UNIT 5: Branch-and-Bound

Assignment 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
- Best first – branch and bound

- **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

Assignment problem

Given n people who need to be assigned to execute n jobs, one person per job, find an assignment with the minimum total cost.

Example:

Solve

	Job 1	Job 2	Job 3	Job 4
Person 1	9	2	7	8
Person 2	6	4	3	7
Person 3	5	8	1	8
Person 4	7	6	9	4

Exhaustive search approach

$$C = \begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix}$$

$<1, 2, 3, 4>$	cost $= 9 + 4 + 1 + 4 = 18$
$<1, 2, 4, 3>$	cost $= 9 + 4 + 8 + 9 = 30$
$<1, 3, 2, 4>$	cost $= 9 + 3 + 8 + 4 = 24$
$<1, 3, 4, 2>$	cost $= 9 + 3 + 8 + 6 = 26$
$<1, 4, 2, 3>$	cost $= 9 + 7 + 8 + 9 = 33$
$<1, 4, 3, 2>$	cost $= 9 + 7 + 1 + 6 = 23$

etc.

Branch and Bound: Assignment problem

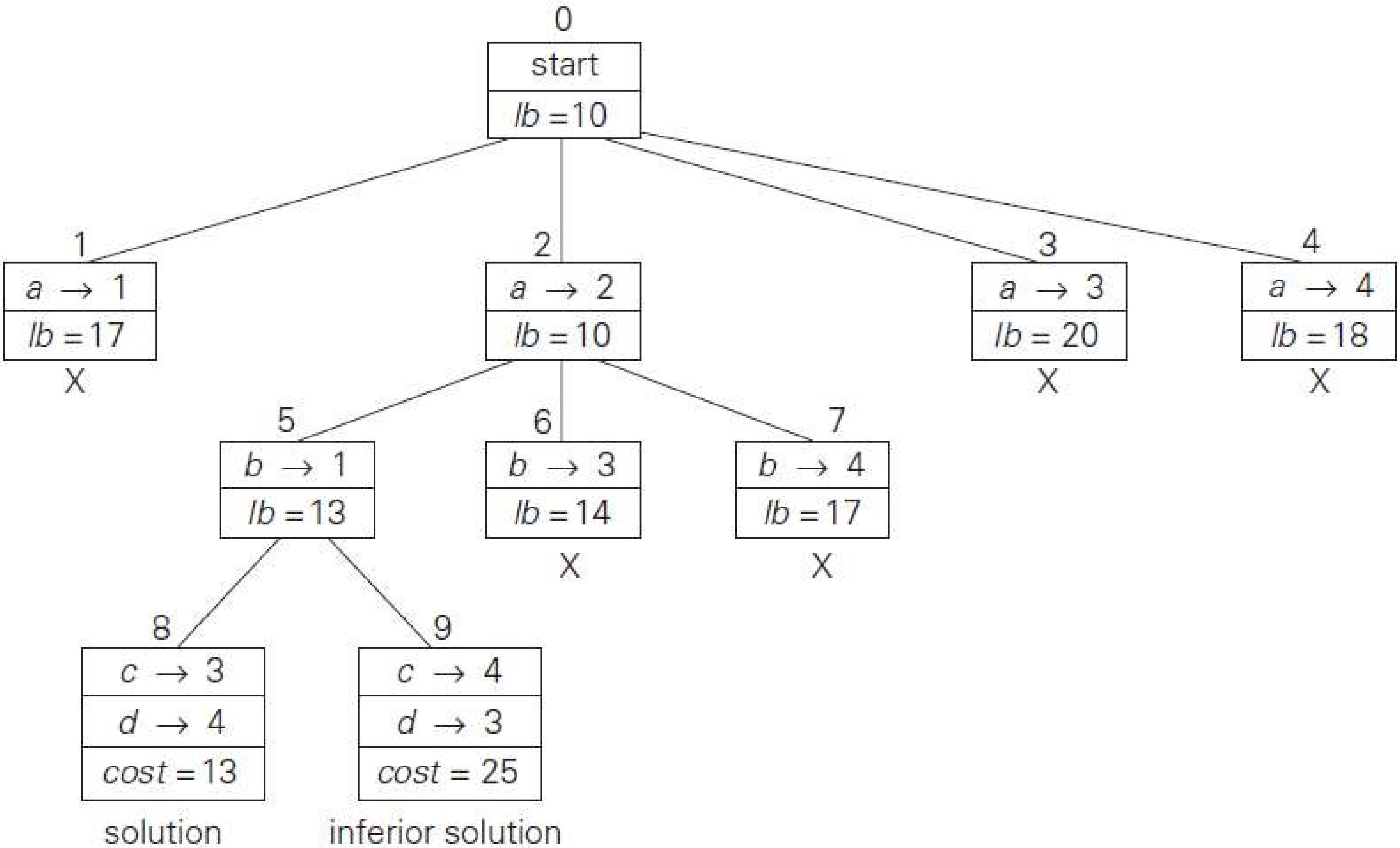
- Compute lower bound on the cost of an optimal selection without actually solving the problem

$$C = \begin{array}{cccc} \text{job 1} & \text{job 2} & \text{job 3} & \text{job 4} \end{array}$$

$$C = \begin{bmatrix} 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 8 \\ 7 & 6 & 9 & 4 \end{bmatrix}$$

lb = 2 + 3 + 1 + 4

= 10



0
start
lb = 10
1
a → 1
lb = 17
X
2
a → 2
lb = 10
3
a → 3
lb = 20
X
4
a → 4
lb = 18
X
5
b → 1
lb = 13
6
b → 3
lb = 14
X
7
b → 4
lb = 17
X
8
c → 3
d → 4
cost = 13
solution
9
c → 4
d → 3
cost = 25
inferior solution

Note:

There is a polynomial-time algorithm for assignment problem called the **Hungarian method**

In the light of this efficient algorithm, solving the assignment problem by branch-and-bound should be considered a **convenient educational device rather than a practical recommendation**.