TSP using Branch and Bound Searching Strategies

Introduction :The traveling salesperson optimization problem

- Given a graph, the TSP Optimization problem is to find a tour, starting from any vertex, visiting every other vertex and returning to the starting vertex, with minimal cost.
- It is NP-hard.
- We try to avoid n! exhaustive search by the branch-and-bound technique on the average case. (Recall that $O(n!)>O(2^n)$.)

The traveling salesperson optimization problem

- E.g. A Cost Matrix for a Traveling Salesperson Problem.

i \ j	1	2	3	4	5	6	7
1	∞	3	93	13	33	9	57
2	4	∞	77	42	21	16	34
3	45	17	∞	36	16	28	25
4	39	90	80	∞	56	7	91
5	28	46	88	33	∞	25	57
6	3	88	18	46	92	∞	7
7	44	26	33	27	84	39	∞

The basic idea

- There is a way to split the solution space (branch)
- There is a way to predict a lower bound for a class of solutions. There is also a way to find a upper bound of an optimal solution. If the lower bound of a solution exceeds the upper bound, this solution cannot be optimal and thus we should terminate the branching associated with this solution.

Splitting

- We split a solution into two groups:
  - One group including a particular arc
  - The other excluding the arc
- Each splitting incurs a lower bound and we shall traverse the searching tree with the “lower” lower bound.

The traveling salesperson optimization problem

- The Cost Matrix for a Traveling Salesperson Problem.

Step 1 to reduce: Search each row for the smallest value

to j

from i

i \ j	1	2	3	4	5	6	7
1	∞	3	93	13	33	9	57
2	4	∞	77	42	21	16	34
3	45	17	∞	36	16	28	25
4	39	90	80	∞	56	7	91
5	28	46	88	33	∞	25	57
6	3	88	18	46	92	∞	7
7	44	26	33	27	84	39	∞

Step 2 to reduce: Search each column for the smallest value

The traveling salesperson optimization problem

- Reduced cost matrix:

i \ j	1	2	3	4	5	6	7	
1	∞	0	90	10	30	6	54	(-3)
2	0	∞	73	38	17	12	30	(-4)
3	29	1	∞	20	0	12	9	(-16)
4	32	83	73	∞	49	0	84	(-7)
5	3	21	63	8	∞	0	32	(-25)
6	0	85	15	43	89	∞	4	(-3)
7	18	0	7	1	58	13	∞	(-26)

reduced:84

A Reduced Cost Matrix.

The traveling salesperson optimization problem

j / i	1	2	3	4	5	6	7
1	∞	0	83	9	30	6	50
2	0	∞	66	37	17	12	26
3	29	1	∞	19	0	12	5
4	32	83	66	∞	49	0	80
5	3	21	56	7	∞	0	28
6	0	85	8	42	89	∞	0
7	18	0	0	0	58	13	∞
			(-7)	(-1)			(-4)

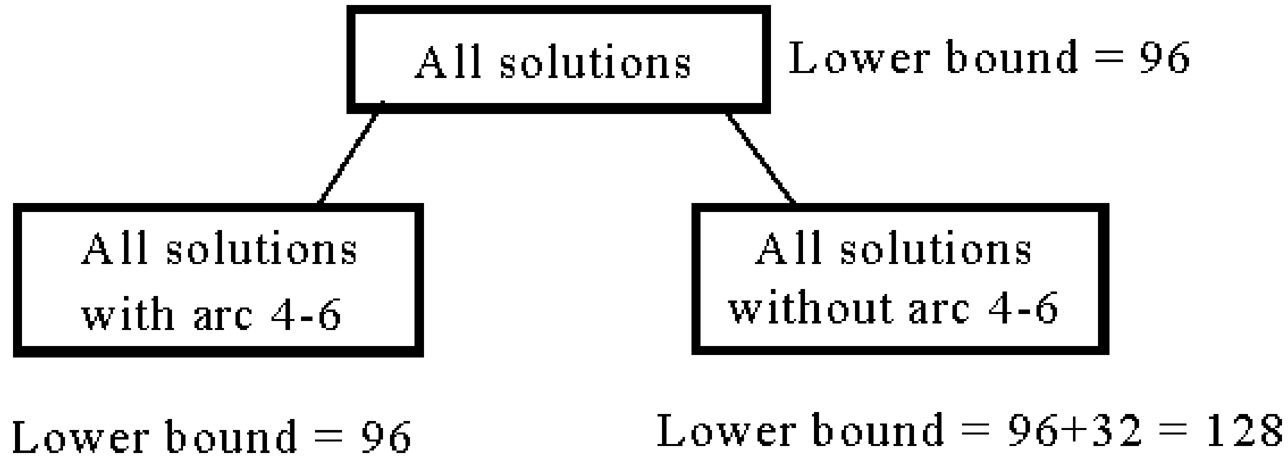
Table 6-5 Another Reduced Cost Matrix.

Lower bound

- The total cost of 84+12=96 is subtracted. Thus, we know the lower bound of feasible solutions to this TSP problem is 96.

The traveling salesperson optimization problem

- Total cost reduced: 84+7+1+4 = 96 (lower bound) decision tree:



The Highest Level of a Decision Tree.

- If we use arc 3-5 to split, the difference on the lower bounds is 17+1 = 18.

Heuristic to select an arc to split the solution space

- If an arc of cost 0 (x) is selected, then the lower bound is added by 0 (x) when the arc is included.
- If an arc <i,j> is not included, then the cost of the second smallest value (y) in row i and the second smallest value (z) in column j is added to the lower bound.
- Select the arc with the largest (y+z)-x

We only have to set c4-6 to be ∞.

For the right subtree (Arc 4-6 is excluded)

j / i	1	2	3	4	5	6	7
1	∞	0	83	9	30	6	50
2	0	∞	66	37	17	12	26
3	29	1	∞	19	0	12	5
4	32	83	66	∞	49	∞	80
5	3	21	56	7	∞	0	28
6	0	85	8	42	89	∞	0
7	18	0	0	0	58	13	∞

For the left subtree
(Arc 4-6 is included)

j / i	1	2	3	4	5	7
1	∞	0	83	9	30	50
2	0	∞	66	37	17	26
3	29	1	∞	19	0	5
5	3	21	56	7	∞	28
6	0	85	8	∞	89	0
7	18	0	0	0	58	∞

A Reduced Cost Matrix if Arc 4-6 is included.

1. 4^{th} row is deleted.
2. 6^{th} column is deleted.
3. We must set c6-4 to be ∞. (The reason will be clear later.)

For the left subtree

- The cost matrix for all solution with arc 4-6:

j / i	1	2	3	4	5	7	
1	∞	0	83	9	30	50	
2	0	∞	66	37	17	26	
3	29	1	∞	19	0	5	
5	0	18	53	4	∞	25	(-3)
6	0	85	8	∞	89	0	
7	18	0	0	0	58	∞	

A Reduced Cost Matrix for that in Table 6-6.

- Total cost reduced: 96+3 = 99 (new lower bound)

Upper bound

- We follow the best-first search scheme and can obtain a feasible solution with cost C.
- C serves as an upper bound of the optimal solution and many branches may be terminated if their lower bounds are equal to or larger than C.

⊕ : expanding order (selecting order)
: adding order

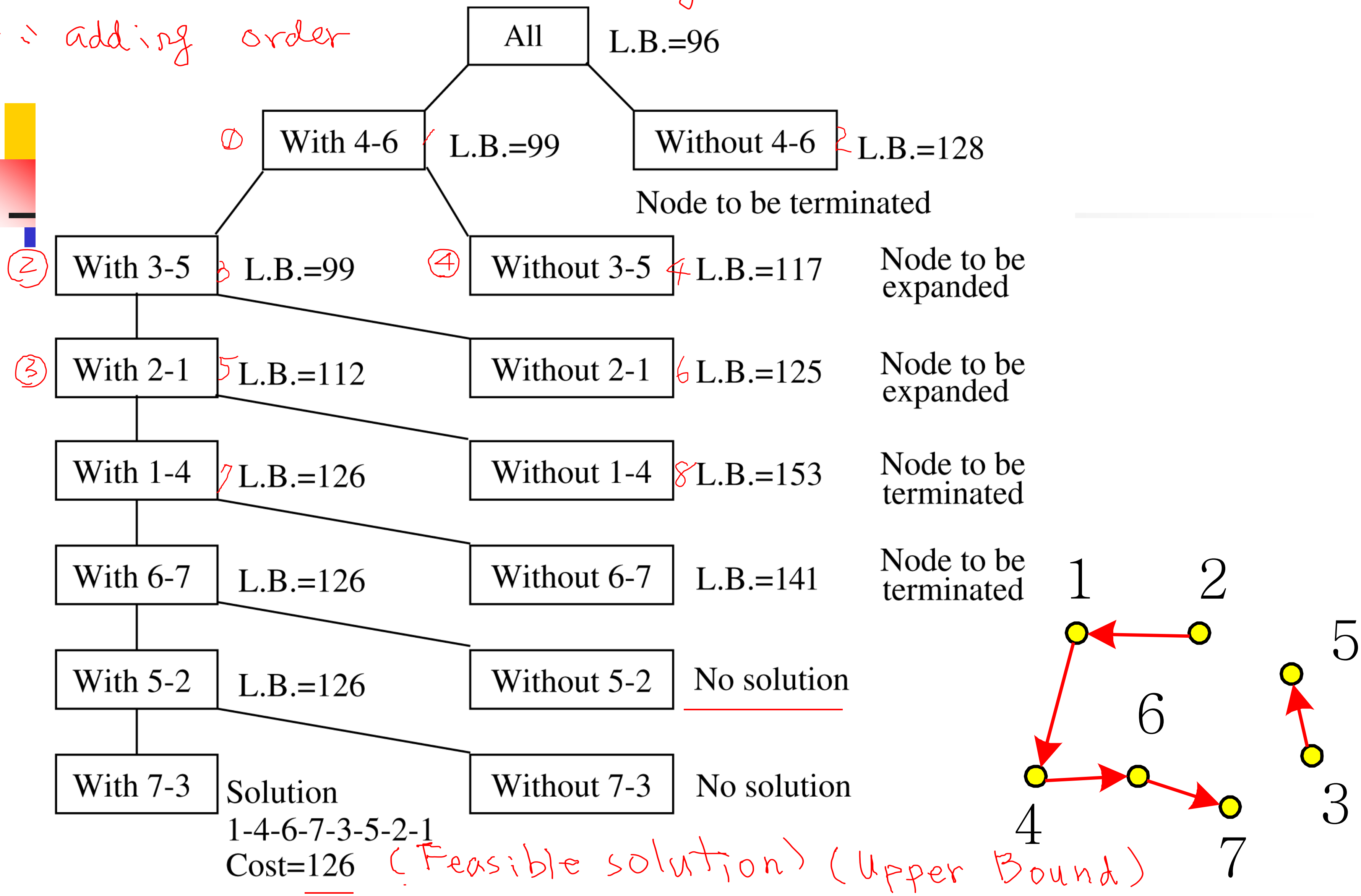


Fig 6-26 A Branch-and-Bound Solution of a Traveling Salesperson Problem.

Preventing an arc

- In general, if paths i_1-i_2-…-i_m and j_1-j_2-…-j_n have already been included and a path from i_m to j_1 is to be added, then path from j_n to i_1 must be prevented (by assigning the cost of j_n to i_1 to be ∞)
- For example, if 4-6, 2-1 are included and 1-4 is to be added, we must prevent 6-2 from being used by setting c6-2=∞. If 6-2 is used, there will be a loop which is forbidden.

Application & Scope of research

- Application: Vehicle route
- Scope of research : an algorithm which improves time complexity of TSP problem

Assignment

Q.1) What are different search to find feasible solution of a problem?

Q.2) Which search is useful to find optimal solution of a given problem.

Q.3) Explain branch & bound design strategy.