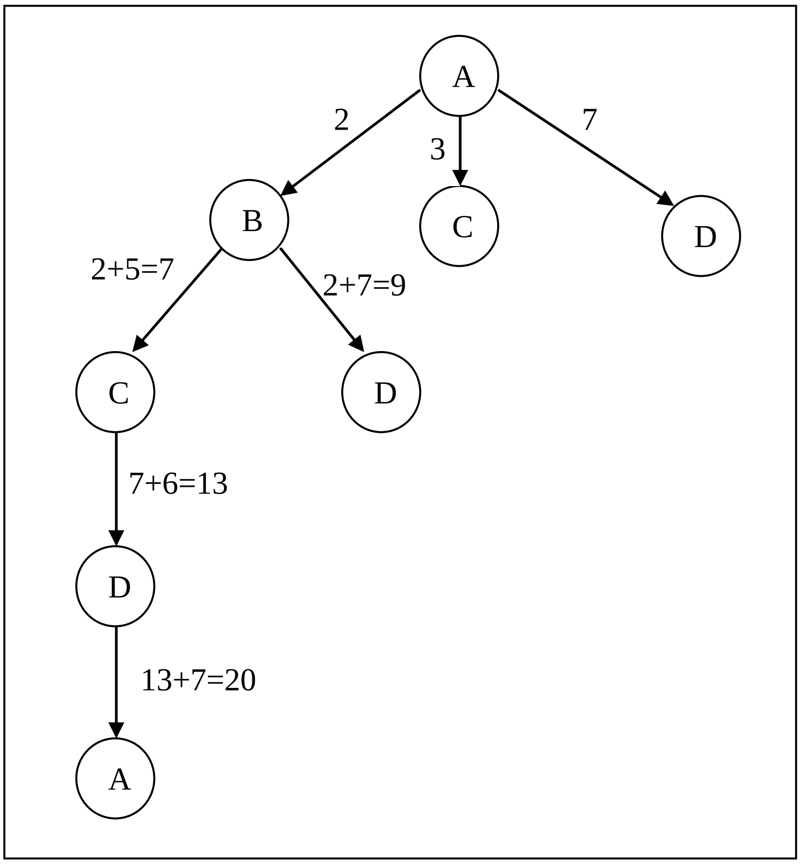
# CHAPTER TWO

# TRAVELING SALESMAN PROBLEM (TSP)

### 2.5.3 Tree Type Heuristic Method (TTHM)

The main step in this method, the objective function is evaluated at all nodes in each level of the search tree, then some of the nodes within each level (with minimum or maximum value) of the search tree are chosen from which to branch. Usually, one node is chosen with each level and stop at the first complete sequence of the nodes to be the solution.

**Example (2.3)**: Call example (2.1)




The route is : A-B-C-D-A with Z=20.

**Exercises (2.2)**:

1. A-C-D-B-A, Z=15.

| | A | B | C | D |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| A | -- | 3 | 3 | 7 |
| B | 3 | -- | 5 | 5 |
| C | 3 | 5 | -- | 4 |
| D | 7 | 5 | 4 | -- |

2. A-B-D-C-E-A, Z=90.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | -- | 15 | 18 | 15 | 21 |
| B | 19 | -- | 18 | 17 | 20 |
| C | 14 | 20 | -- | 19 | 20 |
| D | 24 | 23 | 20 | -- | 22 |
| E | 18 | 21 | 23 | 25 | -- |

### 2.5.4 Dynamic Programming / Held Karp Algorithm

http://www.facebook.com/tusharroy25

Dynamic programming (DP) is a mathematical technique well suited for the optimization of multistage decision problems. The DP for n-variables problem is represented as a sequence of n single variable problems which are solved successively. The main objective of DP is finding the shortest path in a state space graph G in which vertices corresponding to subset S and in which arcs corresponding to a decision where by the transition to a new state from a previous state is archived by sequencing the nodes. Clearly, there are $2^n$ vertices in the graph.

**Example (2.4)**: Solve the following TSP using DP.

| | A | B | C | D |
|---|---|---|---|---|
| A | -- | 1 | 15 | 6 |
| B | 2 | -- | 7 | 3 |
| C | 9 | 6 | -- | 12 |
| D | 10 | 4 | 8 | -- |

Solution:

Start at vertex=A

$2^n=2^3$ status:

1. {A}.

2. {B},{C},{D}.
3. {B,C},{B,D},{C,D}.
4. {B,C,D}.

1. {A}

| | | Cost | Parent |
|---|---|---|---|
| [A,B] | = | 1 | A |
| [A,C] | = | 15 | A |
| [A,D] | = | 6 | A |

2. {B},{C},{D}

| | | | | | | | Cost | Parent |
|---|---|---|---|---|---|---|---|---|
| [{B},D] | = | [A,B]+B→C | | | | | | |
| | = | B→C+[A,B] | = | 7+1 | = | | 8 | B |
| [{B},D] | = | B→D+[A,B] | = | 3+1 | = | | 4 | B |
| [{C},B] | = | C→B+[A,C] | = | 6+15 | = | | 21 | C |
| [{C},D] | = | C→D+[A,C] | = | 12+15 | = | | 27 | C |
| [{D},B] | = | D→B+[A,D] | = | 4+6 | = | | 10 | D |
| [{D},C] | = | D→C+[A,D] | = | 8+6 | = | | 14 | D |

3. {B,C},{B,D},{C,D}

| | | | | Cost | Parent |
|---|---|---|---|---|---|
| [{B,C},D] | = | min{B→D+[{C},B], C→D+[{B},C]} | | | |
| | = | min {3+21,12+8} | = | 20 | C |
| [{B,D},C] | = | min{B→C+[{D},B], D→C+[{B},D]} | | | |
| | = | min {7+10,8+4} | = | 12 | D |
| [{C,D},B] | = | min{C→B+[{D},C], D→B+[{C},D]} | | | |
| | = | min {6+14,4+27} | = | 20 | C |

4. {B,C,D}

[{B,C,D},A]= min{B→A+[{C,D},B], C→A+[{B,D},C], D→A+[{B,C},D]}

= min { 2+20,9+12,10+20} = 21 C

The route is: **A→B→D→C→A**, with Z=1+3+8+9=21.

**Time complexity:** since we have $2^n$ subsets and matrix $n \times n = n^2$, then the complexity is $O(2^n \times n^2)$.

**Exercises (2.3)**:

1. A-D-B-C-E-A, Z=16.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | -- | 3 | 6 | 2 | 3 |
| B | 3 | -- | 5 | 2 | 3 |
| C | 6 | 5 | -- | 6 | 4 |
| D | 2 | 2 | 6 | -- | 6 |
| E | 3 | 3 | 4 | 6 | -- |

2. A-E-F-C-D-B-A, Z=103.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | -- | 20 | 23 | 27 | 29 | 34 |
| B | 21 | -- | 19 | 26 | 31 | 24 |
| C | 26 | 28 | -- | 15 | 36 | 26 |
| D | 25 | 16 | 25 | -- | 23 | 18 |
| E | 23 | 40 | 13 | 31 | -- | 10 |
| F | 27 | 18 | 12 | 35 | 16 | -- |

### 2.5. 5 Greedy Method (GRM)

This algorithm starts by sorting the edges by length, and always adding the shortest remaining available edge to the tour. The shortest edge is available if it is not yet added to the tour and if adding it would not create a 3-degree vertex or a cycle with edges less than n.