UNIT 5: Branch-and-Bound

0/1 Knapsack problem

Branch and Bound:

- proposed by **Ailsa Land** and **Alison Doig** during their discrete programming research at the London School of Economics in 1960
- algorithm design paradigm for discrete and combinatorial **optimization problems**, as well as mathematical optimization
- the most commonly used tool for solving NP-hard optimization problems

Branch and Bound: Idea

- systematic enumeration of candidate solutions using state space search

- **Limitation:** depends on efficient estimation of the lower and upper bounds of regions/branches of the search space. If no bounds are available, the algorithm degenerates to an exhaustive search

Branch and Bound: terminating a search path in state space tree

1. The value of the node's bound is not better than the value of the best solution seen so far **OR**
2. The node represents no feasible solutions because the constraints of the problem are already violated **OR**
3. The subset of feasible solutions represented by the node consists of a single point (and hence no further choices can be made)

Branch and Bound: Applications

used for solving a number of NP-hard problems:

- Integer programming
- Nonlinear programming
- Travelling salesman problem (TSP)
- Quadratic assignment problem (QAP)
- Maximum satisfiability problem (MAX-SAT)
- Nearest neighbor search
- Flow shop scheduling
- Cutting stock problem
- Computational phylogenetics
- Set inversion
- Parameter estimation
- 0/1 knapsack problem
- Set cover problem
- Feature selection in machine learning
- Structured prediction in computer vision

Knapsack problem

Given n items of known weights w1, ... , wn and values v1, ... , vn, and a knapsack of capacity W, find the most valuable subset of the items that fit into the knapsack.

Branch and Bound: 0/1 Knapsack problem

- order the items of a given instance in descending order by their value-to-weight ratios. (first item gives the best payoff per weight unit and the last one gives the worst payoff per weight unit, with ties resolved arbitrarily)

$$v_1/w_1 \geq v_2/w_2 \geq \ldots \geq v_n/w_n$$

- Compute the upper bound

$$\boxed{ub = v + (W - w)(v_{i+1}/w_{i+1})}$$

item	weight (kg)	Profit (Rs.)
1	7	42
2	3	12
3	4	40
4	5	25

Knapsack capacity = 10 kg

$$v_1/w_1 \geq v_2/w_2 \geq \ldots \geq v_n/w_n$$

item	weight (kg)	Profit (Rs.)	
1	7	42	$\rightarrow$ 42/7 = 6
2	3	12	$\rightarrow$ 12/3 = 4
3	4	40	$\rightarrow$ 40/4 =10
4	5	25	$\rightarrow$ 25/5 =5

Knapsack capacity = 10 kg

$\Rightarrow$

item	weight (kg)	Profit (Rs.)
3	4	40
1	7	42
4	5	25
2	3	12

Example:

Solve

item	weight	value
1	4	$40
2	7	$42
3	5	$25
4	3	$12

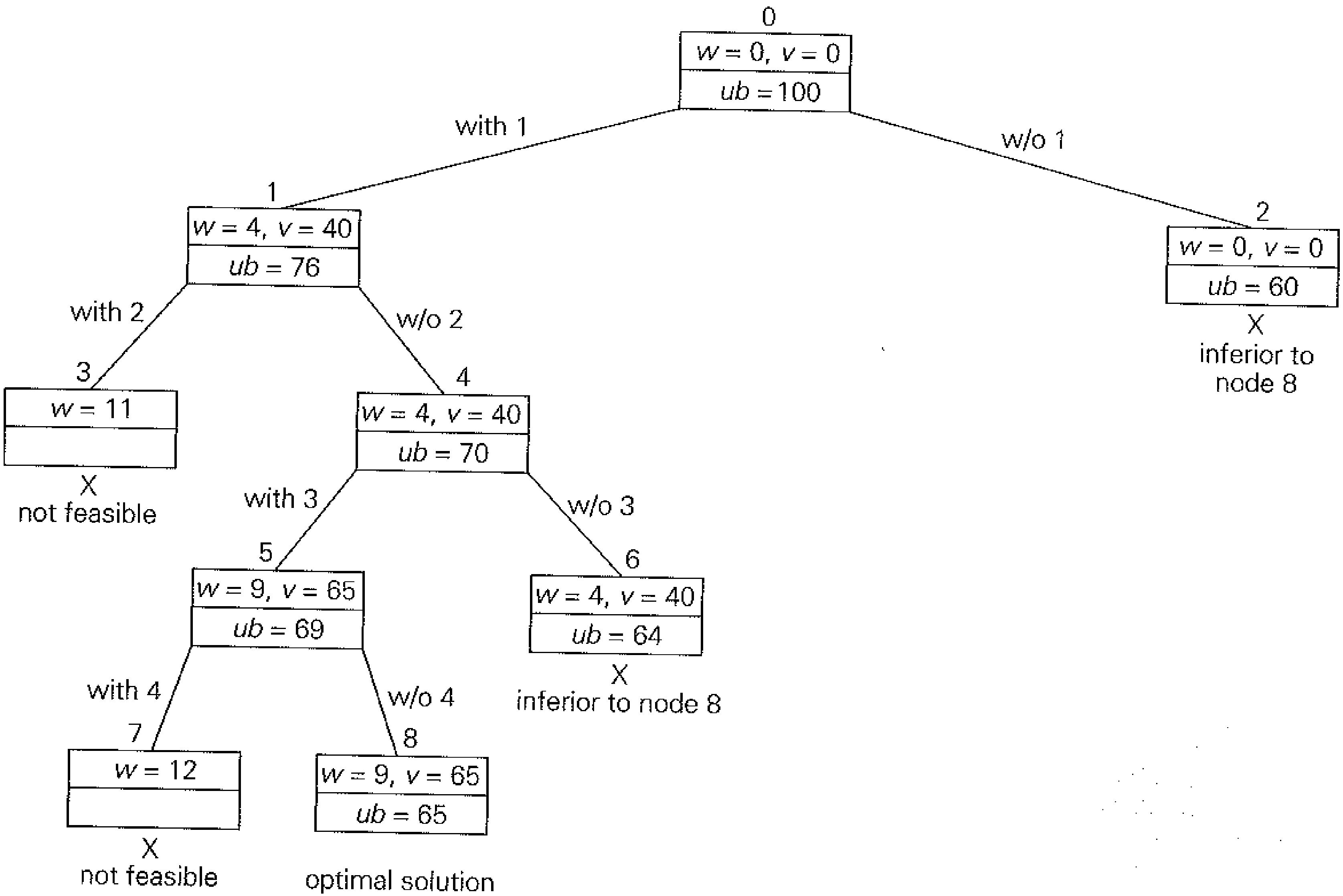
Knapsack capacity = 10 kg

$$v_1/w_1 \geq v_2/w_2 \geq \ldots \geq v_n/w_n$$

$$ub = v + (W - w)(v_{i+1}/w_{i+1})$$

item	weight	value	$\frac{\text{value}}{\text{weight}}$
1	4	$40	10
2	7	$42	6
3	5	$25	5
4	3	$12	4

Knapsack solution – Branch and Bound



0
w = 0, v = 0
ub = 100
with 1
w/o 1
1
w = 4, v = 40
ub = 76
2
w = 0, v = 0
ub = 60
X
inferior to node 8
with 2
w/o 2
3
w = 11
X
not feasible
4
w = 4, v = 40
ub = 70
with 3
w/o 3
5
w = 9, v = 65
ub = 69
6
w = 4, v = 40
ub = 64
X
inferior to node 8
with 4
w/o 4
7
w = 12
X
not feasible
8
w = 9, v = 65
ub = 65
optimal solution

Let’s check our understanding…

Solve the following instance of the knapsack problem by the branch-and bound algorithm.

item	weight	value
1	2	\$12
2	1	\$10
3	3	\$20
4	2	\$15

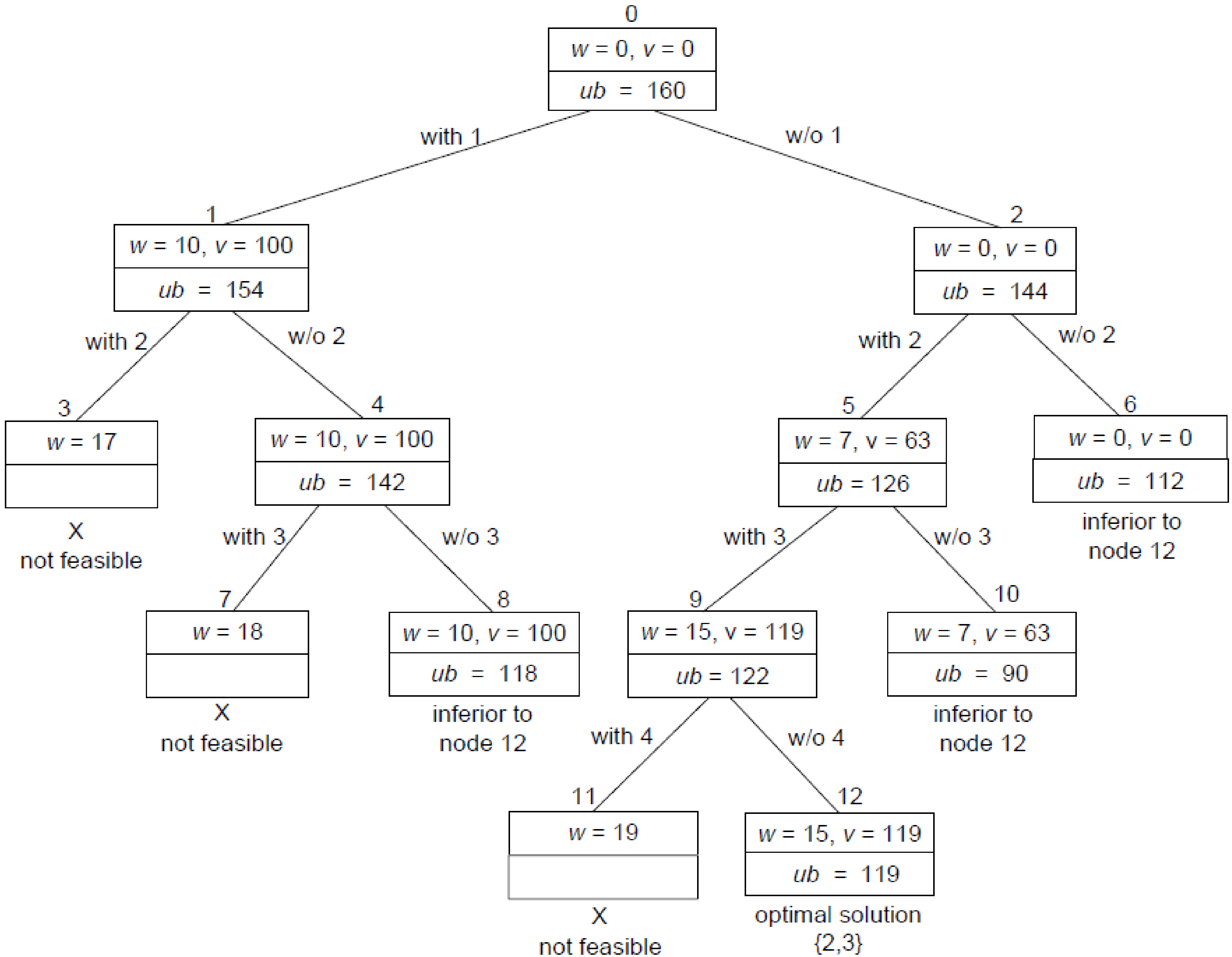
capacity $W = 5$

Let’s check our understanding…

Solve the following instance of the knapsack problem by the branch-and bound algorithm.

item	weight	value
1	10	\$100
2	7	\$63
3	8	\$56
4	4	\$12

$W = 16$



The found optimal solution is {item 2, item 3} of value $119.

Next session…

Branch-and-Bound :

Travelling salesperson problem