Design and Analysis of Algorithms

L36: Branch and Bound

Job Assignment Problem
TSP Problem

Dr. Ram P Rustagi
Sem IV (2020-Even)
Dept of CSE, KSIT
rprustagi@ksit.edu.in

Resources

- Text book 1: Levitin
  - Sec *12.2*
- Text book 2: Horowitz
  - Sec 8.2
  - R1: Introduction to Algorithms
    - Cormen et al.

Example use cases

- Job Assignment problem
- Traveling Salesperson Problem
- `0-1` Knapsack problem

Exact Solution Strategies

- Exhaustive search (brute force)
  - Search for all possible combinations (exponential time)
    - e.g. $O(k^n)$, $O(n!)$, $O(n^n)$
  - Useful only for small instances
- *Dynamic programming*
  - Applicable to some problems
    - Where problem can be recursively mapped to smaller other problems
- *Backtracking*
  - Build a state space tree
  - Eliminates unnecessary cases from consideration
  - Solutions may still take exponential time
- *Branch-and-bound*
  - Further refines backtracking for optimization problems

Branch and Bound

- Additional mechanisms in addition to backtracking
  - Provide a bound on the best value of objective function for every node of the state-space tree
  - The value of best solution so far obtained at
    - Current node of state space tree
- Approach
  - Compare the node's bound value with the value of best solution seen so far.
    - If the bound is not better, terminate the search (prune the solution)
      - It is not smaller than the best sol^n in a minimization problem
      - It is not greater than the best sol^n in a maximization problem

Branch and Bound

- Termination criteria of the search path in state space tree using branch-n-bound algo:
  - The value of node's bound is worse than the value of best solution seen so far
  - The node represents no feasible solution because of the constraints of the problem are already violated
  - The subset of feasible solutions represented by the node consists of a single point
    - i.e. reached the end of solution and no more choices
    - Compare the value of objective function with that the best solution seen so far
      - Update the latter if former is better than latter.

Assignment Problem

- Consider a problem of assigning `n` jobs to `n` people so that cost is minimised.
  – One person does one job only, no more, no less
  – The cost of each job done by each person is given
  – Represented in a matrix.
- Consider an example below for job assignment costs of 4 persons

	J_1	J_2	J_3	J_4
P_a	9	2	7	8
P_b	6	4	3	7
P_c	5	8	1	8
P_d	7	6	9	4

Assignment Problem

	J_1	J_2	J_3	J_4
P_a	9	2	7	8
P_b	6	4	3	7
P_c	5	8	1	8
P_d	7	6	9	4

- Problem can be stated as follows
  - Select one element in each row, such that
    - No two selected elements are in same column/row
    - Their sum is smallest possible.
- Solution with branch and bound
  - Consider the lowest possible sum
  - Take the lowest element in each row.
  - This may not be feasible but can act as lower bound
    - Two elements may belong to same column
  - Smallest possible values for above example

$(P_a(J_2)=2)+(P_b(J_3)=3)+(P_c(J_3)=1)+(P_d(J_4)=4)$
$=2+3+1+4=10$

    - It is not legitimate though
      - (P_b and P_c assgined same job J_3)

Assignment Problem

	J_1	J_2	J_3	J_4
P_a	9	2	7	8
P_b	6	4	3	7
P_c	5	8	1	8
P_d	7	6	9	4

- Backtracking:
  - Generate a child of last promising node
  - i.e. last active node (called E-node)
- Branch and Bound approach
  - Generate the child of most promising node
    - Among non-terminated live leaves in current tree
    - Achieved by comparing lower bounds of all live nodes
  - Intuitively, it is better to consider a node with best bound as most promising
    - It may not lead to optimal solution
    - It may lie in other branch of the tree.

Assignment Problem

	J_1	J_2	J_3	J_4
P_a	9	2	7	8
P_b	6	4	3	7
P_c	5	8	1	8
P_d	7	6	9	4

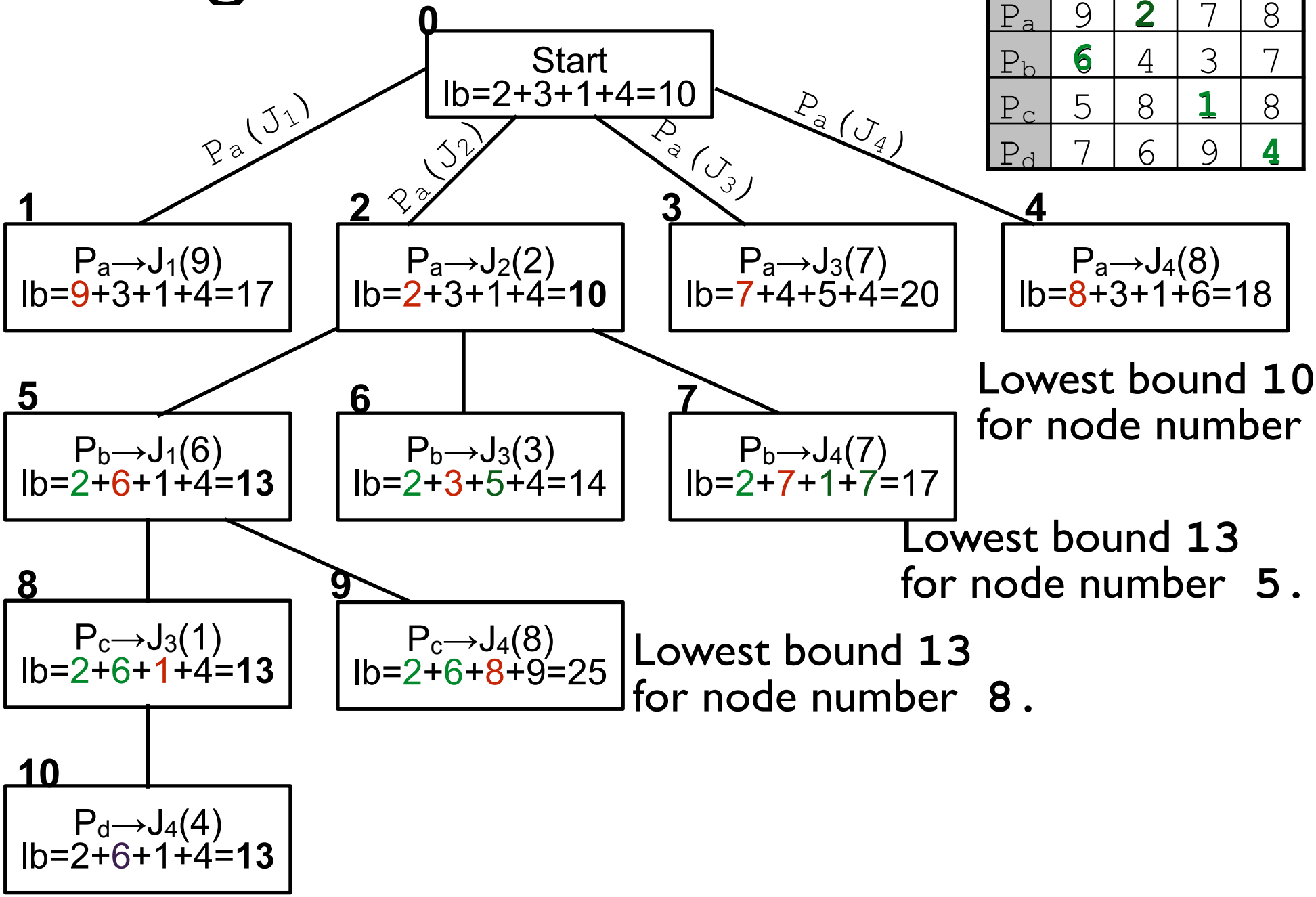
Start
lb=2+3+1+4=10

- Start at root
  - No elements selected from matrix
  - Lower bound lb=10
- It (root) has 4 live nodes for P_a
  $J_1=9$, $J_2=2$, $J_3=7$, $J_4=8$,
  - Lower bounds for each of these for P_a are
    - LB for $(J_1=9)$ $=9+3+1+4=17$

  $P_a(J_1=9)+P_b(J_3=3)+P_c(J_3=1)+P_d(J_4=4)$
    - LB for $(J_2=2)$ $=2+3+1+4=10$
    - LB for $(J_3=7)$ $=7+4+5+4=20$
    - LB for $(J_4=8)$ $=8+3+1+6=18$
- Most promising node for P_a is $J_2=2$.
  - Explore this node further.

Assignment Problem

	J_1	J_2	J_3	J_4
P_a	9	**2**	7	8
P_b	**6**	4	3	7
P_c	5	8	**1**	8
P_d	7	6	9	**4**



Lowest bound 10 for node number 2.

Lowest bound 13 for node number 5.

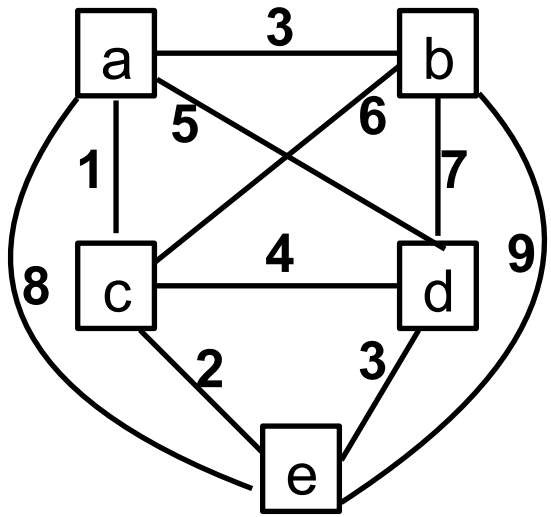
Lowest bound 13 for node number 8.

Traveling Salesperson Problem

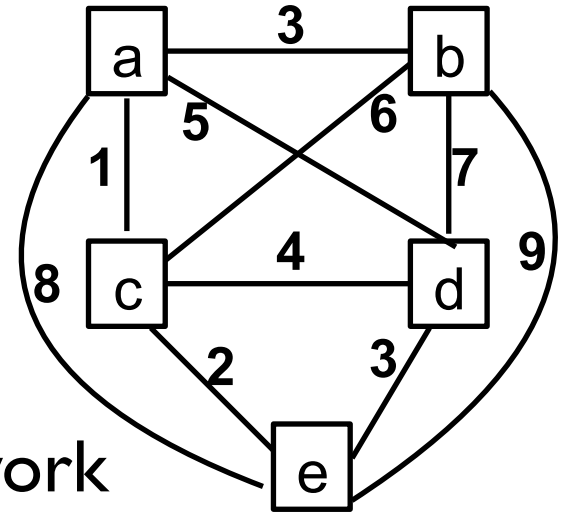
- BB approach: define a lower bound
- Simple approach:
  - Take the lowest edge cost
  - Multiply it by number of nodes
    - `LB=1*5=5`
- More informative but less obvious
  - Does not require much computation too
  - For each node, find two nearest nodes
    - Find the average of two
  - Sum this average (ceiling) for all nodes

`LB=((1+3)+(3+6)+(1+2)+(3+4)+(2+3))/2=14`

- When any tour includes a particular edge,
  - Update the lower bound accordingly
    - Using the included edge



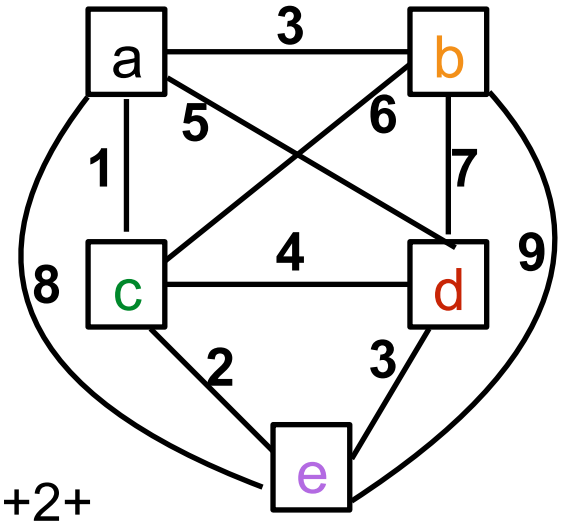
Traveling Salesperson Problem



- When any tour includes a particular edge,
  - Compute the lower bound using that edge.
- Example: Consider `(a,d)` is included.

  `LB=⌈((5+1)+(3+6)+(1+2)+`
  `(5+3)+(2+3))/2⌉=⌈31/2⌉=16`
- Further, to reduce the amount of potential work
  - We can consider that tour starts at node `a`, and
  - Since graph is undirected, impose the restriction
    - Generate tours in which `b` appears before `c`.
  - After visiting `n-1 (=4)` nodes,
    - Tour has to visit the last unvisited node, and
    - Return to the starting node.

Traveling Salesperson Problem



0
a
lb=14

(1+3+3+6+1+2+
3+4+2+3)/2=14

(1+5+3+6+1+2+
3+5+2+3)/2=16

(3+1+3+6+
1+2+3+4+
2+3)/2=14

1
a,b
lb=14

2
a,c
lb=14
X

b is after c

3
a,d
lb=16

lb≥node 11

4
a,e
lb=19

(1+8+3+6+1+2+
3+4+2+8)/2=19

lb≥node 11

(3+1+3+6+6+1+
3+4+2+3)/2=16

(3+1+3+7+1+2+
7+3+2+3)/2=16

(3+1+3+9+1+2+
3+4+9+2)/2=19

5
a,b,c
lb=16

6
a,b,d
lb=16

7
a,b,e
lb=19

a,b,c,d (e,a)
(3+8+3+6+6+4+
4+3+3+8)/2=24

a,b,c,e (d,a)
(3+5+3+6+6+2+
2+3+3+5)/2=19

a,b,d,c (e,a)
(3+8+3+7+7+4+
4+2+2+8)/2=24

a,b,d,e (c,a)
(3+1+3+7+7+3+
3+2+2+1)/2=16

8
a,b,c,d
(e,a)
lb=24

9
a,b,c,e
(d,a)
lb=19

10
a,b,d,c
(e,a)
lb=24

11
a,b,d,e
(c,a)
lb=16

1st tour

better tour

inferior tour

optimal tour

Branch and Bound

- Finding a good bound function is a challenging task
  – May not be always easy to find one
- Bounding function should be easy to compute
- It should not be too simple
  – It may fail to prune the many branches of state space tree as soon as possible
- Finding the balance between two requirements (Easy to compute, and not too simplistic)
  – May require intensive experimentation
  – With a wide variety of problem in question

Summary

- Assignment problem
- Traveling Salesperson Problem