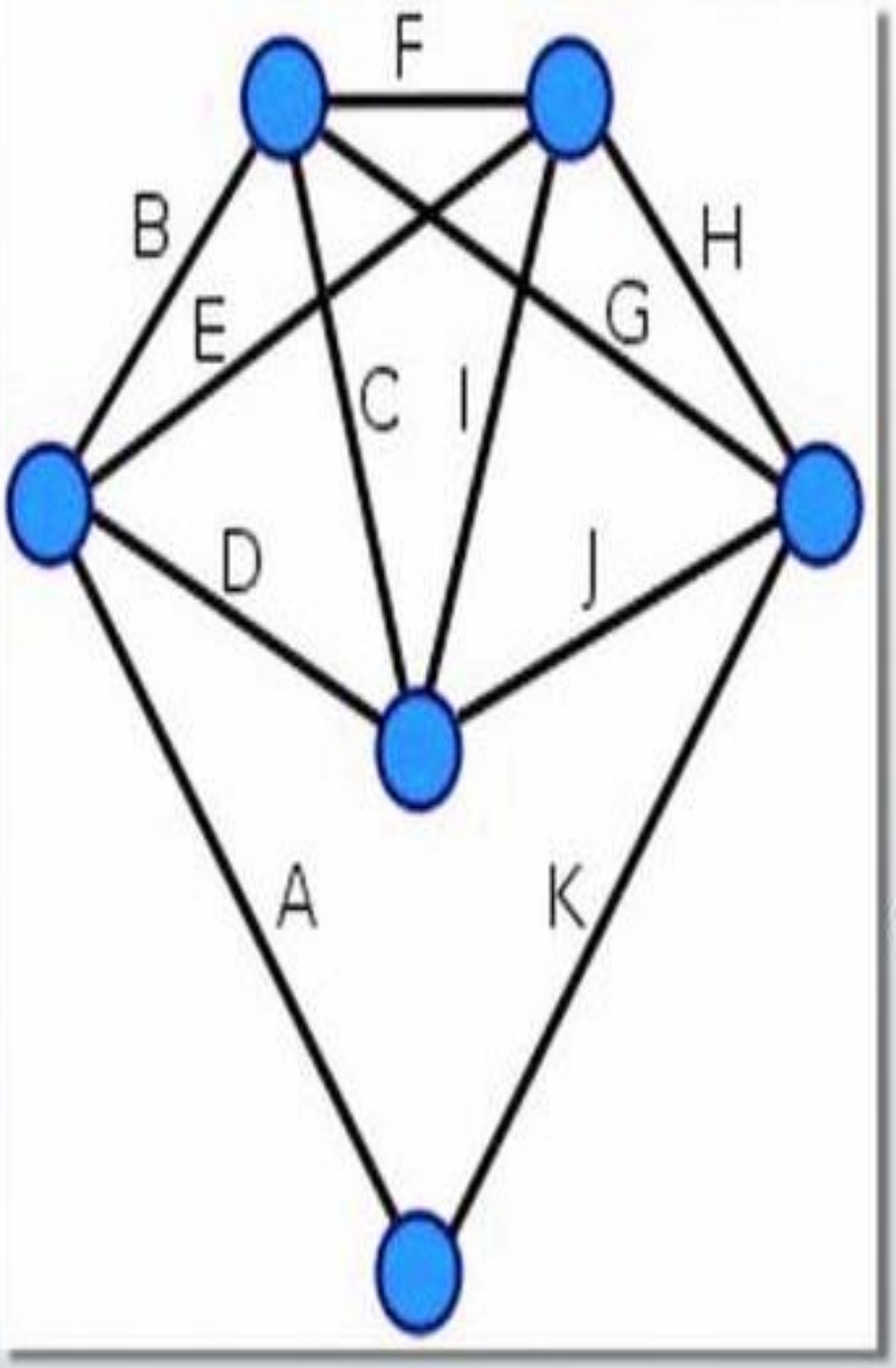
Traveling salesman problem


F
B
H
E
G
C
I
D
J
A
K


How?


JIAM

Problem Statement

- Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- In this assignment you will be solving TSP using local search.

First Choice Hill Climbing

```
current = Randomly generated initial state
count = 0
while (true)
    for( i=0; i < σ; i++)
        ran = random number between [0,1]
        if (ran <0.5)
                neighbor <- a successor of current generated
                            by "heuristic1"
        else
                neighbor <- a successor of current generated
                            by "heuristic2"

        if neighbor better than current
                break;
     if(i == σ)
          return current
     current = neighbor
     count++
```

Heuristics for generating neighbors

Randomly pick two cities from the route and exchange them.

```
oneOneExchange(route)
{
      n = length  of route;
      ran = random integer between [1,n]
      ran2 = random integer between [1,n] and not equal to ran
      swap the cities at ran and ran2
      return route
}
```

Heuristics for Tour improvement

It attempts to improve the current tour by first moving a chain of three consecutive vertices in a different location (and possibly reversing it) to find an improved route. All possible chain is checked. If no improvement can be obtained, the process is then repeated with chains of two consecutive vertices, and then with single vertices.

Heuristics for Tour improvement

```
or-opt(route)
{
     n = length of route
     for (chain_size=3;chain_size>=1;chain_size--)
     {
          for(i=1; i + chain_size-1<=n; i++)
          {
               chain = the subroute of route starting at
                position i and having a length of chain_size
               tmp_route = remove chain from route
               nn = length of tmp_route
               for(j=1;j<=nn;j++)
               {
                         new_route = insert chain to tmp_route
                                     at position j
                         if new_route is better than route
                              return new_route
               }
          }
     }
     return route
}
```