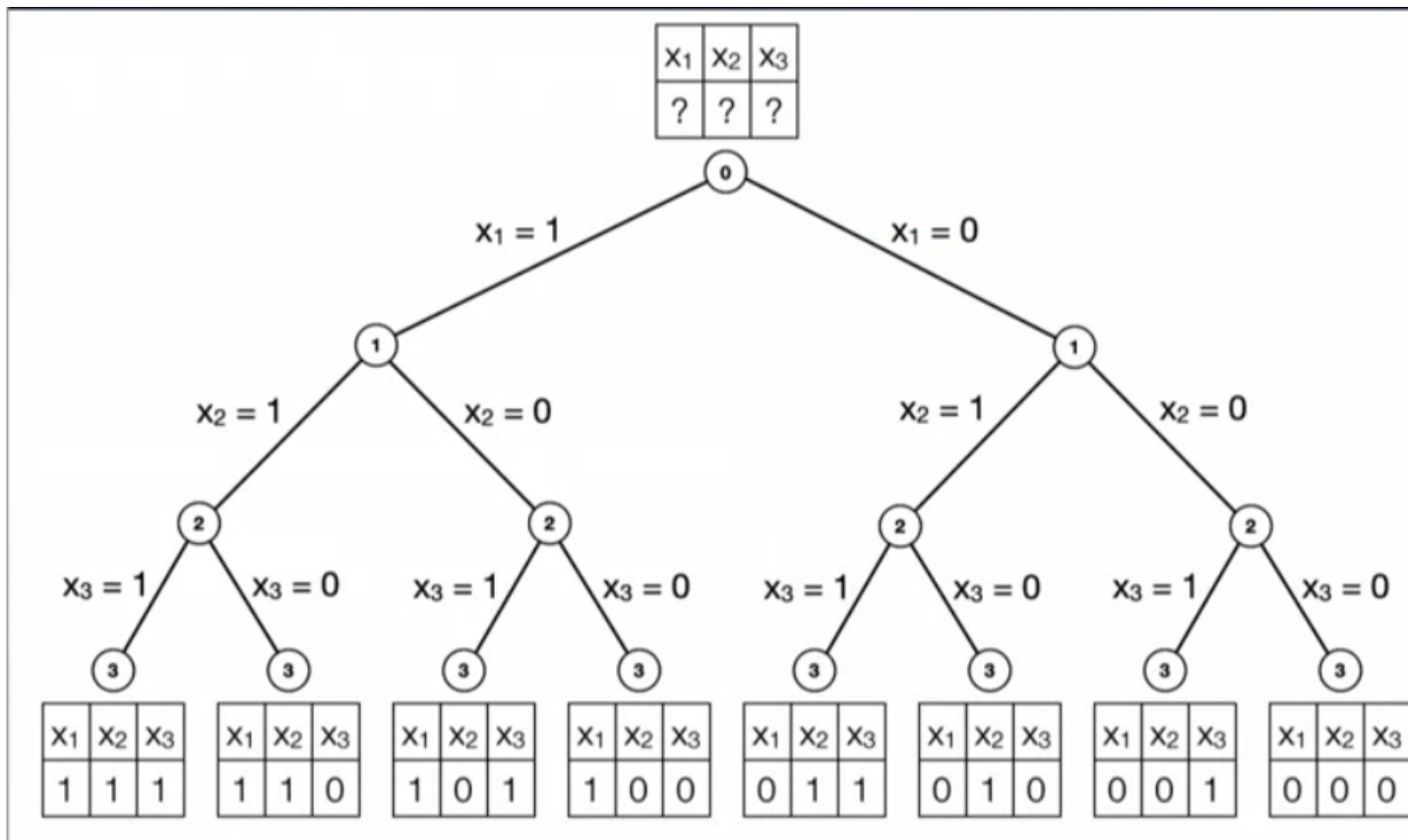
subject to $5x_1 + 8x_2 + 3x_3 \leq 10$

$$x_i \in \{0, 1\}$$

and the value will be decided depending upon whether the item is chosen or not. Now relaxing the capacity constraint -

We initially consider the value of knapsack which is zero, the room in knapsack which is 10 and estimate is 128 which is the maximum value we can get (when all the items are selected).

Now we go on updating the value, room left and estimate by selecting or excluding a particular item and we also build a tree structure where there is a split at each node when the next item is either included (denoted by $x_i = 1$) or excluded (denoted by $x_i = 0$).



Let us consider another relaxation method, where we consider that it is possible to include an item fractionally, that is $0 \leq x_i \leq 1, i \in \{1, 2, 3\}$. This method is basically linear relaxation of knapsack, which means we are relaxing the integrality constraint.

For this method, we order the items by decreasing value of V_i/W_i and then we select the items while the capacity is not exhausted and finally select a fraction of last item.

In the above example $V_1/W_1=9, V_2/W_2=6, V_3/W_3=11.7$. Thus we select item 3 and 1, and 1/4 of item 2 which gives us estimation as 92. Thus the maximum value that we can get by this

optimistic evaluation is 92. And now this value is used as initial estimate for the Branch and Bound.

But in practice, we can not take the items fractionally and so this relaxation can not be considered.

Now that we have a tree which has been built by taking in consideration the inclusion and exclusion of items, we can explore this graph in different fashions

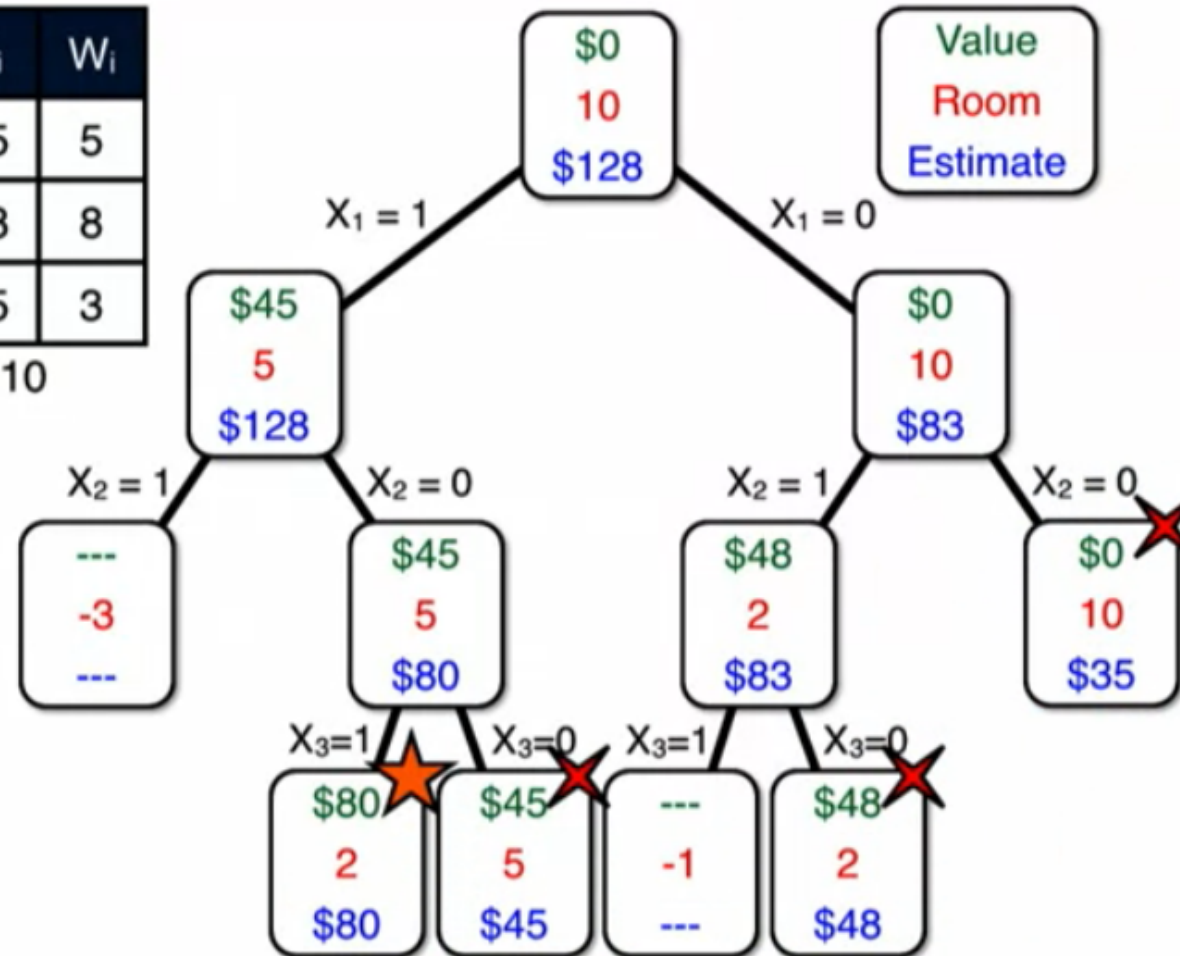
- **Depth-first:**

This search prunes when a node estimation is worse than the **best found solution till now**. That is, the tree is traversed in a depth first manner and we stop the search when we hit a node having lower estimate value than best found solution.

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

$K = 10$



8

- **Best-first:**

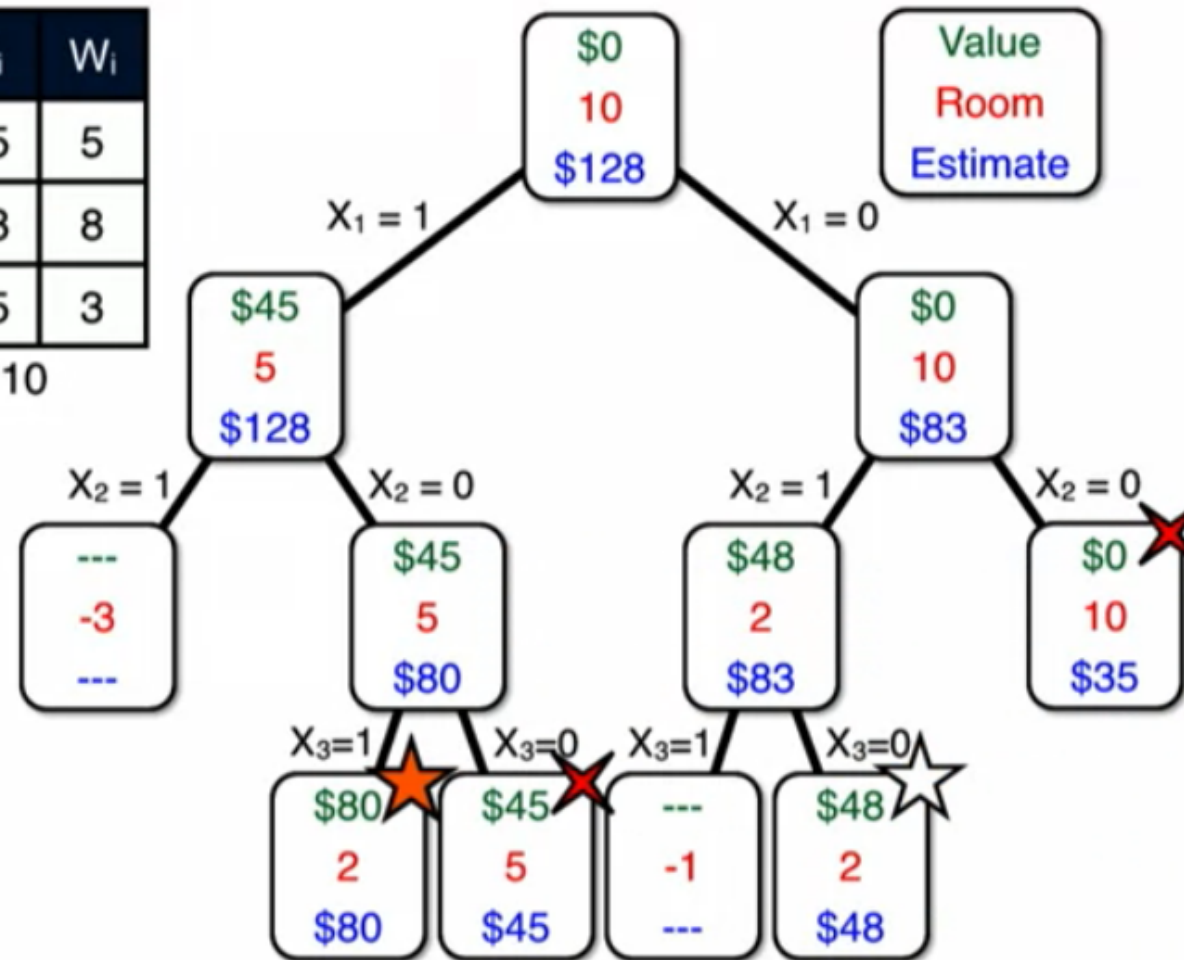
In this case we select the node with **best estimation**. That is, each time we select an item or

reject it, then look at the estimate value and continue traversing the part of the graph which has higher estimate value. Thus, the search prunes when all the nodes are worse than a found solution.

Best-First Branch and Bound

i	V _i	W _i
1	45	5
2	48	8
3	35	3

K = 10



8

- **Limited-Discrepancy:**

We follow a greedy heuristic and then move away from it in a systematic fashion. Let us

assume that the search tree is binary and following the heuristic means branching left and branching right means the heuristic is wrong. Thus here we are exploring the search space in increasing order of mistakes and trusting the heuristic less and less as we move on, (Where mistake means not following the heuristic which means we branch to the right). That is initially we start with zero mistakes and go on increasing the number by 1 each time. This search prunes like the Best-first search.

LDS Branch and Bound

