Operations Research

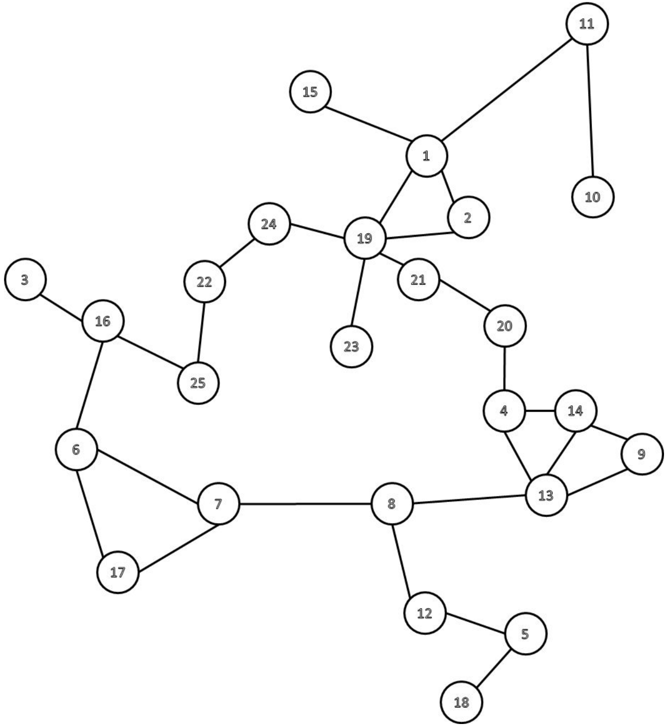
Final Presentation

Members : 周永堂、鍾孟芳、林家毅、童安弘

Agenda

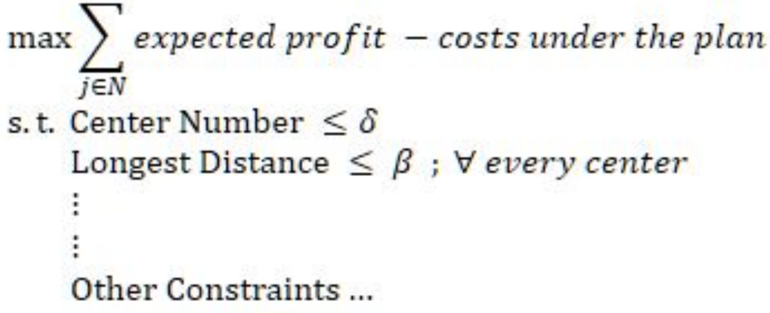
- Research Topics
    - Traditional Allocation Problem
    - Allocation with connection : Articulation Point
- Operations Research (How we do it)
    - Data Input & Result Overview
    - Tarjan’s Algorithm and Dilemmas
    - Improving the Model : Optimization Limit
    - Steps
        - Find Articulation Points
        - Iteration & Solution
- Takeaway : Future Thoughts to Generalize Model

Traditional Allocation Problem



Consider a City Network :

How to build a logistic center plan?



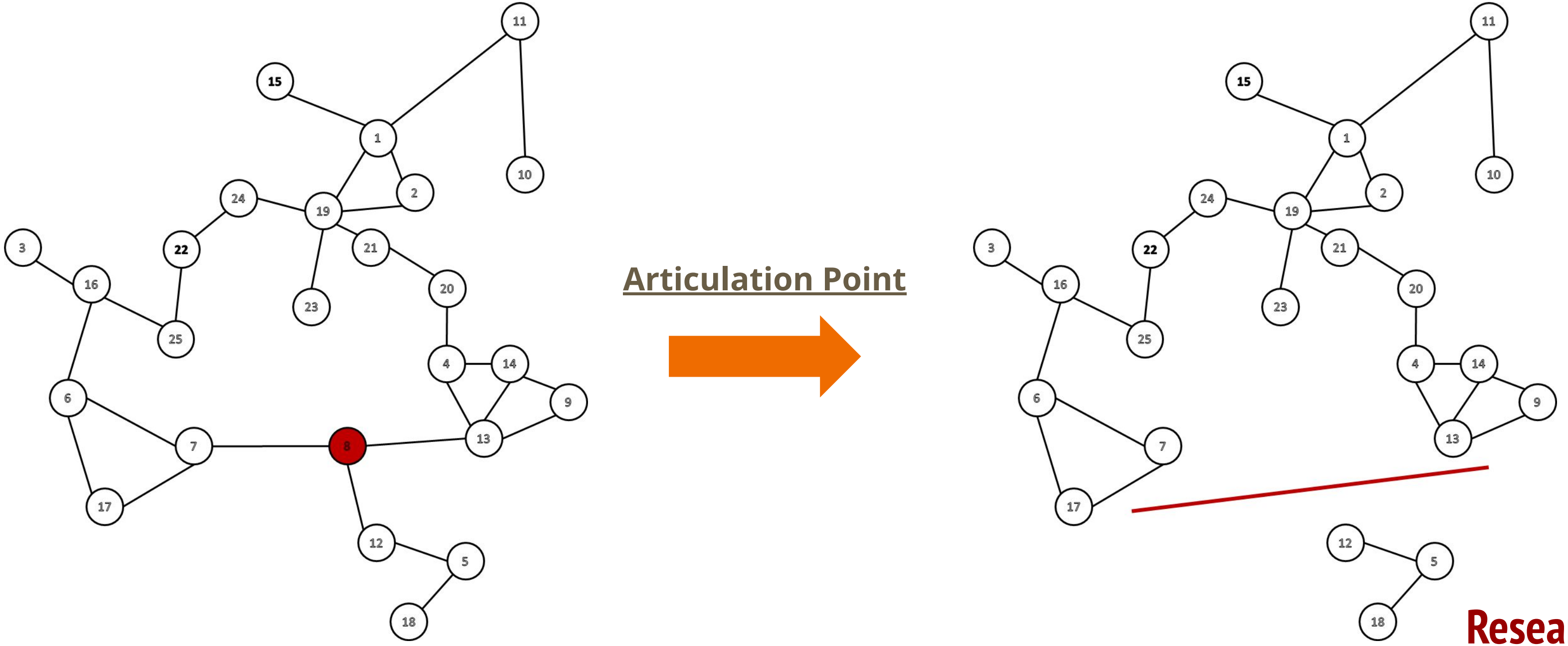
$$\max \sum_{j \in N} \textit{expected profit} \; - \textit{costs under the plan}$$

$$\begin{aligned} \text{s.t.}\;\; & \text{Center Number} \;\le \delta \\ & \text{Longest Distance} \;\le \beta \;\; ; \; \forall \textit{ every center} \\ & \vdots \\ & \vdots \\ & \text{Other Constraints} \ldots \end{aligned}$$

Imagine a war is approaching....

Connections are first to consider

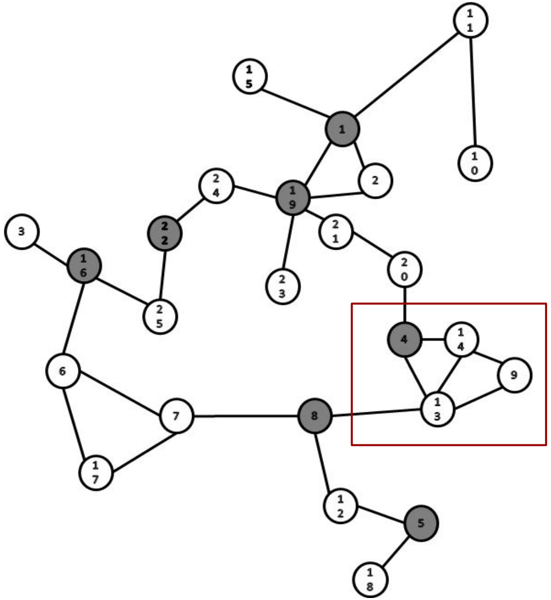
- Circles represent cities.
- Lines represent roads connecting cities.

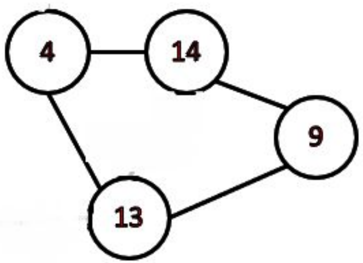


Articulation Point

However we can’t take all cities once because…

- **Capabilities** : We cannot build/explode in as many cities as we want

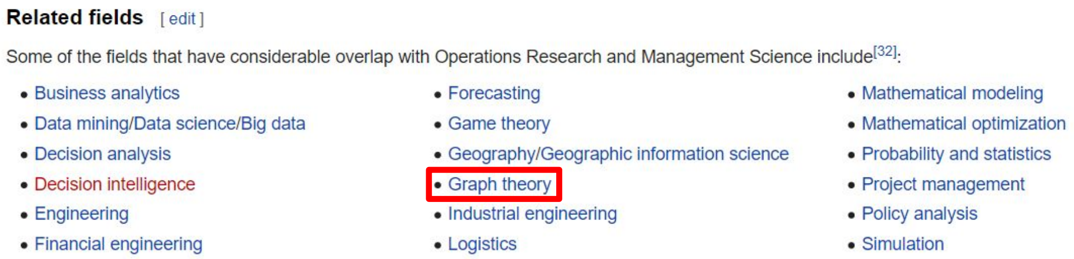




Stop when encoutering cycle

Research Goal

- Finding the articulation points, each with its own benefits defined by:
  - How many pieces it breaks the original graph?
  - How many total distance it can create?
- This is a **graph theory (a branch of operations research)** with **iterative searching solutions (DFS or BFS)** since whether the point is an articulation point will only be given at that state, we must iterate to find the optimization solution.



Related fields [edit]

Some of the fields that have considerable overlap with Operations Research and Management Science include[32]:

- Business analytics
- Data mining/Data science/Big data
- Decision analysis
- Decision intelligence
- Engineering
- Financial engineering
- Forecasting
- Game theory
- Geography/Geographic information science
- Graph theory
- Industrial engineering
- Logistics
- Mathematical modeling
- Mathematical optimization
- Probability and statistics
- Project management
- Policy analysis
- Simulation

Source: wikipedia

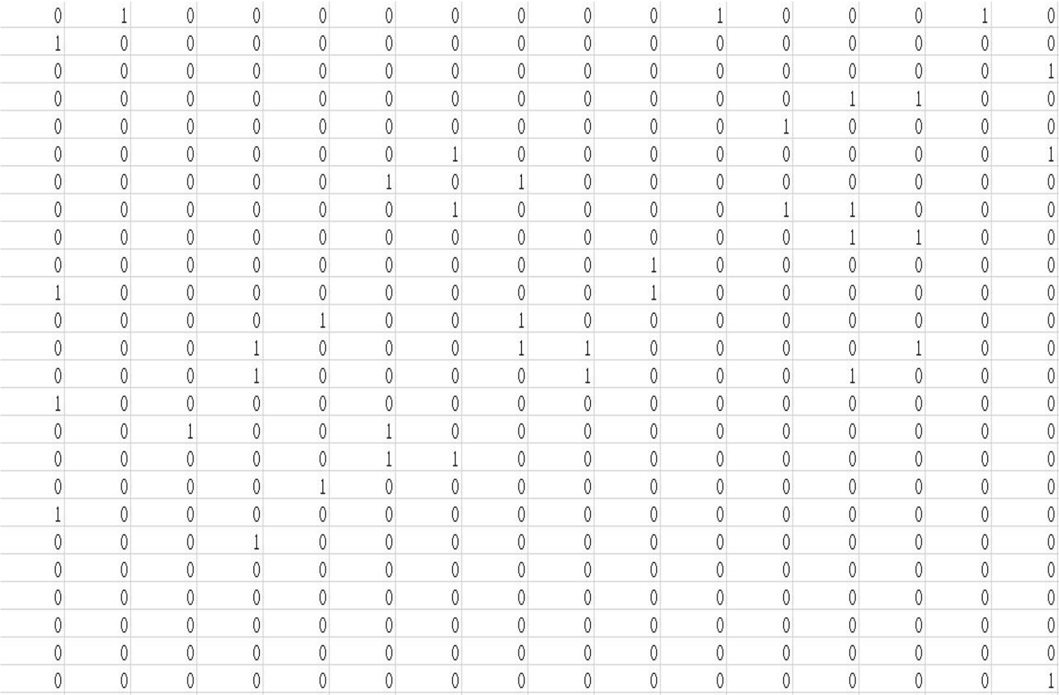
Data Overview

- Total number of cities : **25**
- Adjacency Matrix : **25*25**
- Distance Matrix : **25*25**
- Cost for each explode : λ
- Maximum Explosion : 10

Objective Function :

Pieces Number * Total Distance

- Connections



0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Mathematics Formulation

- Objective Function and Constraints

$$\max \text{sum}_t(Ptj*Dtj) \quad t \in T,\ j \in \text{length}(At)$$

$$\text{s.t.} \quad At = B(S(At\text{-}1))\ ;\ \forall t = 2,\ldots,T$$

$$A1 = B(G)$$

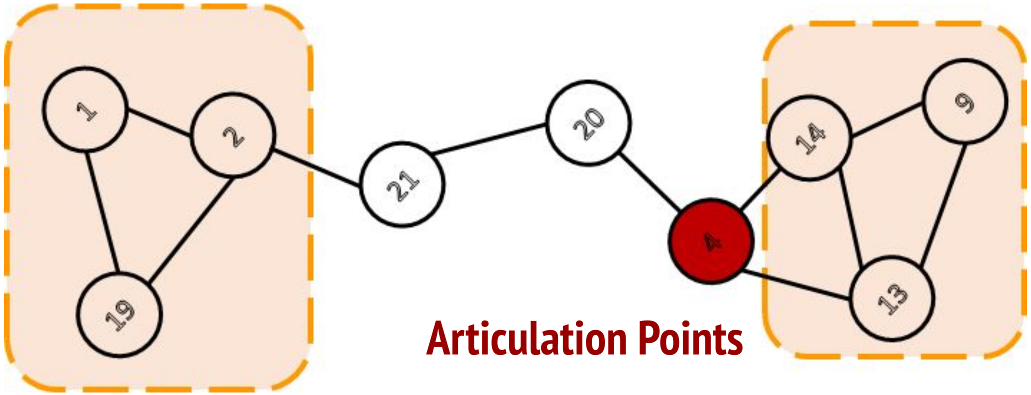
$$0 < t \leq T$$

$$j \in At$$

- t = current level
- At = articulation set at t
- j ∈ {1,...,25}, index
- S(A) = a function that generates subplots from the given articulation points
- B(Fragment) = a function that finds all articulation points
- Ptj = number of pieces of articulation j at level t
- Dtj = total distance an articulation j at level t
- G = original graph
- T = maximum iterations

Articulation Points & Biconnected Components

- Articulation Point disconnects an undirected graph
- Biconnected Component does not generate articulation points

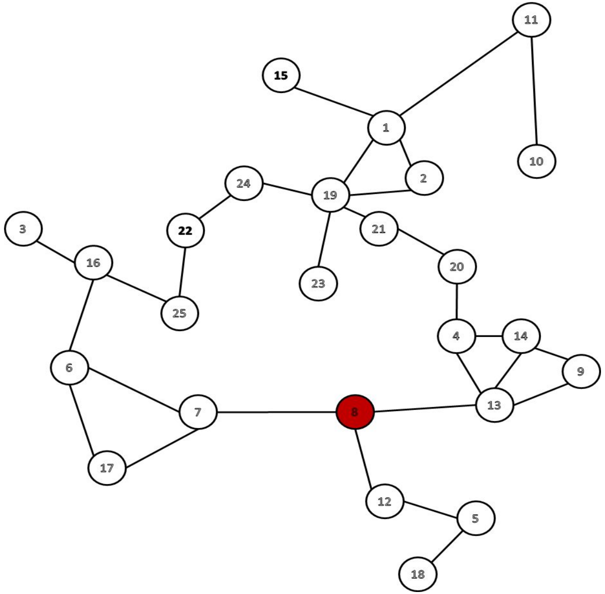


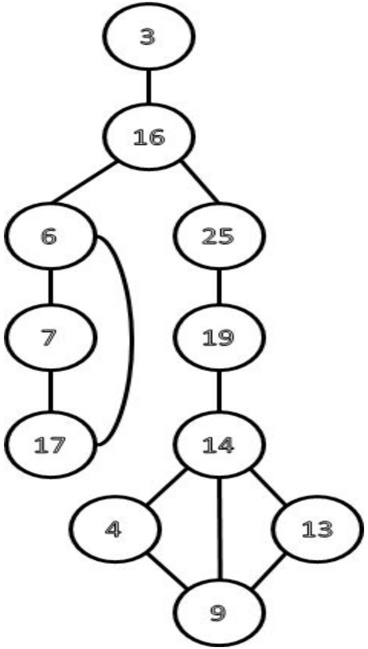
Articulation Points

Biconnected Component

How to find articulation points : Tarjan’s Algorithm

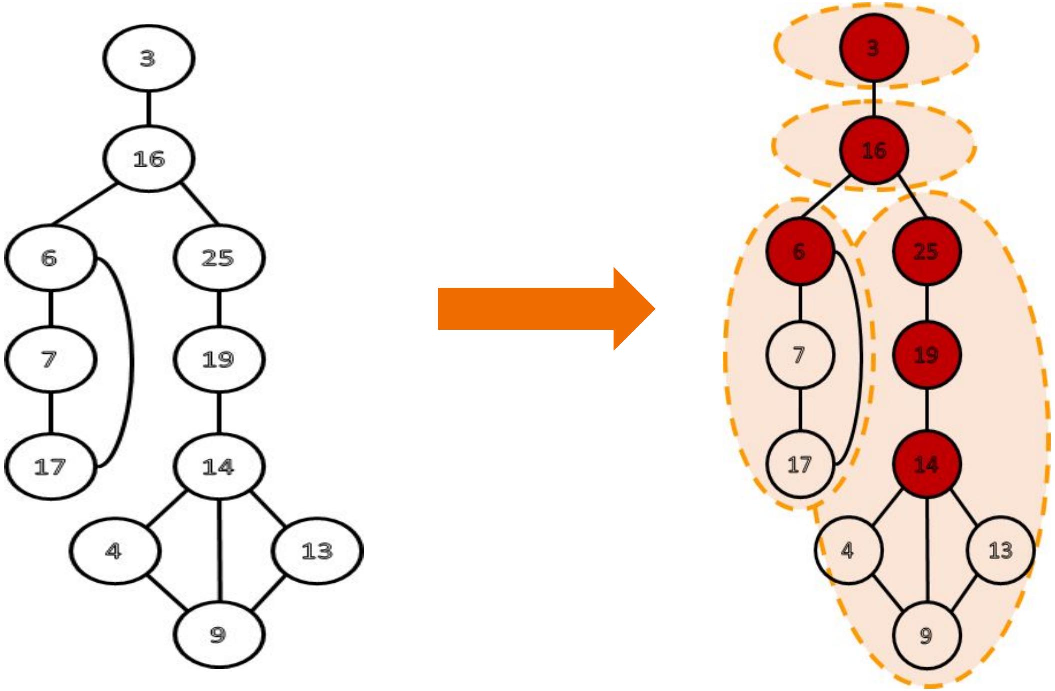
- The algorithm is to find all **Biconnected Components** and thus we have all the **Articulation Points**





How to find articulation points : Tarjan's Algorithm

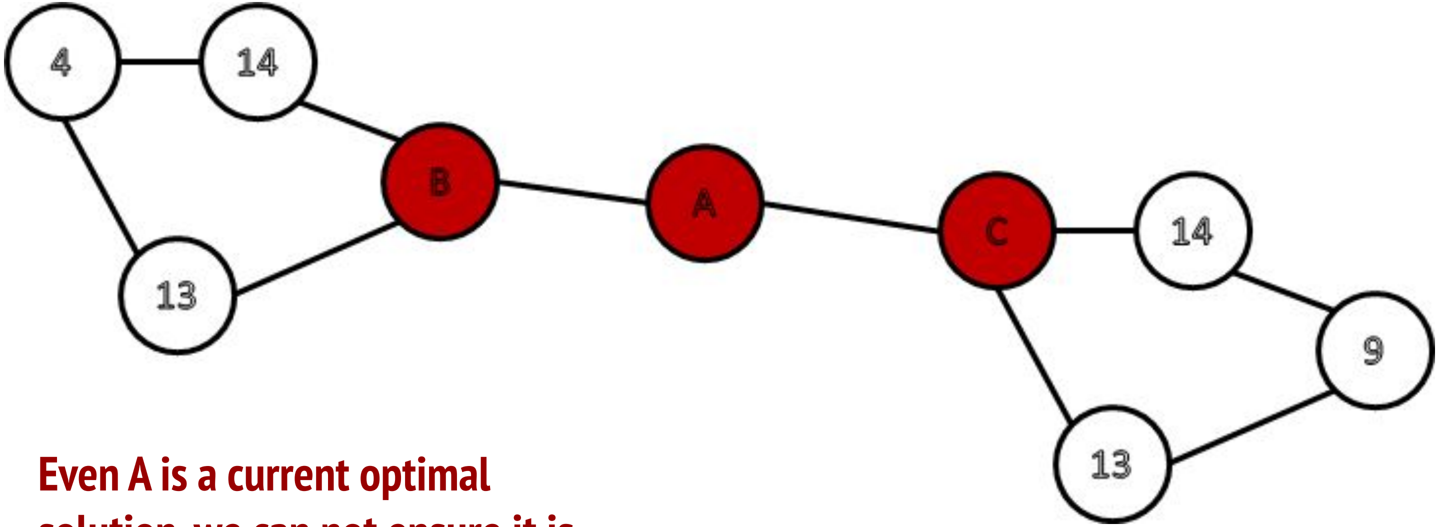
- Step 2 : DFS each point and find highest ancestor for each point



If the ancestor of a point is itself, it's a biconnected component

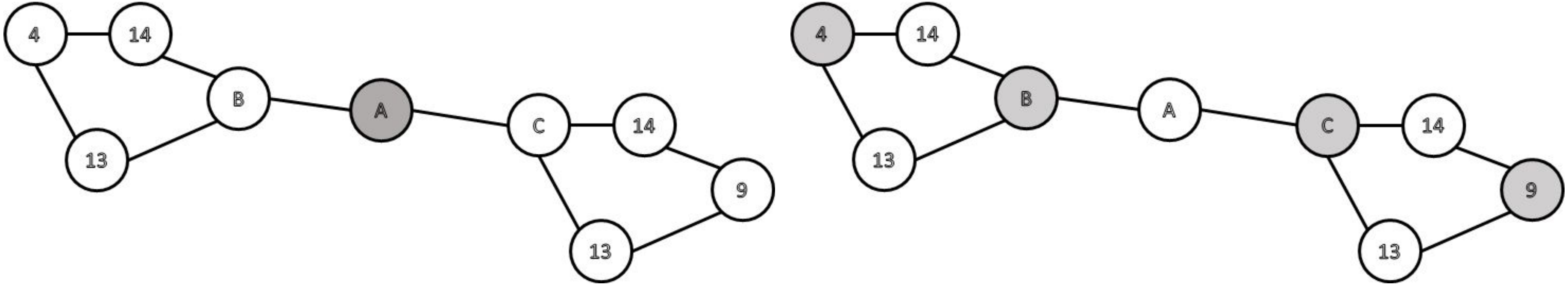
The overlapping part of each BCC is an articulation point

Algorithm Dilemma : Not optimization solution



➡ Even A is a current optimal solution, we can not ensure it is also a global solution.

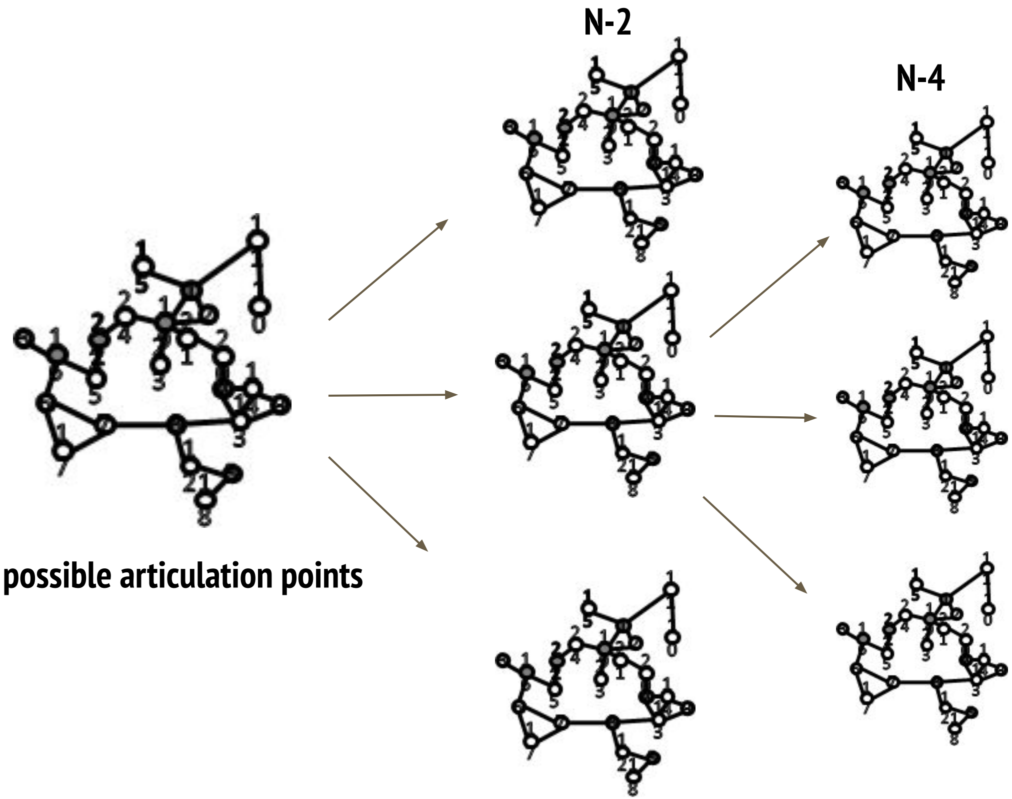
Algorithm Dilemma : Not optimization solution



No one knows which division is better considered pieces and distance

→ Iteration Solutions

Another Dilemma : Is it solvable?



N-2

N-4

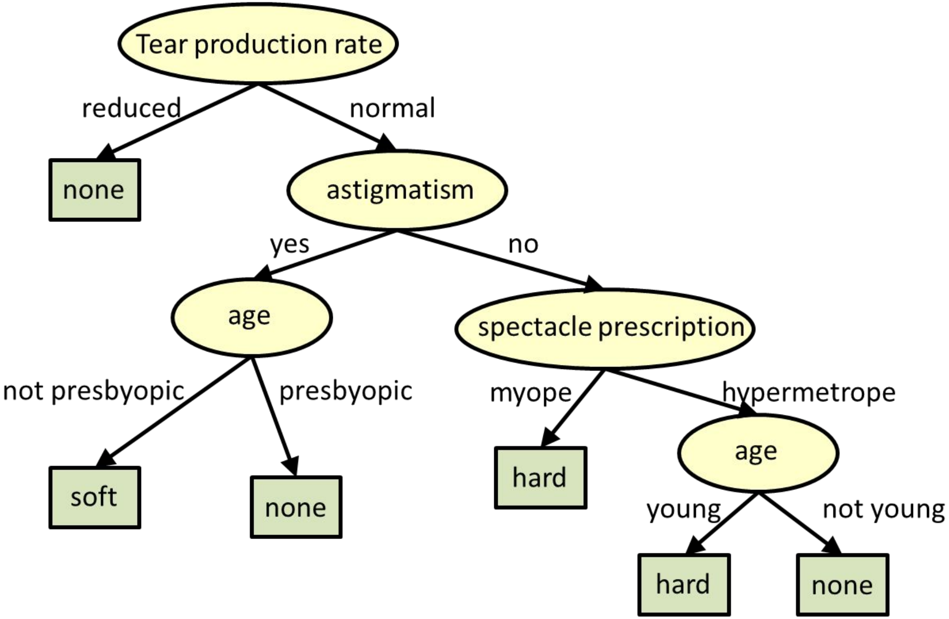
N possible articulation points

Time Complexity?

$$2N * (2n-2) * \cdots 2 = O(2^n n!)$$

... TSP only O(n!)

Improving Model : Tree-model Adjustment



Solving This is Like a Decision Tree

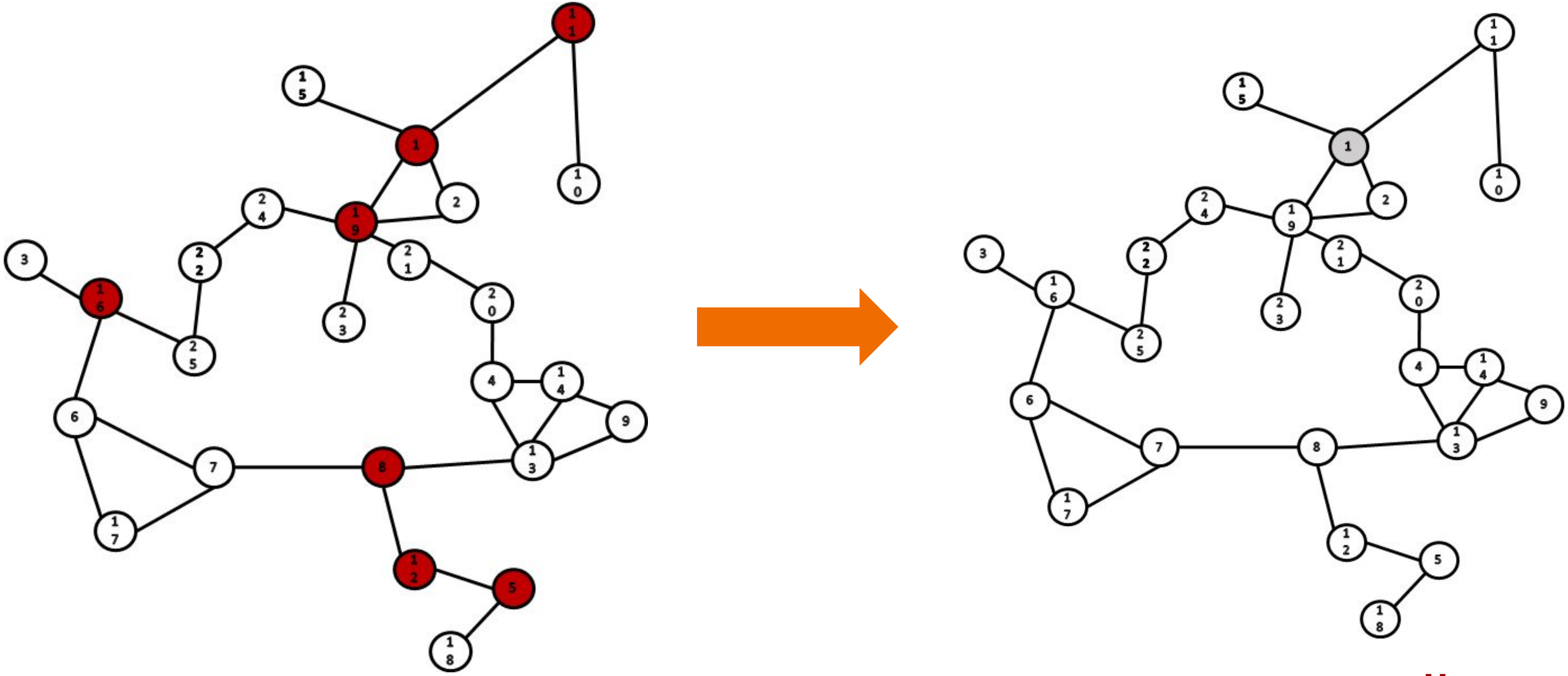
- **Max_Depth** : Stop if tree depth larger than this number (we can launch infinite missiles)
- **Min_Leaf_Split** : Do not accept the solution if profit less than this number (cost of each launch)
- **Tolerance** : If profit increase less than this number, stop generating new solution (cost of each launch)

Solvable when data is small but not satisfying

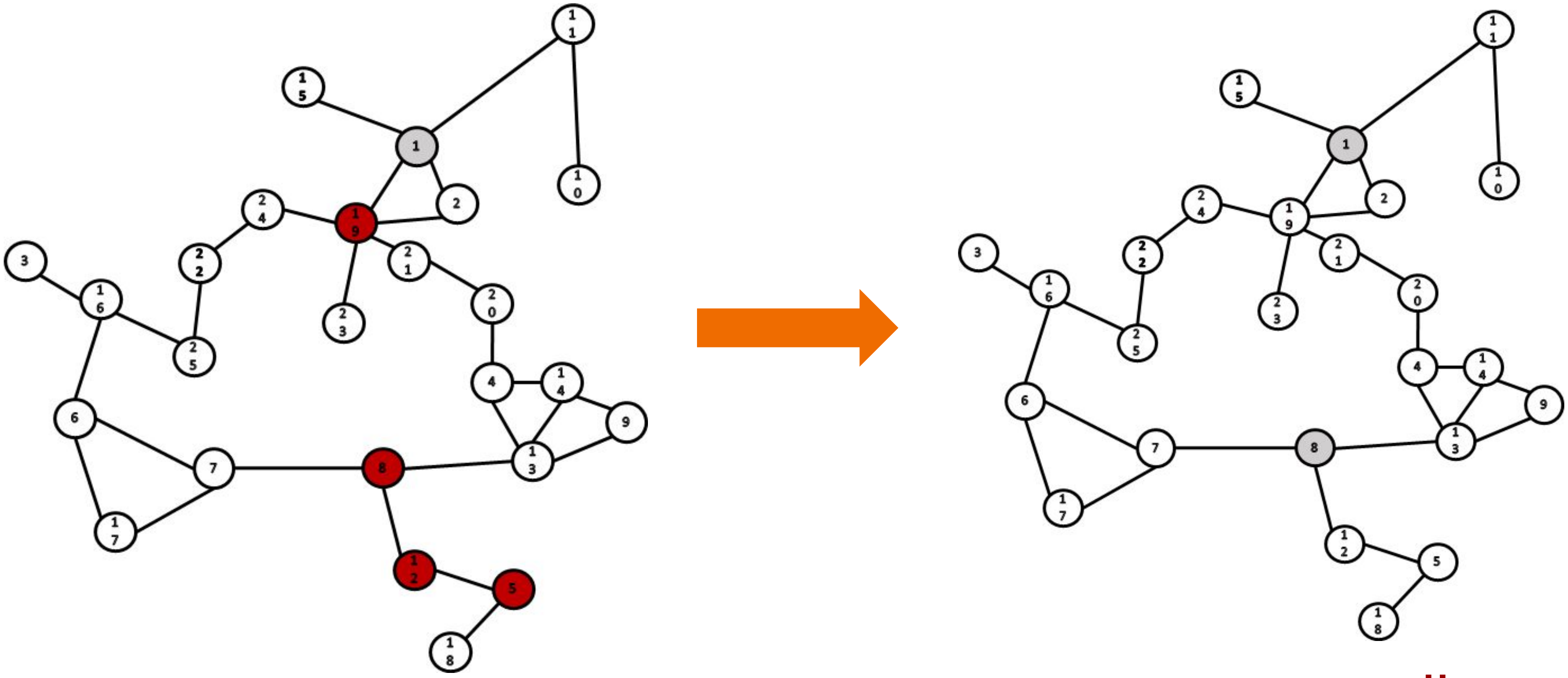
Steps : Virtual Code

```
void CityPlanning (adj matrix : graph, distance) {
    /* Articulation Set = [Articulation Point, Sub_graph, Layer] */
    Articulation_Set = Get_articulation_points(graph);
    while Articulation_Set not empty {  /* BFS Find Solution */
        sub_graph = Articulation Set.top()[1];
        solution_set = Get_articulation_points(sub_graph);
        Articulation_Set.pop() ; Articulation Set.push(solution set)
    }
}
```

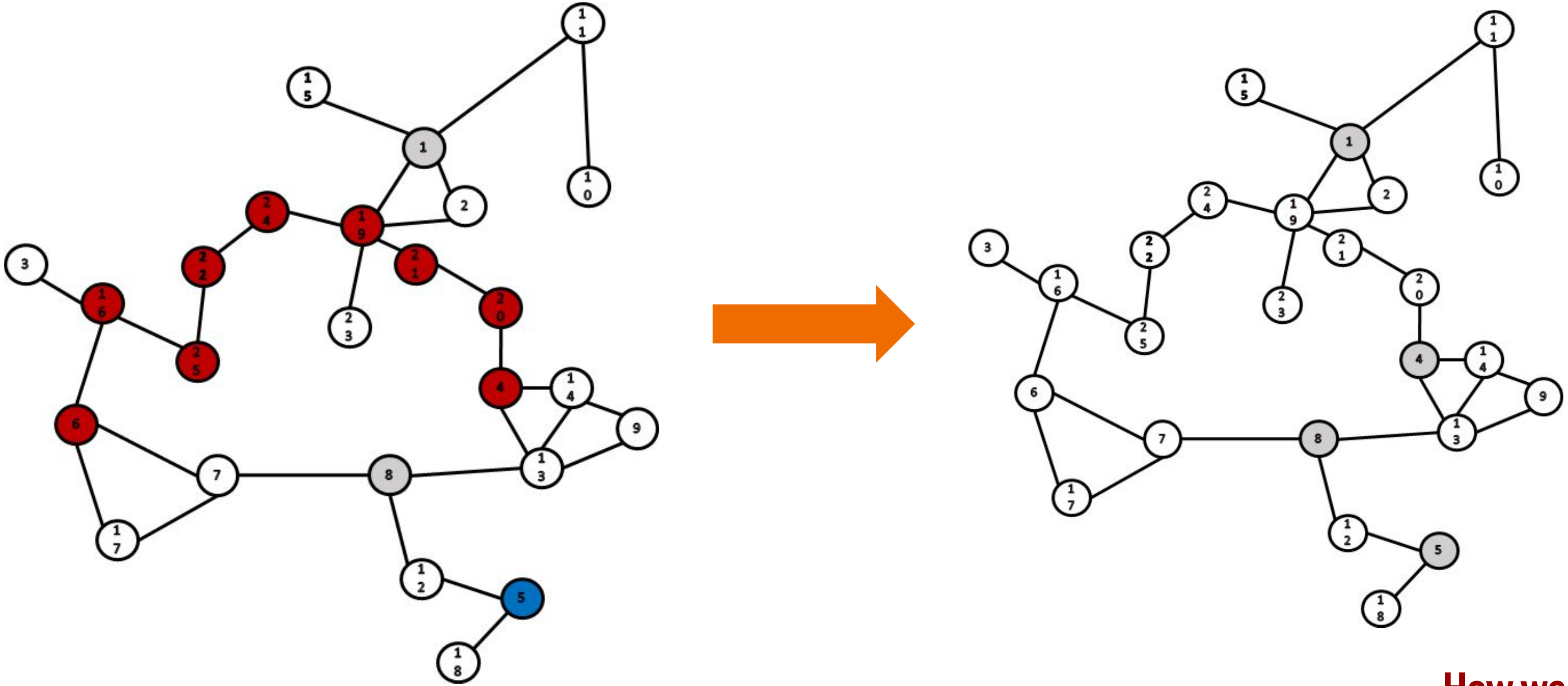
Steps : Visualization



Steps : Visualization

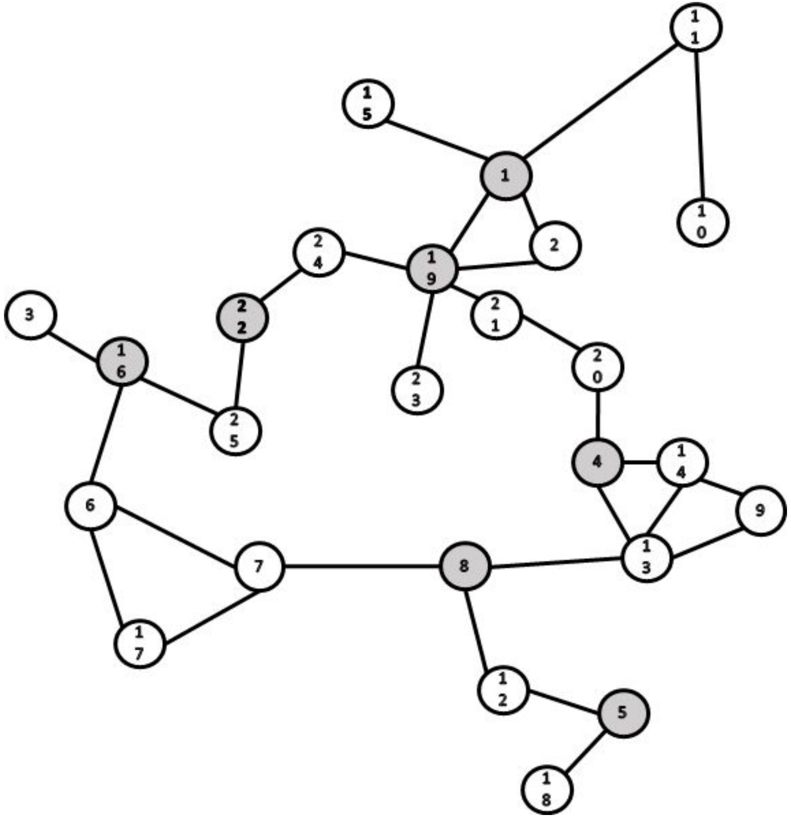


Steps : Visualization



Solution Interpretation

- Order : [1, 8, 19, 5, 4, 14, 20]
- Total Profit : 32714
- Interpretation : We follow the order to explode each city, and we do not run out all of our missiles because launching eighth missiles do not benefit in our planning.



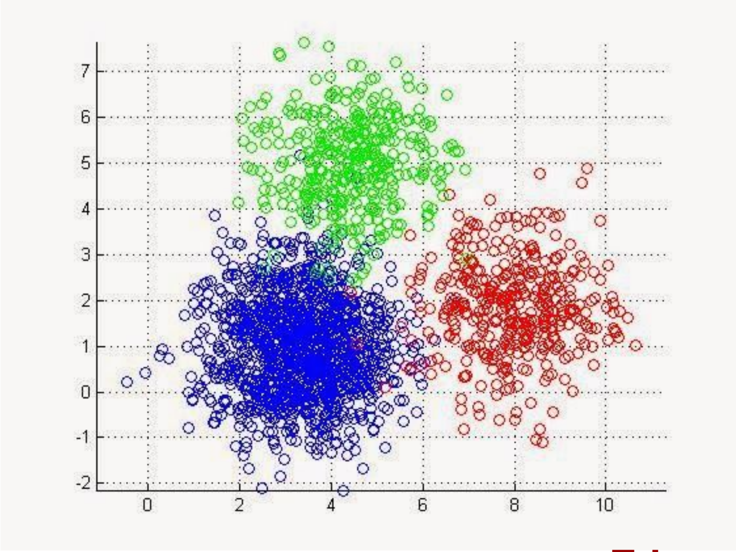
Takeaway : Future Improvement

- The model is not generalized enough, especially for big data
- Idea of the planning is like K-Means, which have :

$$l(G_1, \ldots, G_n) = \sum_{j=1}^{k} \sum_{i \in G_j} \left\| x_i - \mu_j \right\|_2^2$$

$$\bigcup_{j=1}^{k} G_j = \{1, 2, \ldots, n\}; G_j \cap G_j' = \phi$$

$$\mu_j = \frac{1}{|G_j|} \sum_{i \in G_j} x_i$$



Takeaway : Apply Community Detection

- A Community is a subgraph that contains nodes more **densely linked**

$$Q = \sum_{c \in C} \left(\frac{I_c}{m} - \left(\frac{D_c}{2m} \right)^2 \right)$$

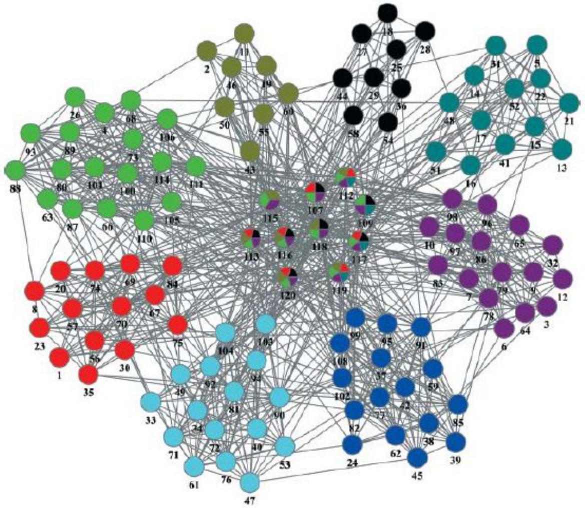
m = number of edges

I$_c$ = number of edges in community c

D$_c$ = sum of degree of all vertex in community

By BMLPA Algorithm, it could be done in O(nlogn)

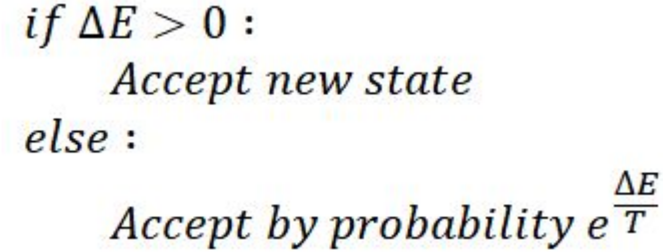
overlapping communities



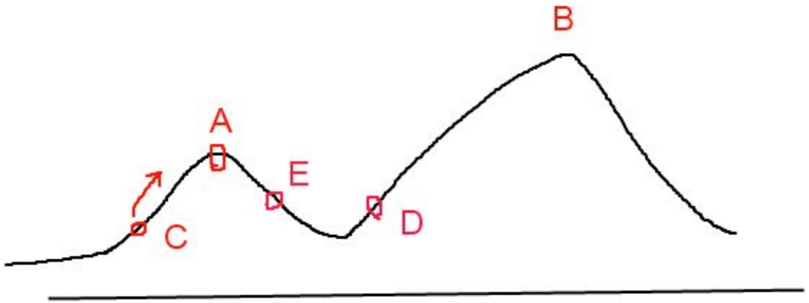
Takeaway : Simulated Annealing (Other Searching)

- SA algorithm can somewhat improve the TSP dilemma.

Accept new state by a specific probability (not definitely optimized solution)



```
if $\Delta E > 0$ :
    Accept new state
else :
    Accept by probability $e^{\frac{\Delta E}{T}}$
```



More possible searching methods

Thank You!

Final Presentation

Members : 周永堂、鍾孟芳、林家毅、童安弘