UE19CS251
Design and Analysis of Algorithms
Unit 5: Limitations of Algorithmic Power and Coping with the Limitations

Branch and Bound

PES University

Outline

Concepts covered

- Backtracking
  - General Approach
  - Knapsack Problem
  - Assignment Problem
  - Travelling Salesman Problem

1 Introduction

- An enhancement of backtracking
- Applicable to optimization problems
- For each node (partial solution) of a state-space tree, computes a bound on the value of the objective function for all descendants of the node (extensions of the partial solution)
- Uses the bound for:
  - ruling out certain nodes as "nonpromising" to prune the tree (if a node's bound is not better than the best solution seen so far)
  - guiding the search through state-space

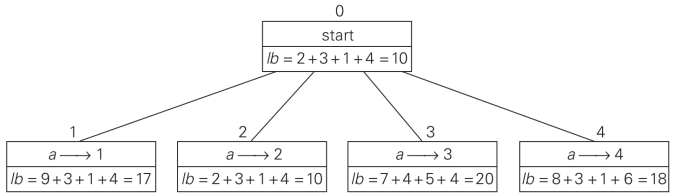
2 Example: Assignment Problem

- Select one element in each row of the cost matrix C so that:
  - no two selected elements are in the same column
  - the sum is minimized
- Example

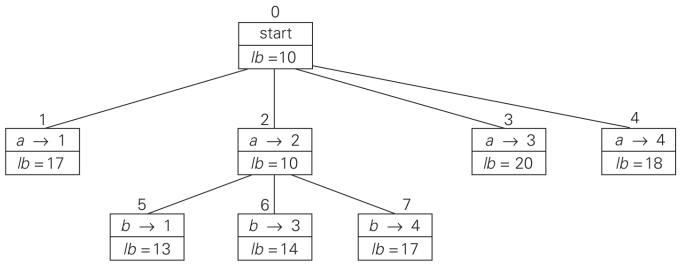
	Job 1	Job 2	Job 3	Job 4
Person a	9	2	7	8
Person b	6	4	3	7
Person c	5	8	1	8
Person d	7	6	9	4

- Lower bound (sum of smallest elements in each row): $2+3+1+4 = 10$
- Best-first branch-and-bound variation: Generate all the children of the most promising node

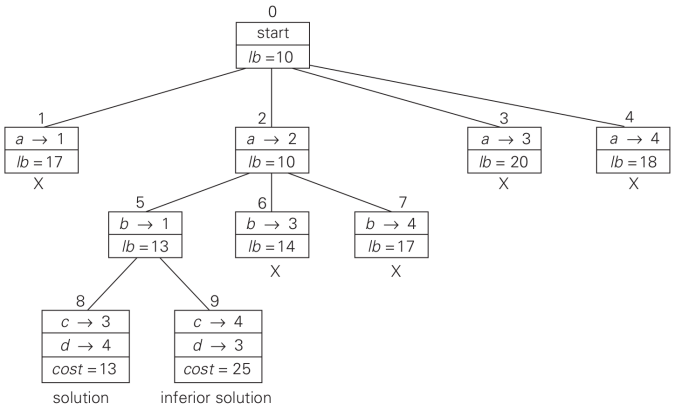
3 Example: First two levels of the state-space tree



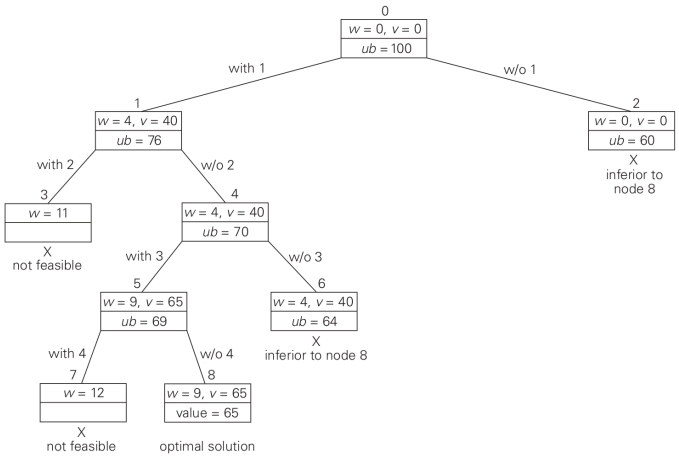
4 Example: First three levels of the state-space tree



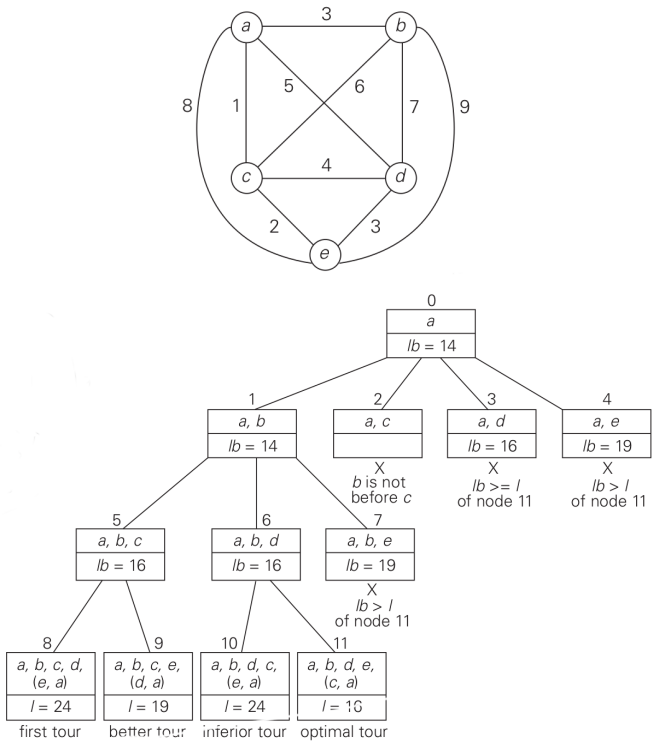
5 Example: Complete state-space tree



6 Example: Knapsack Problem



7 Example: Traveling Salesman Problem



8 Think About It

- What data structure would you use to keep track of live nodes in a best-first branch-and-bound algorithm?
- Solve the assignment problem by the best-first branch-and-bound algorithm with the bounding function based on matrix columns rather than rows