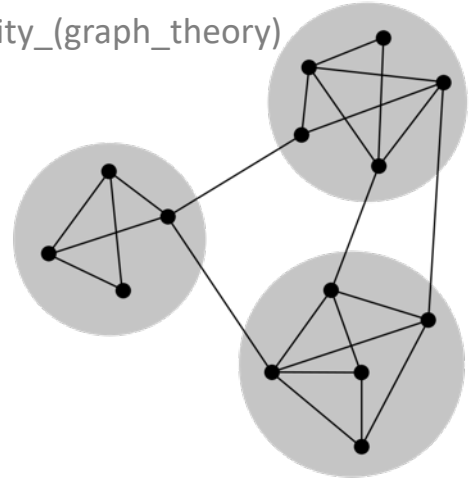
Mini course: Graph theory
Partitioning

4th March 2021

Introduction
Partitioning vs Community Detection

https://en.wikipedia.org/wiki/Connectivity_(graph_theory)

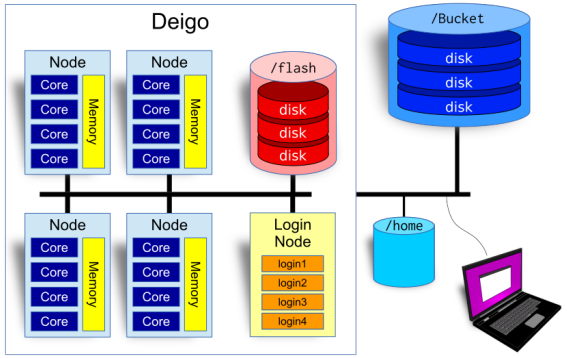


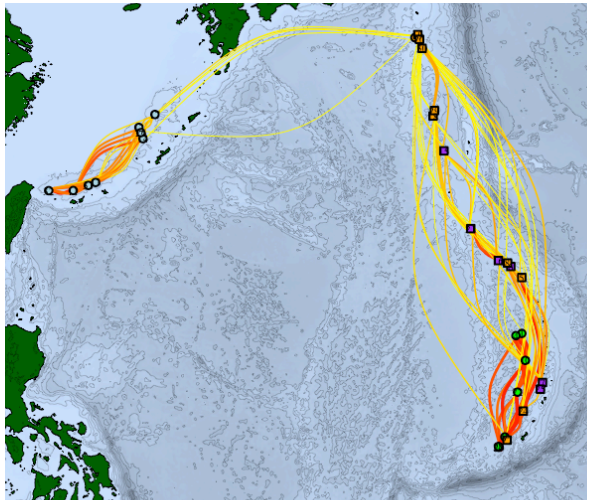
- **Both** look to sub-divide the graph into groups of nodes with minimal between-group edges

- **Partitioning** means you decide the number and size of the groups

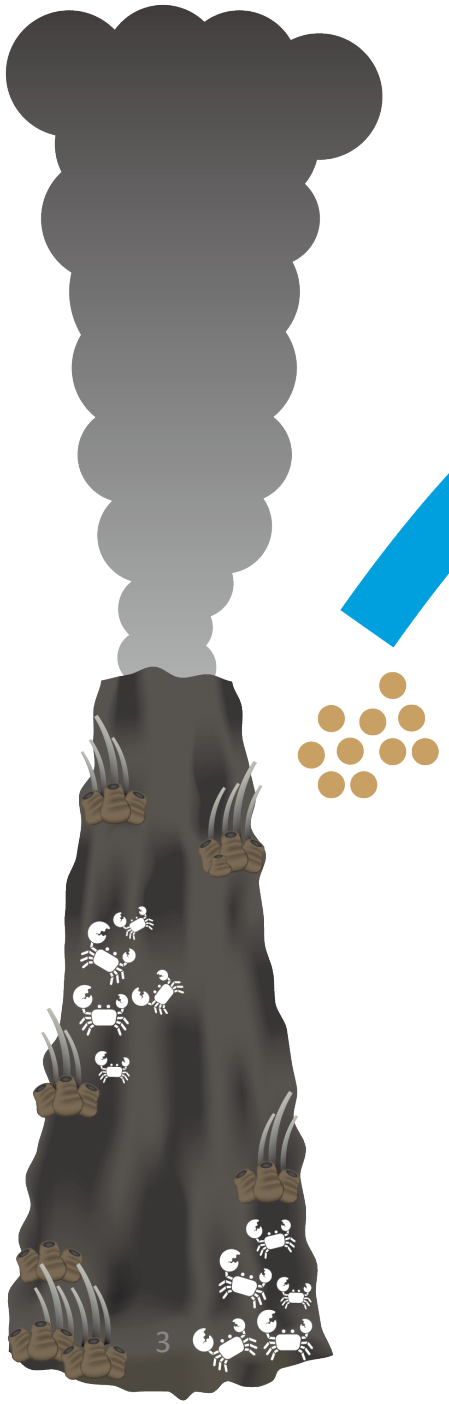
- **Community Detection** has no such a priori restrictions and instead find ‘naturally’ occuring groups of nodes.

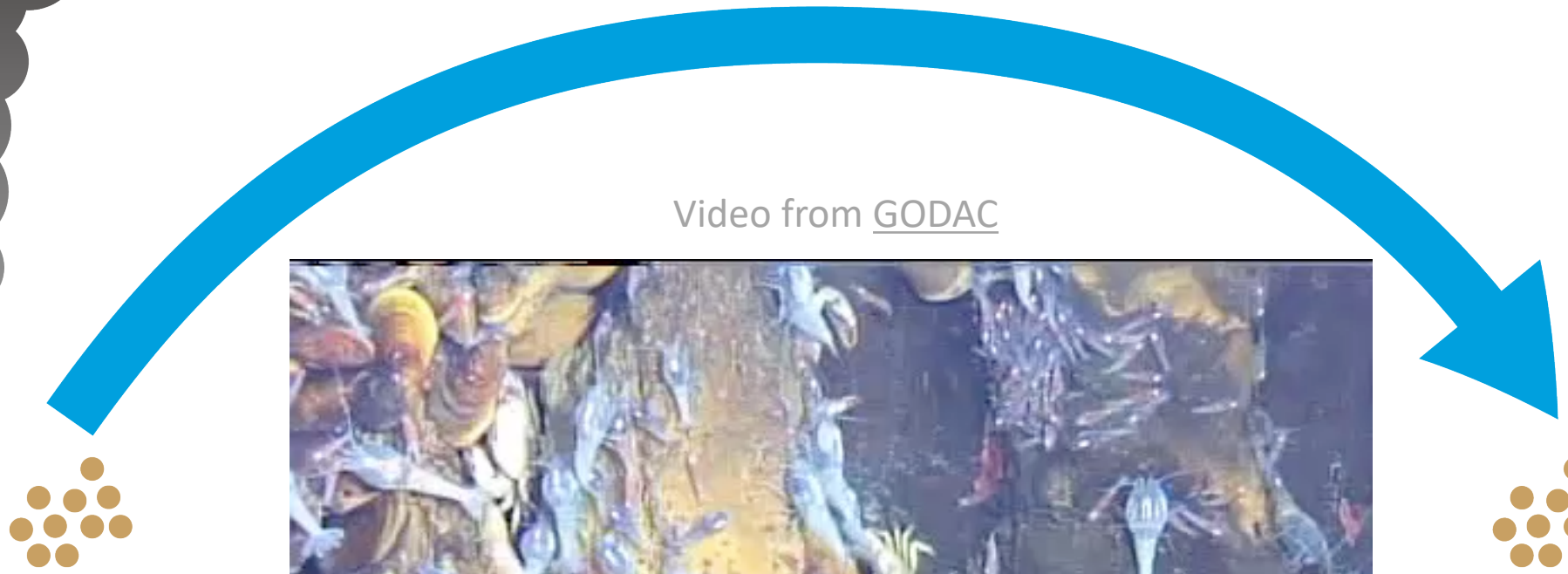
https://groups.oist.jp/scs/getting-started



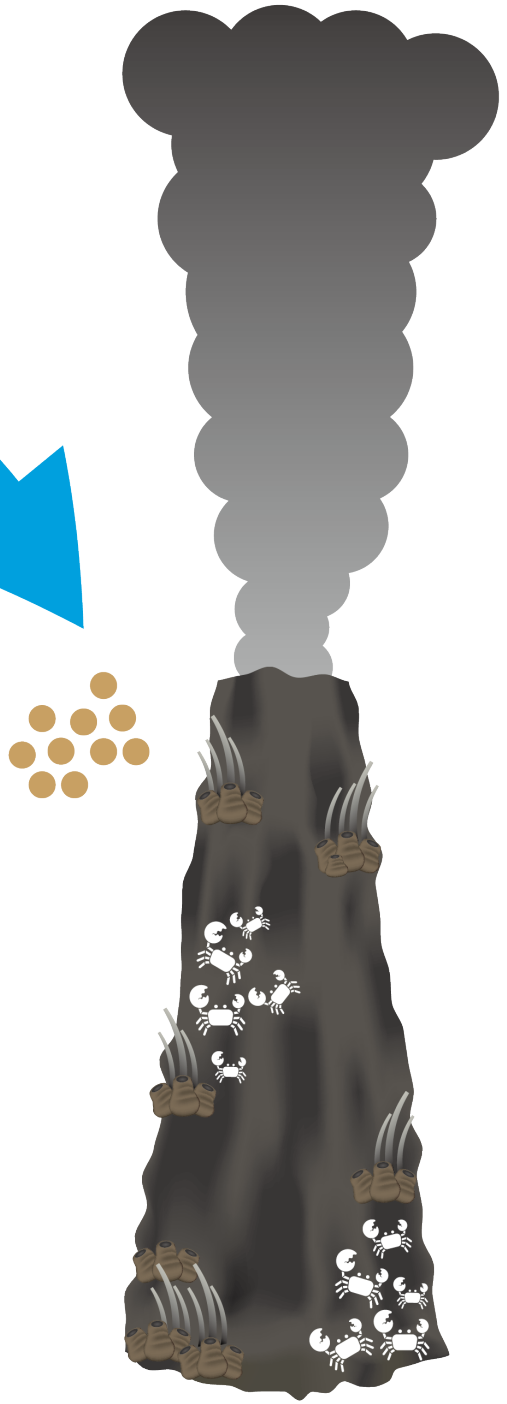


Introduction: My Research

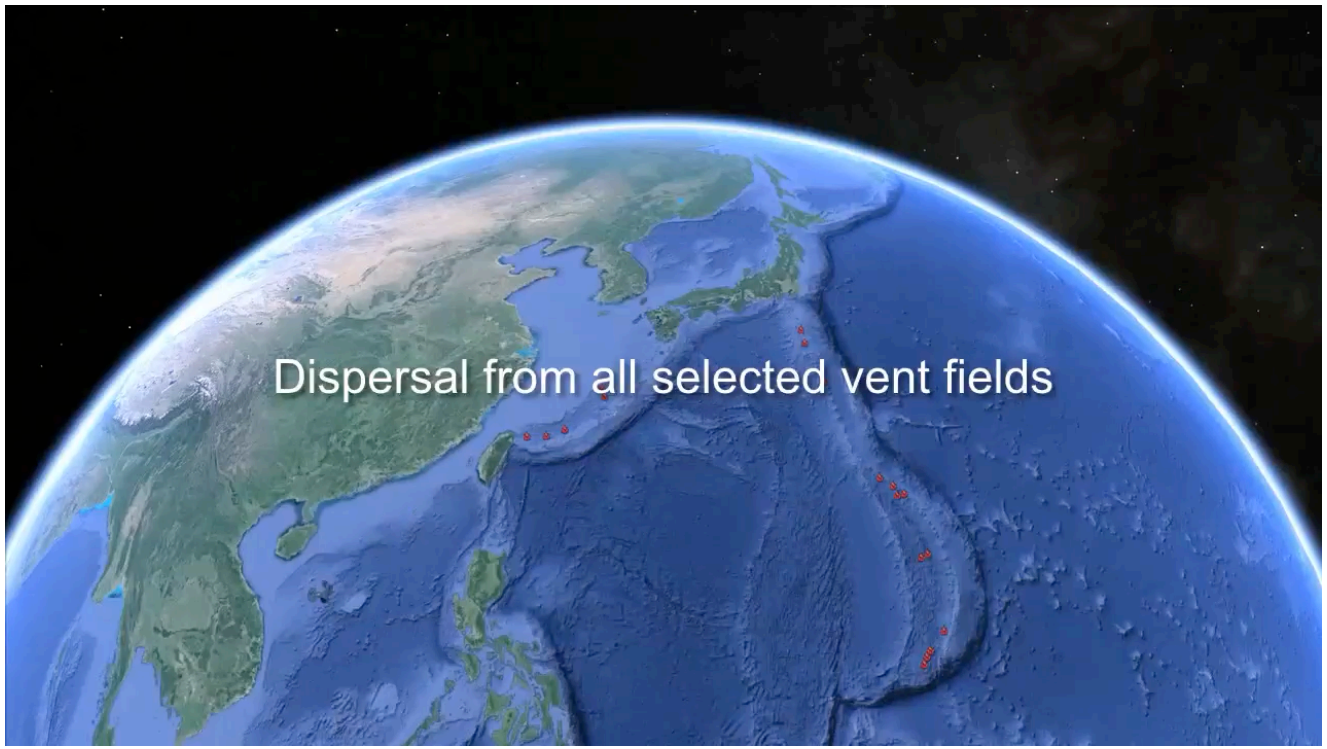




Video from GODAC



Introduction: My Research



Supplimental Figure from Mitarai *et al.,* 2016

Vent ‘Communities’

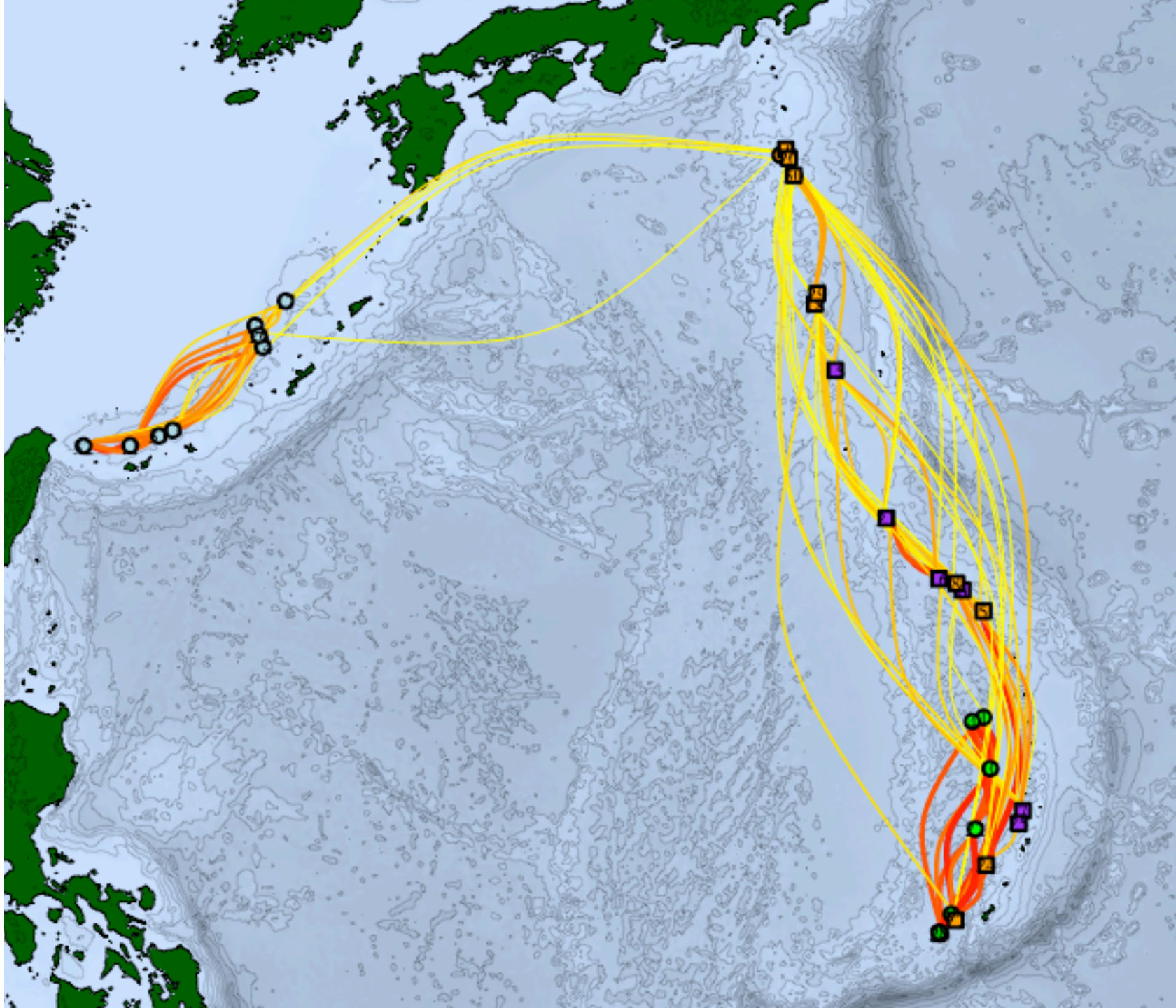
Community 1

Community 2

Community 3

Community 4

Weighted edge value



https://www.visitokinawa.jp/information/karate

Overview

Primary example: Karate Club!

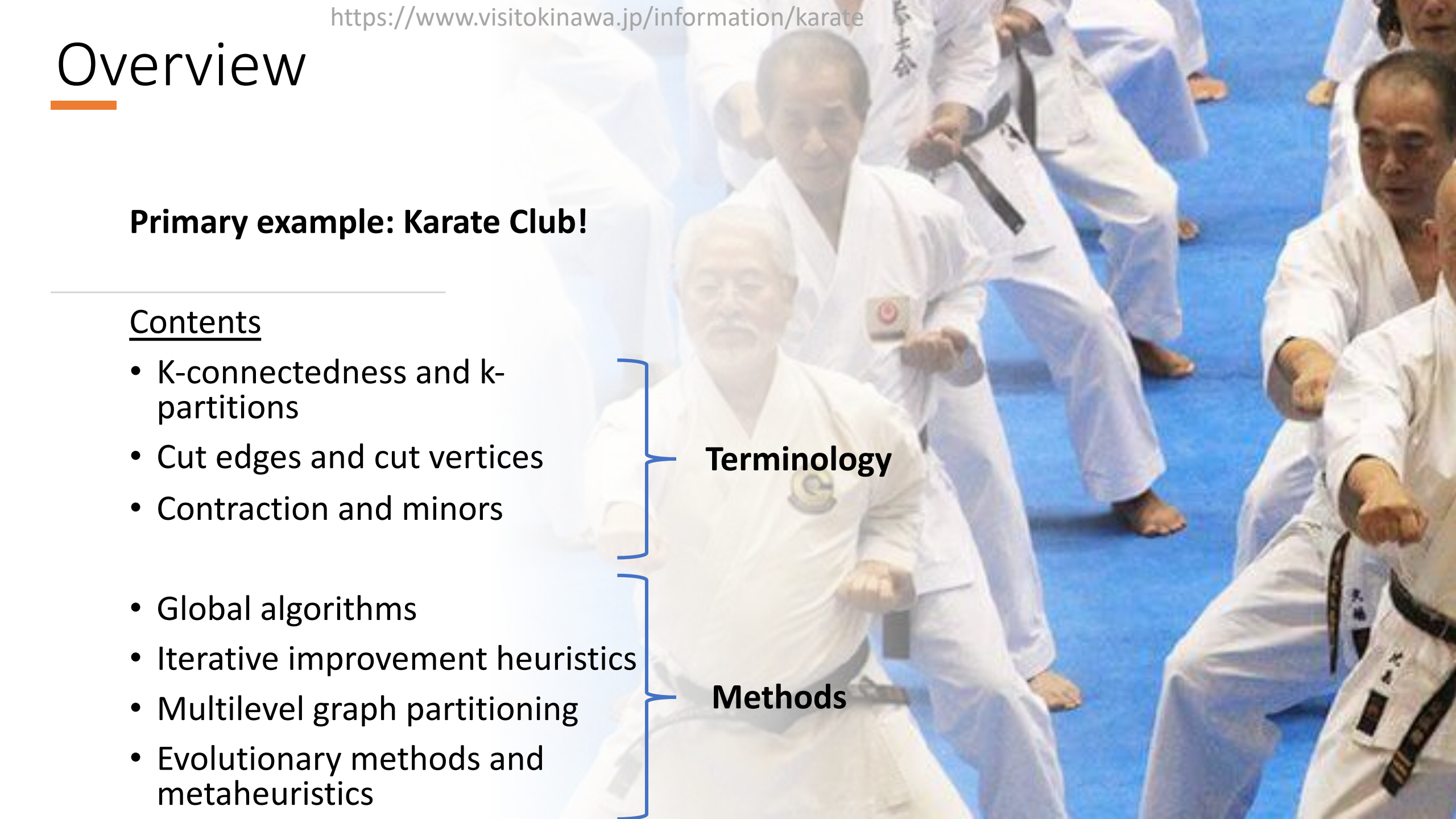
Contents

- K-connectedness and k-partitions
- Cut edges and cut vertices
- Contraction and minors

Terminology

- Global algorithms
- Iterative improvement heuristics
- Multilevel graph partitioning
- Evolutionary methods and metaheuristics

Methods



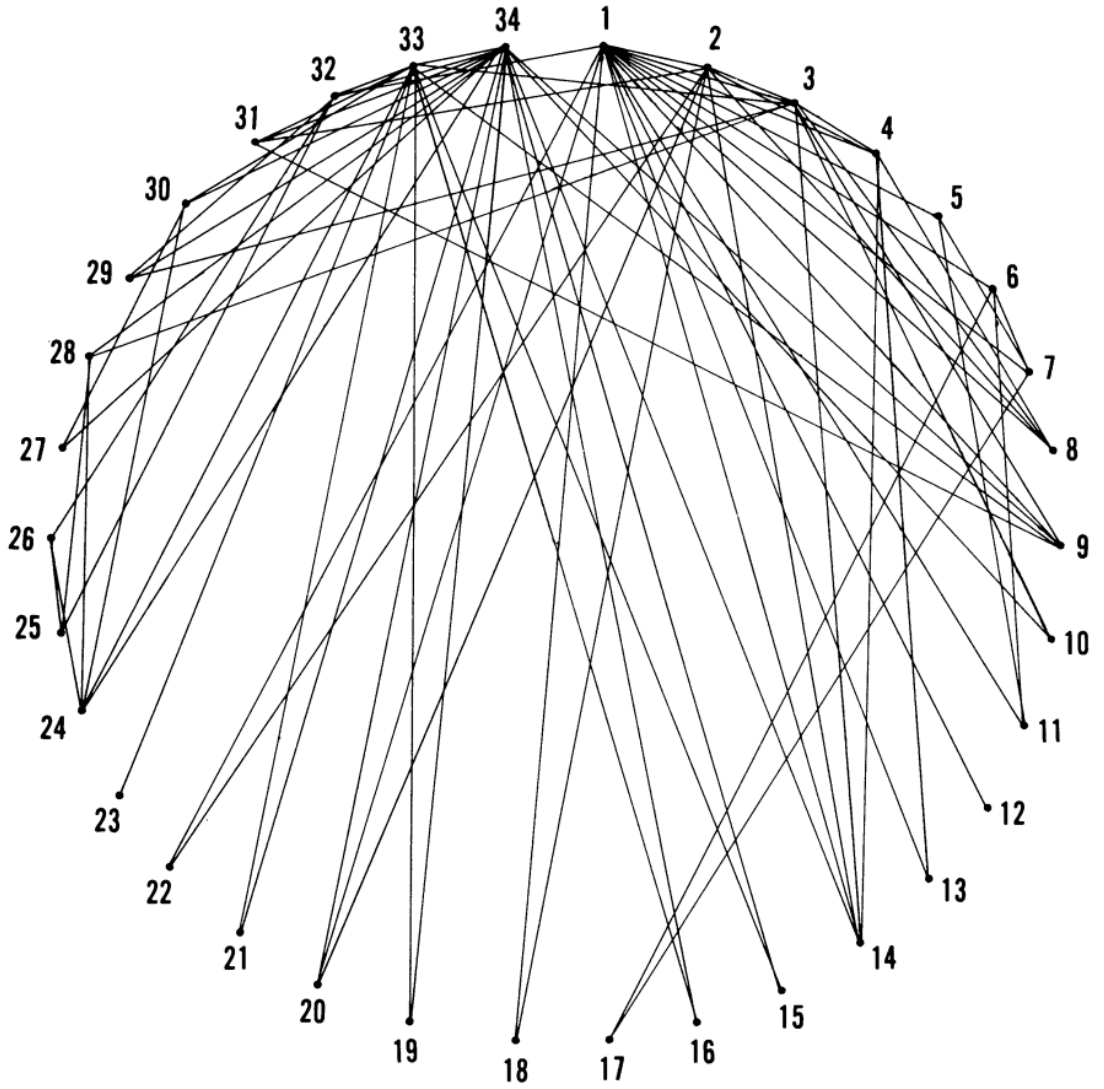
Zachary's Karate Club

Zachary W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33, 452-473.

- Edges = Social Interactions (outside of the club)

- Social network + Conflict
  = **Partitioning!**

FIGURE 1
Social Network Model of Relationships in the Karate Club

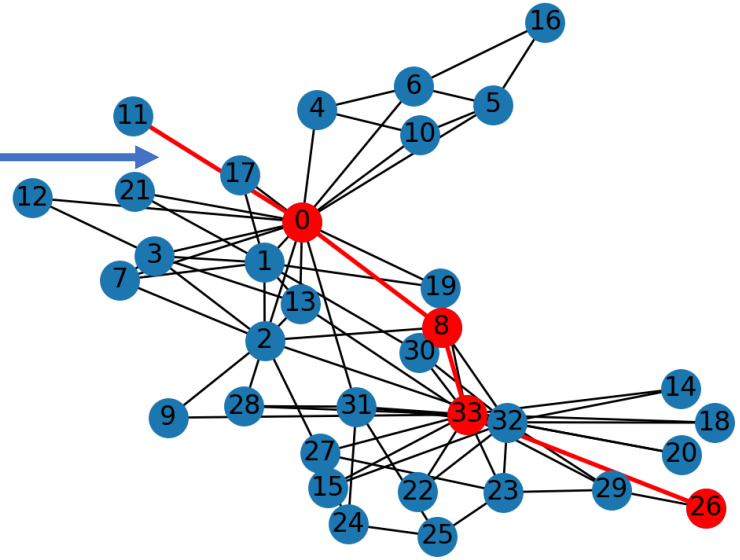


This is the graphic representation of the social relationships among the 34 individuals in the karate club. A line is drawn between two points when the two individuals being represented consistently interacted in contexts outside those of karate classes, workouts, and club meetings. Each such line drawn is referred to as an edge.

K-connectedness and k-partitions

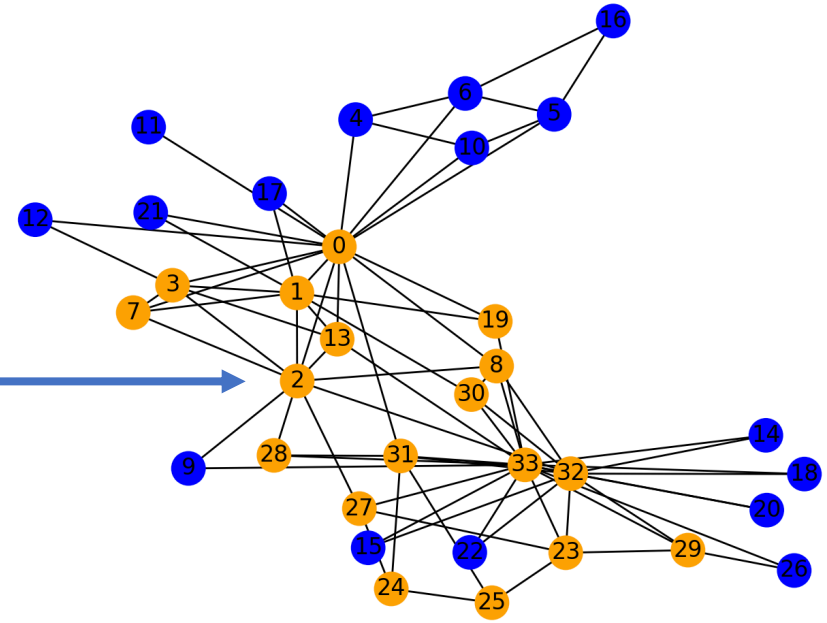
- Connected = Path between all vertices
  - K connected = number of vertices that need to be removed to make graph disconnected
  - What does this tell us about the Karate club?

1

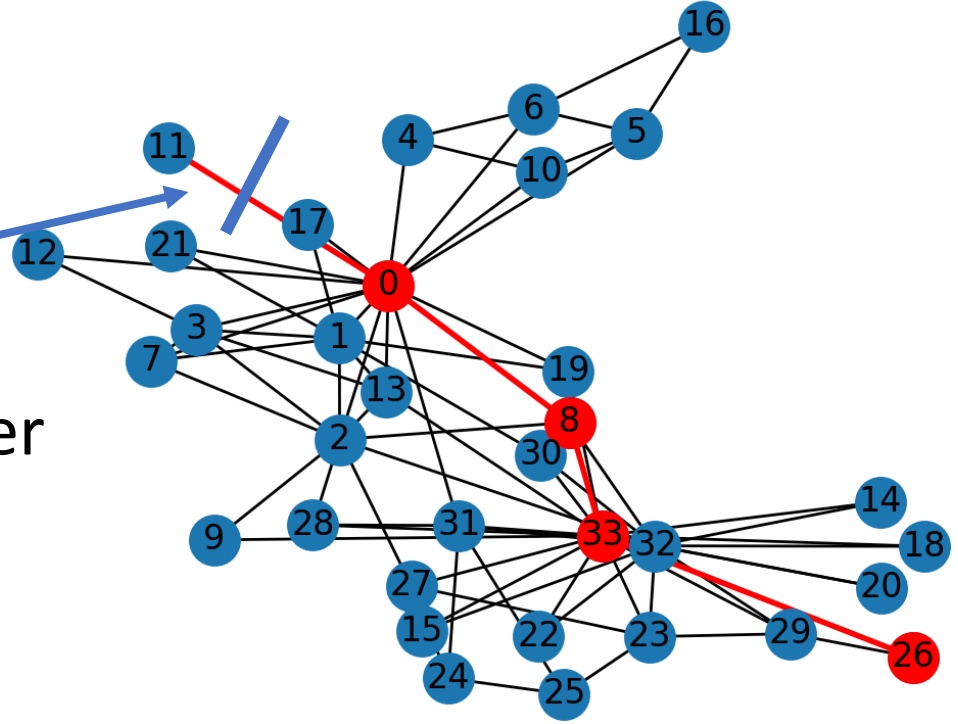


- Define the partitioning of a graph based on the desired K-connectedness
  - To disconnect the yellow subgraph, you would have to remove **3** vertices

K = 3



Cut Edges/Vertices



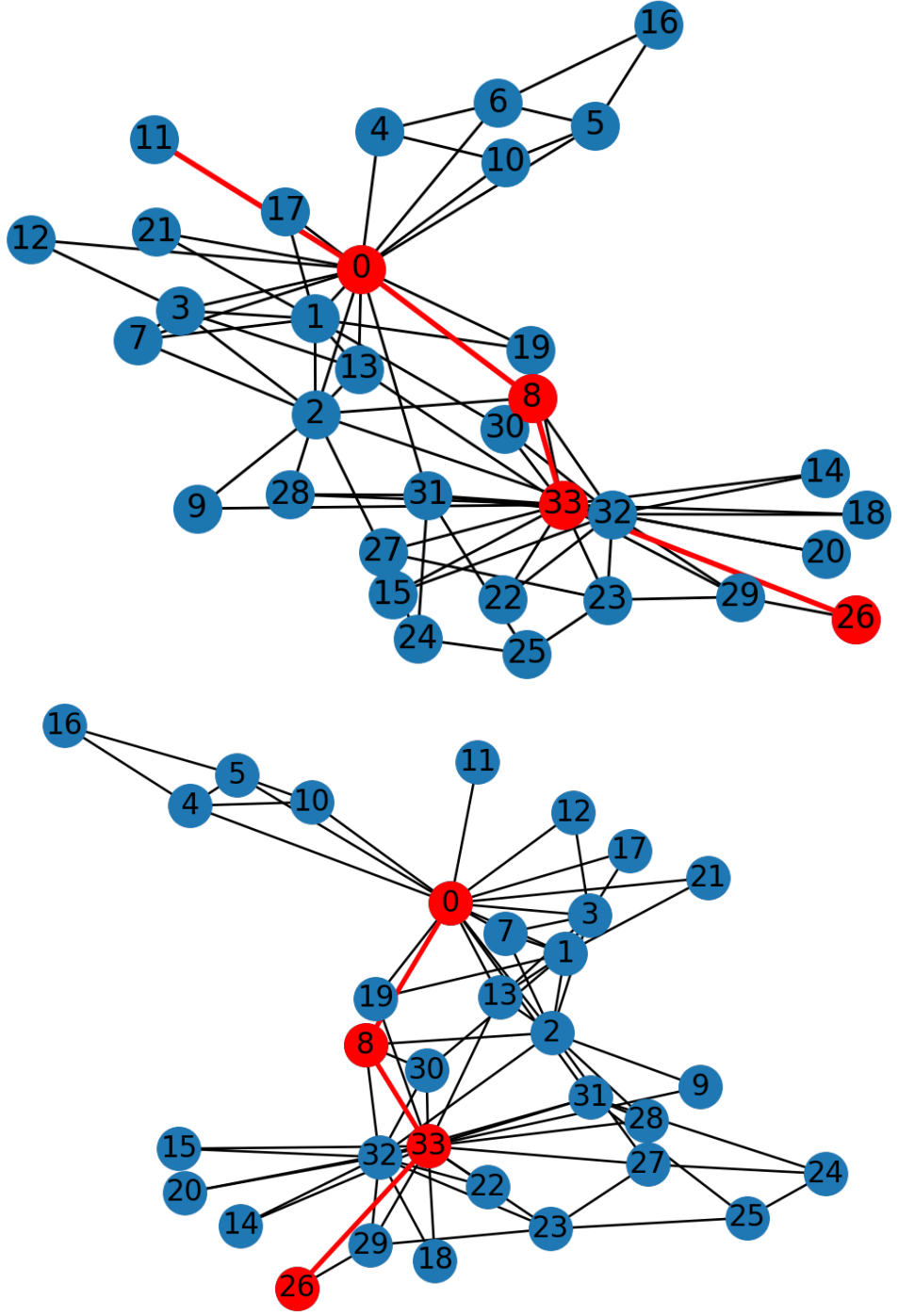
- Removing a cut edge/node increases number of **components** (components ≠ clusters)
  - The cut edge is clear but what about the cut vertecise?

- Impotrant bridges between communities
  - Cut size and stability?
  - ‘Mr. High’ and ‘John’ are cut vertices

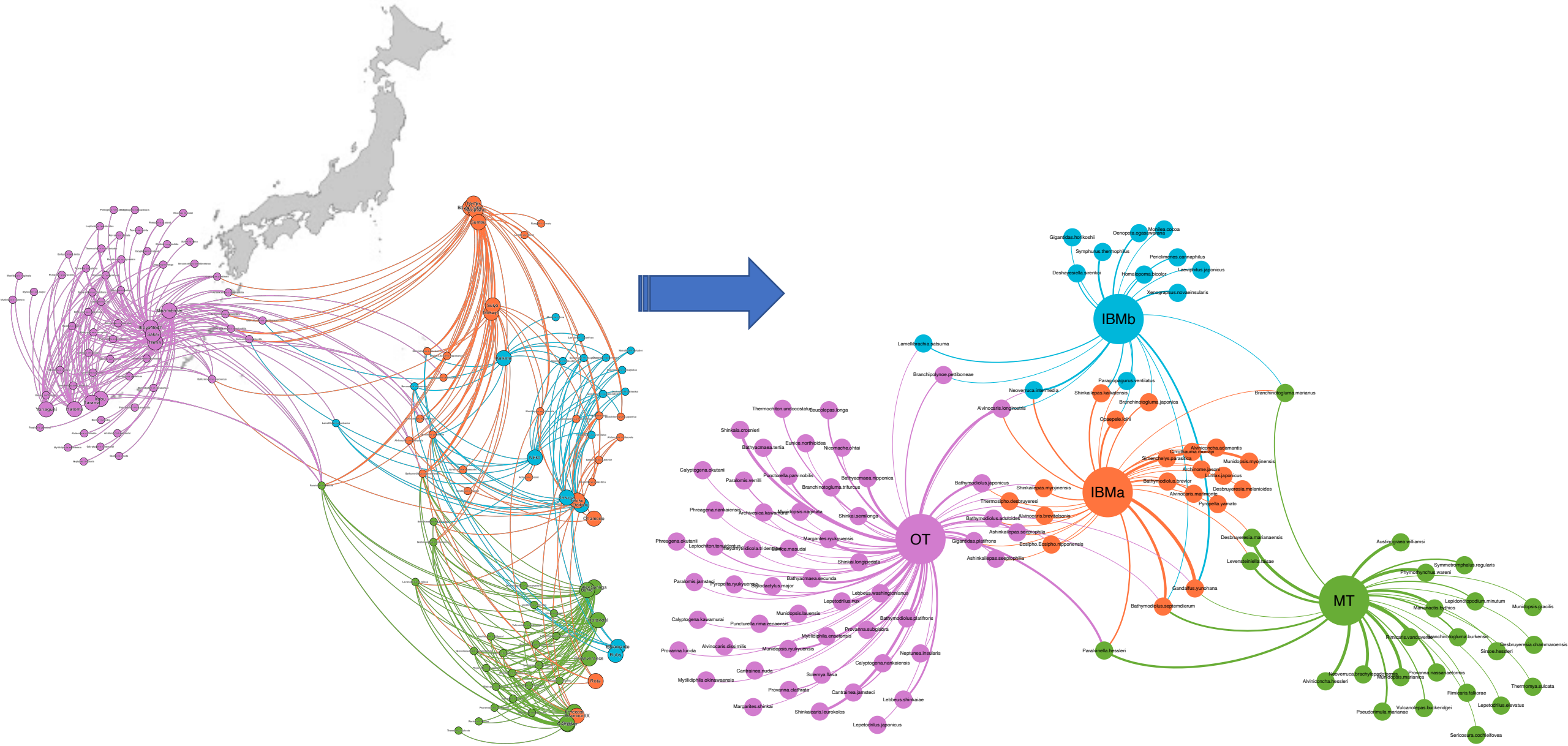
Contraction and minors

- Contracting is like combining vertices or edges
  - The information is not always combined

- Reasons to contract:
  - Visual simplification
  - Higlight partitions (vent network)
  - Multi-level partitioning



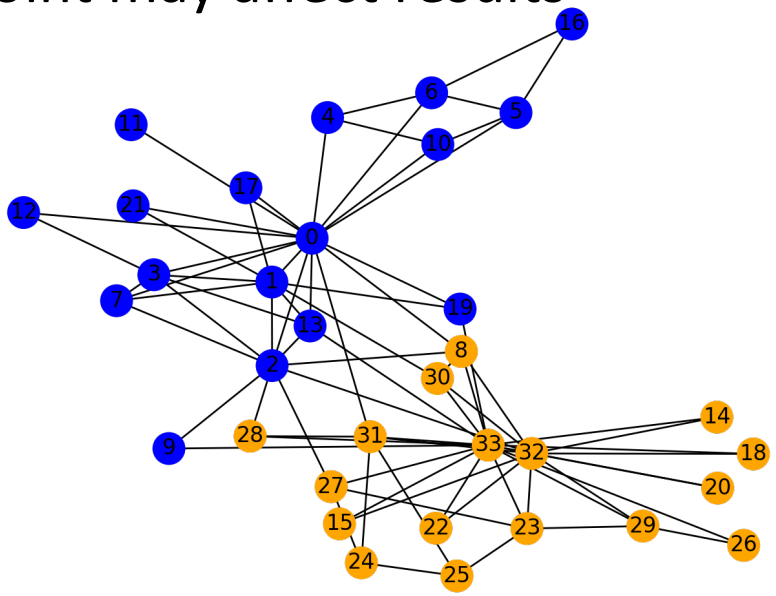
Contraction of vents by community

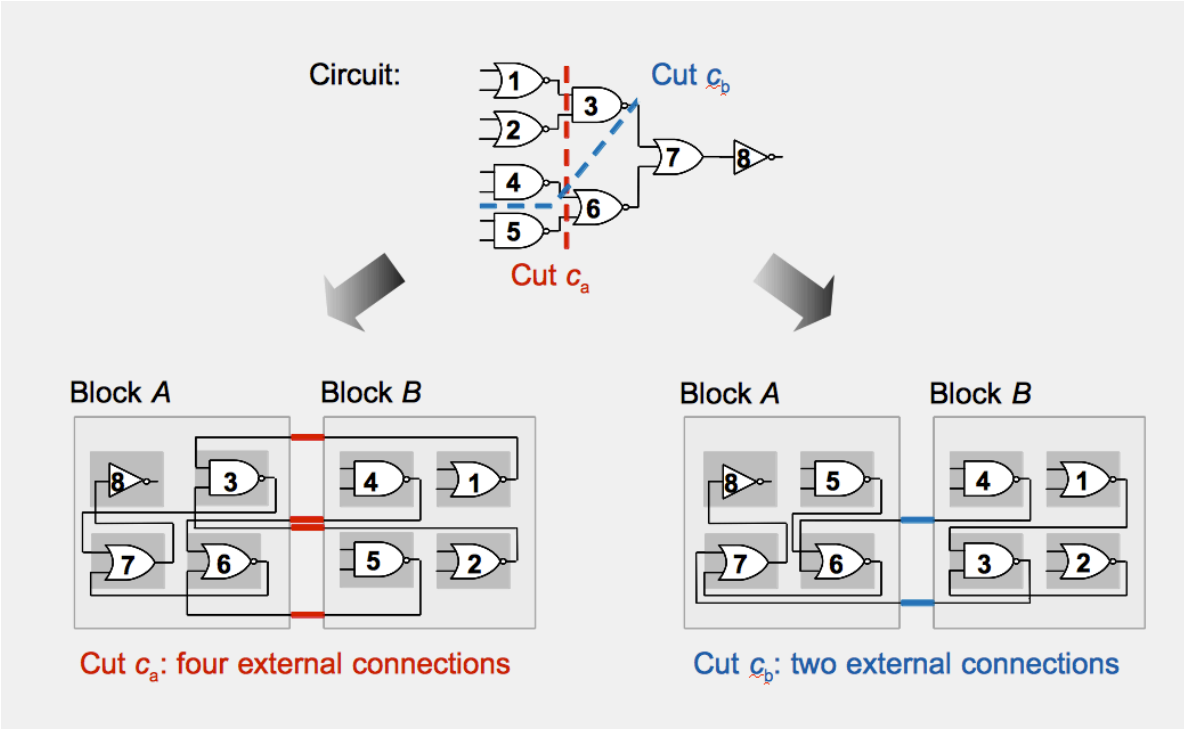


Local:
Iterative improvement heuristics

- Heuristic means they find approximate (not perfect) solutions to the partitioning
  - Concept of accuracy vs efficiency

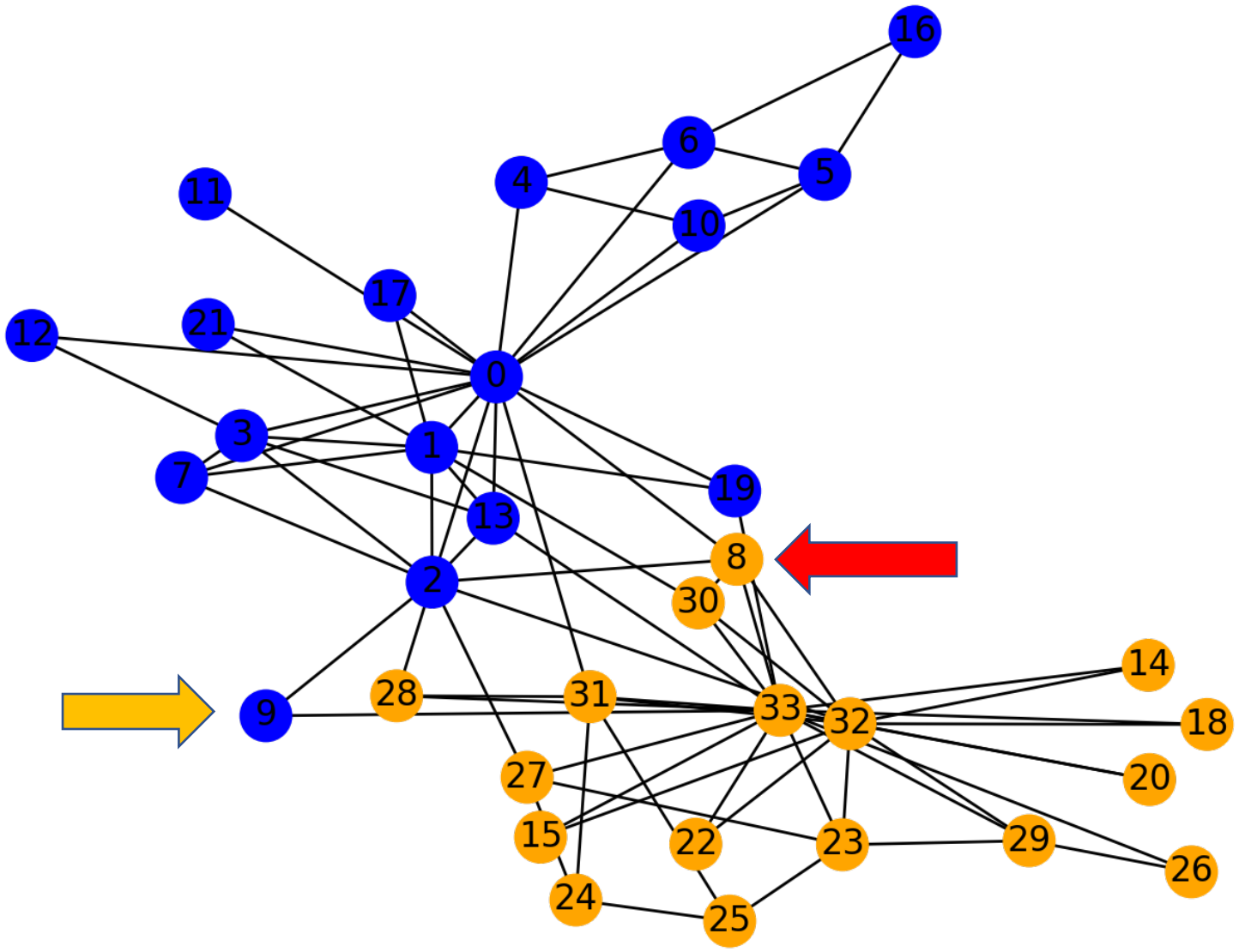
- Partitioning Example: The Kernighan-Lin algorithm Intuitive but slow
  - Start point may affect results





Kahng, A.B., Lienig, J., Markov, I.L. and Hu, J., 2011. *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media.

Local:
Iterative improvement heuristics



ID -1

TABLE 3

EVALUATION OF THE HYPOTHESES

INDIVIDUAL NUMBER IN MATRIX *C*	FACTION MEMBERSHIP FROM DATA	FACTION MEMBERSHIP AS MODELED	HIT/ MISS	CLUB AFTER SPLIT FROM DATA	CLUB AFTER SPLIT AS MODELED	HIT/ MISS
1	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
2	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
3	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
4	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
5	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
6	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
7	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
8	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
9	John	John	Hit	Mr. Hi's	Officers'	Miss
10	John	John	Hit	Officers'	Officers'	Hit
11	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
12	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
13	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
14	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
15	John	John	Hit	Officers'	Officers'	Hit
16	John	John	Hit	Officers'	Officers'	Hit
17	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
18	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
19	John	John	Hit	Officers'	Officers'	Hit
20	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
21	John	John	Hit	Officers'	Officers'	Hit
22	Mr. Hi	Mr. Hi	Hit	Mr. Hi's	Mr. Hi's	Hit
23	John	John	Hit	Officers'	Officers'	Hit
24	John	John	Hit	Officers'	Officers'	Hit
25	John	John	Hit	Officers'	Officers'	Hit
26	John	John	Hit	Officers'	Officers'	Hit
27	John	John	Hit	Officers'	Officers'	Hit
28	John	John	Hit	Officers'	Officers'	Hit
29	John	John	Hit	Officers'	Officers'	Hit
30	John	John	Hit	Officers'	Officers'	Hit
31	John	John	Hit	Officers'	Officers'	Hit
32	John	John	Hit	Officers'	Officers'	Hit
33	John	John	Hit	Officers'	Officers'	Hit
34	John	John	Hit	Officers'	Officers'	Hit
TOTALS		34 hits, 0 misses 100% hits, 0% misses			33 hits, 1 miss 97% hits, 3% misses	

This table gives the results of the NETFLOW runs used to test the two hypotheses (see pp. 462 ff.). The faction membership (column 2) and the club joined after the fission (column 5) entries were taken from the ethnographic data. These columns merely state what the individuals actually did. Column 3 gives the faction membership as predicted by the model (based on which side of the minimum cut the individual was placed). Column 4 gives the accuracy of each of these predictions. The model was 100% accurate in predicting faction membership. Column 6 gives the membership in the two clubs formed after the fission, again as predicted by the model (based on which side of the minimum cut the individual was placed). Column 7 gives the results of these predictions. The model was 97% accurate in predicting club membership after the split. Thus, both hypotheses can be accepted.

https://neo4j.com/blog/graph-algorithms-neo4j-louvain-modularity/

Local:
Iterative improvement heuristics

- Community detection example:
- Modularity Maximization

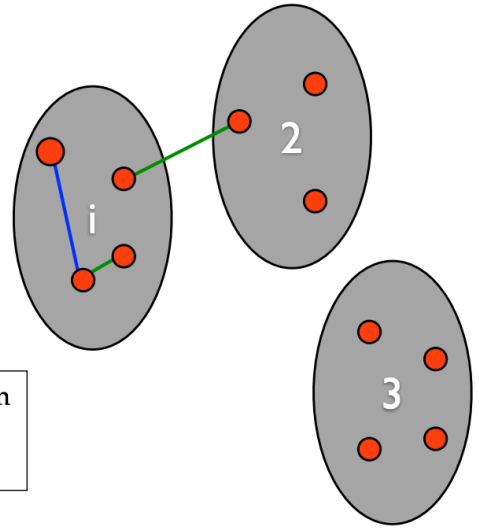
Modularity

e_{ii} = % edges in module i

$e_{ii}=|\{(u,v) : u \in V_i, v \in V_i, (u,v)\in E\}| / |E|$

a_i = % edges with at least 1 end in module i

$a_i=|\{(u,v) : u \in V_i, (u,v) \in E\}| / |E|$



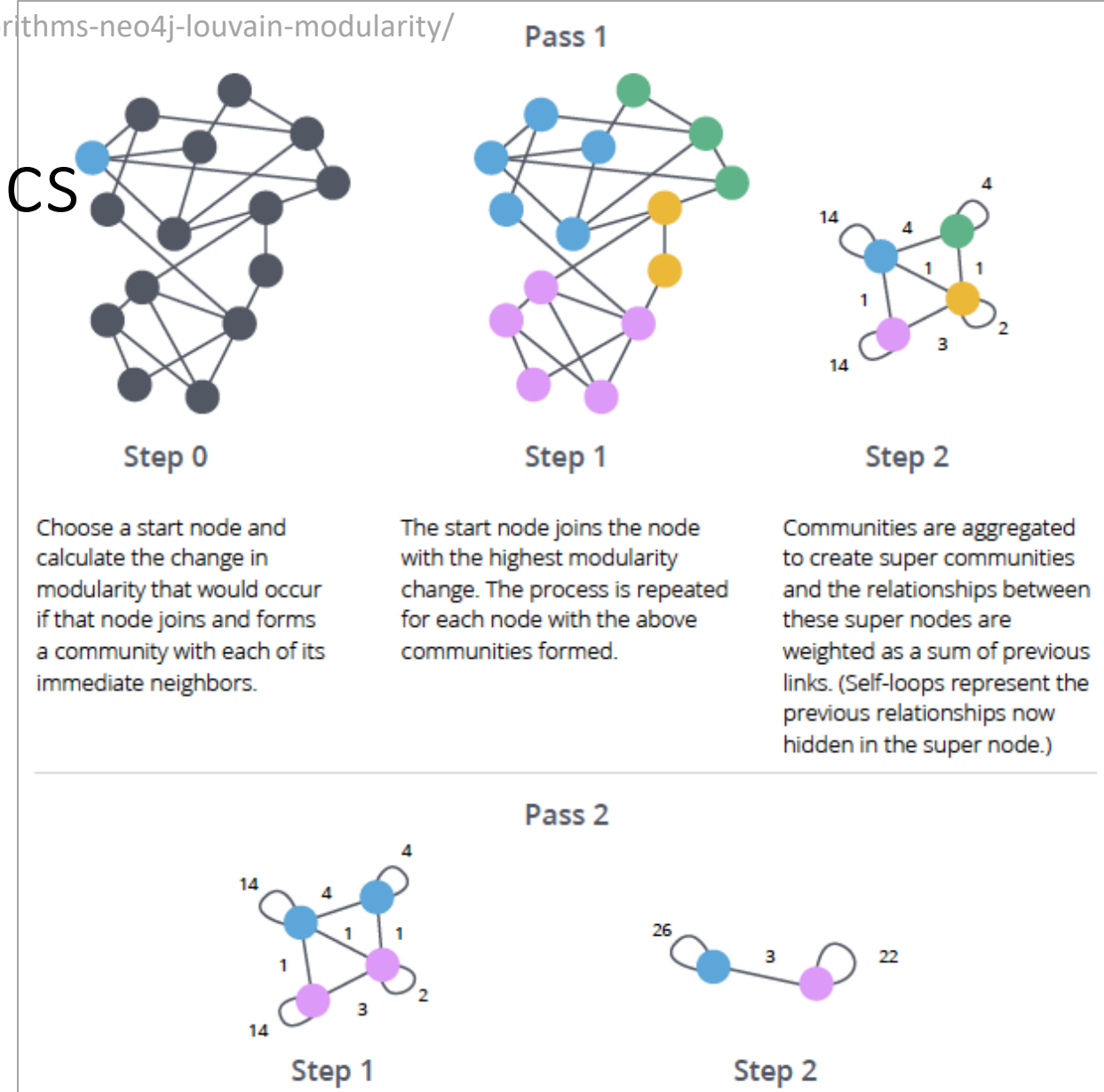
Modularity is:

$$Q = \sum_{i=1}^{k} (e_{ii} - a_i^2)$$

probability a random edge would fall into module i

probability edge is in module i

High modularity ⇒ more edges within the module that you expect by chance.



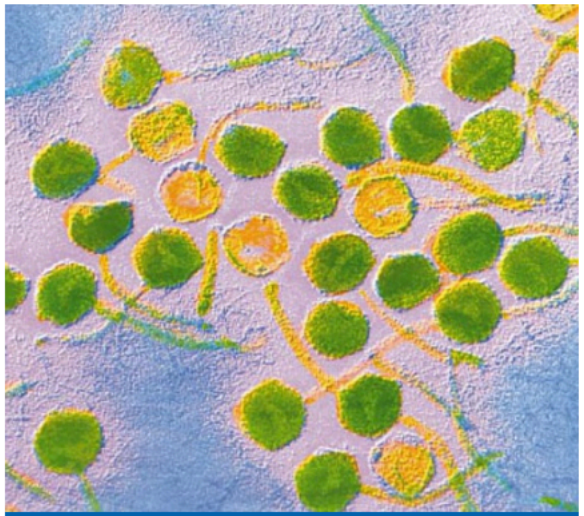
Most Biological Networks are Modular

impacts

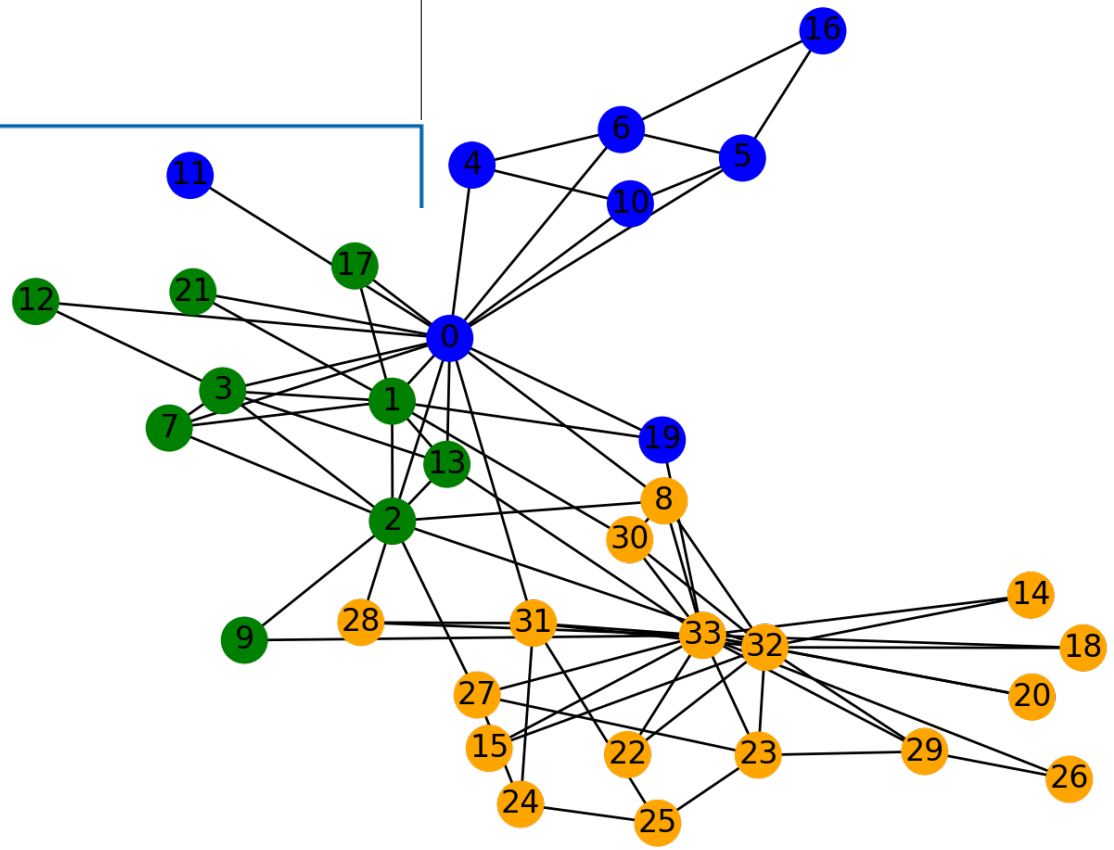
From molecular to modular cell biology

Leland H. Hartwell, John J. Hopfield, Stanislas Leibler and Andrew W. Murray

Box 2 A decision-making module in bacteriophage lambda



False-colour transmission electron micrograph of lambda bacteriophages (×13,500).



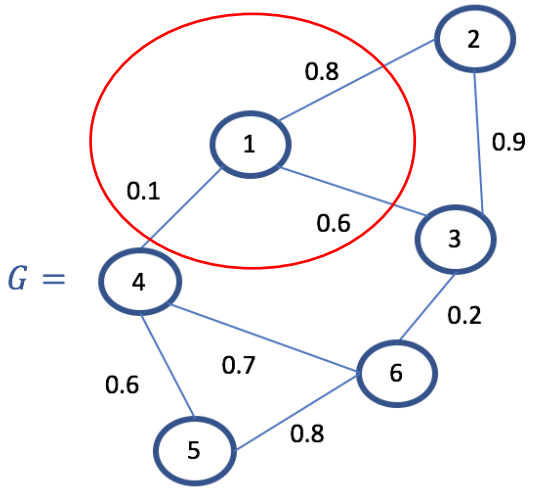
https://towardsdatascience.com/spectral-clustering-82d3cff3d3b7

Global: Spectral Partitioning

$L = D - A$,
where A is the adjacency matrix and D is the degree matrix such that

$$d_i = \sum_{\{j|(i,j)\epsilon E\}} w_{ij} \qquad Thus, L_{ij} = \begin{cases} d_i & if\ i = j \\ -w_{ij} & if\ (i,j)\epsilon E \\ 0 & if\ (i,j) \notin E \end{cases}$$

$G =$

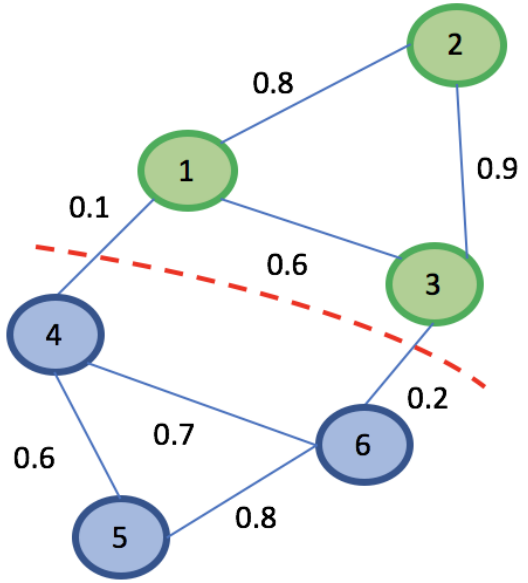


$L =$

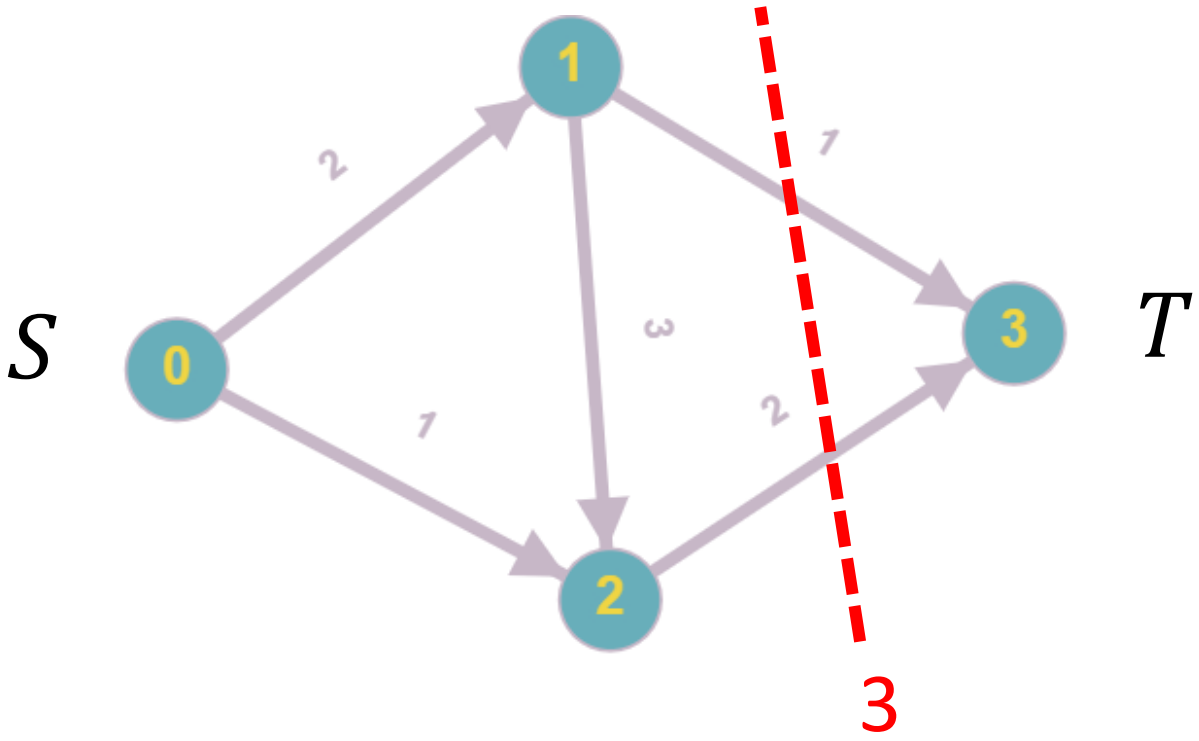
	1	2	3	4	5	6
1	1.5	-0.8	-0.6	-0.1	0	0
2	-0.8	1.7	-0.9	0	0	0
3	-0.6	-0.9	1.7	0	0	-0.2
4	-0.1	0	0	1.4	-0.6	-0.7
5	0	0	0	-0.6	1.4	-0.8
6	0	0	-0.2	-0.7	-0.8	1.7

1. Calculate eigenvector corresponding to the 2nd eigenvalue to assign values to each node.
2. The 2nd eigenvalue is 0.189 and the corresponding eigenvector v2 = [0.41, 0.44, 0.37, -0.4, -0.45, -0.37]
3. Split the nodes such that all nodes with value > 0 are in one cluster, and all other nodes are in the other cluster.

	v2
1	0.41
2	0.44
3	0.37
4	-0.40
5	-0.45
6	-0.37



Honourable Mention:
Max-flow min-cut theorem



- Maximum flow: largest possible flow from two vertices designated as source S and sink T
- Minimum cut: the cut with smallest possible capacity

Multilevel graph partitioning: Step 1 - Coarsen

Match vertices then contract their edges to create lower-resolution graph.

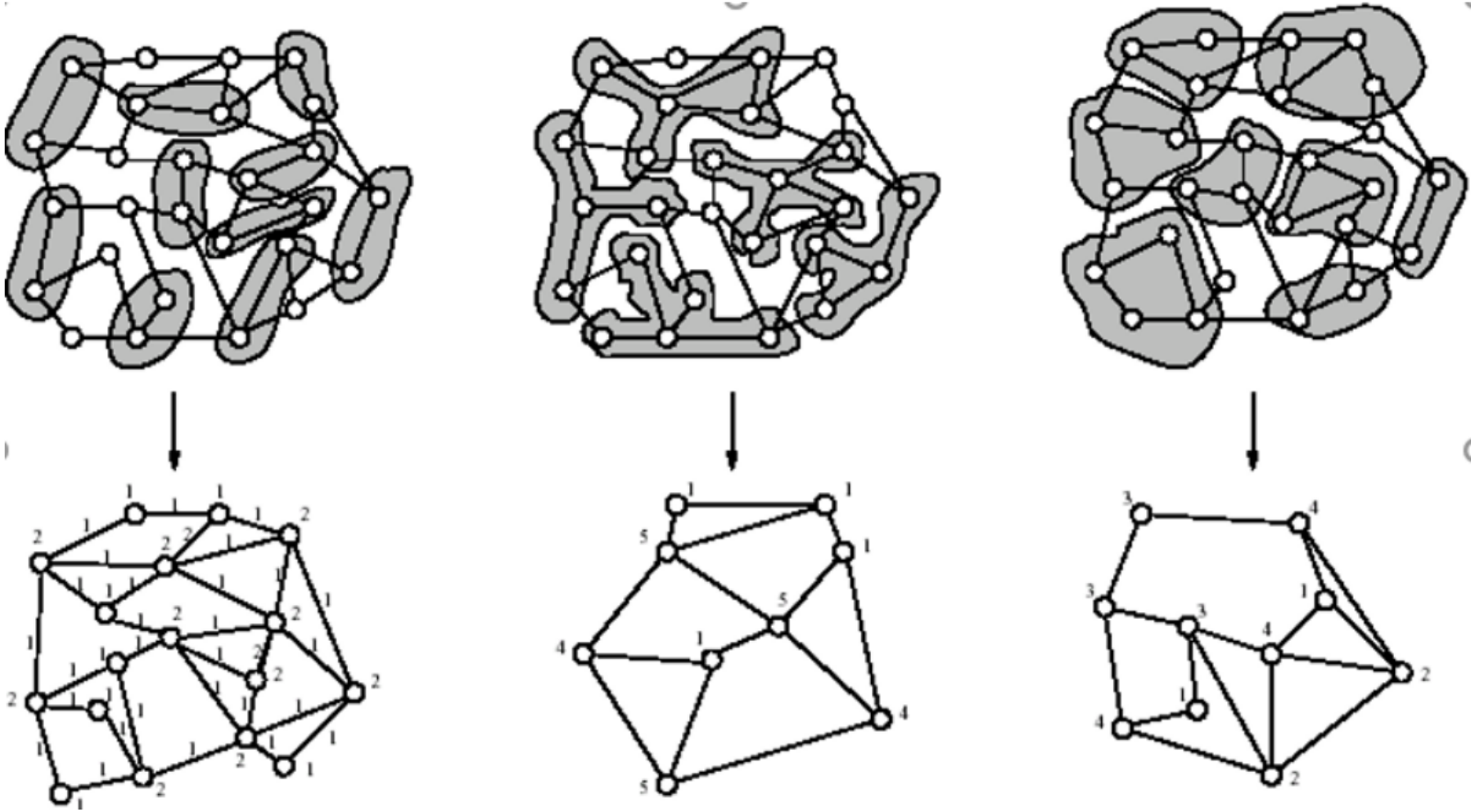


Figure from Yijiang Huang, MIT

Step 2: Partition at coarse level

Step 3: Uncoarsen (and possible refinement at higher level)

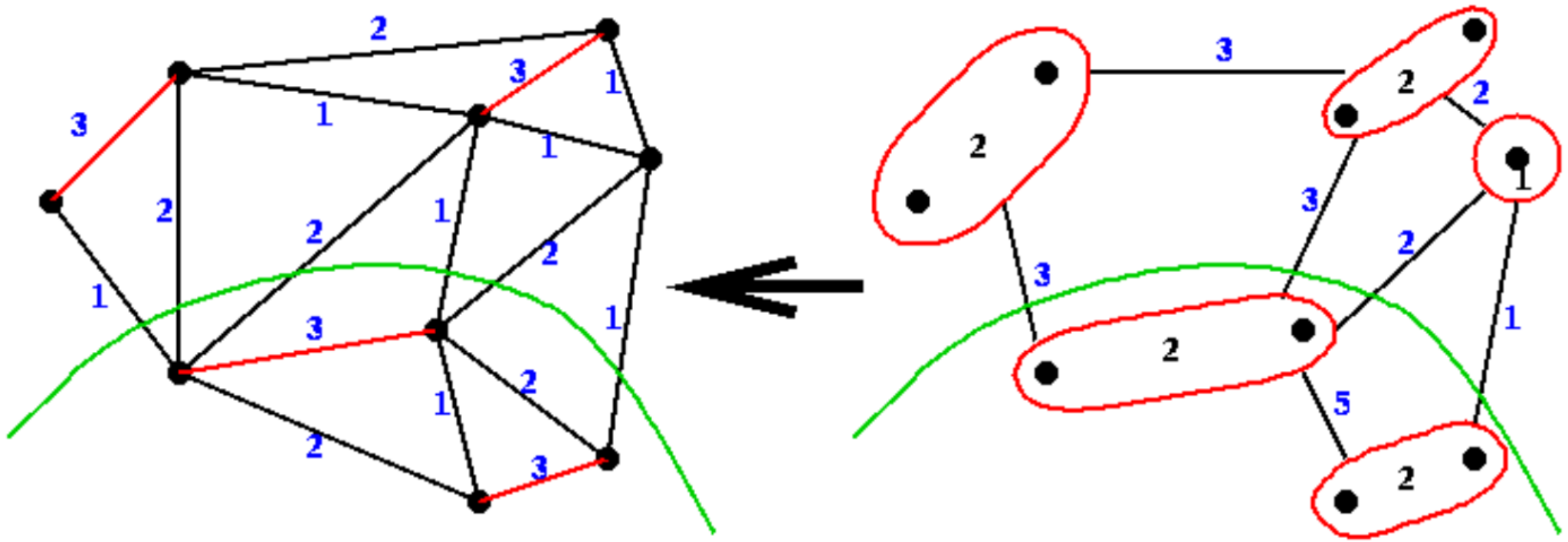


Figure from Yijiang Huang, MIT

Global:
Modularity Maximization

Simulated annealing

1. Start with a random potential solution.
2. Calculate energy/cost/loss of this solution.
3. Calculate energy/cost/loss of neighbouring solutions.
4. With some *probability*, switch to best neighbour or a random neighbour.
5. Iterate until “fixed point”.

e.g. for Travelling Salesman Problem

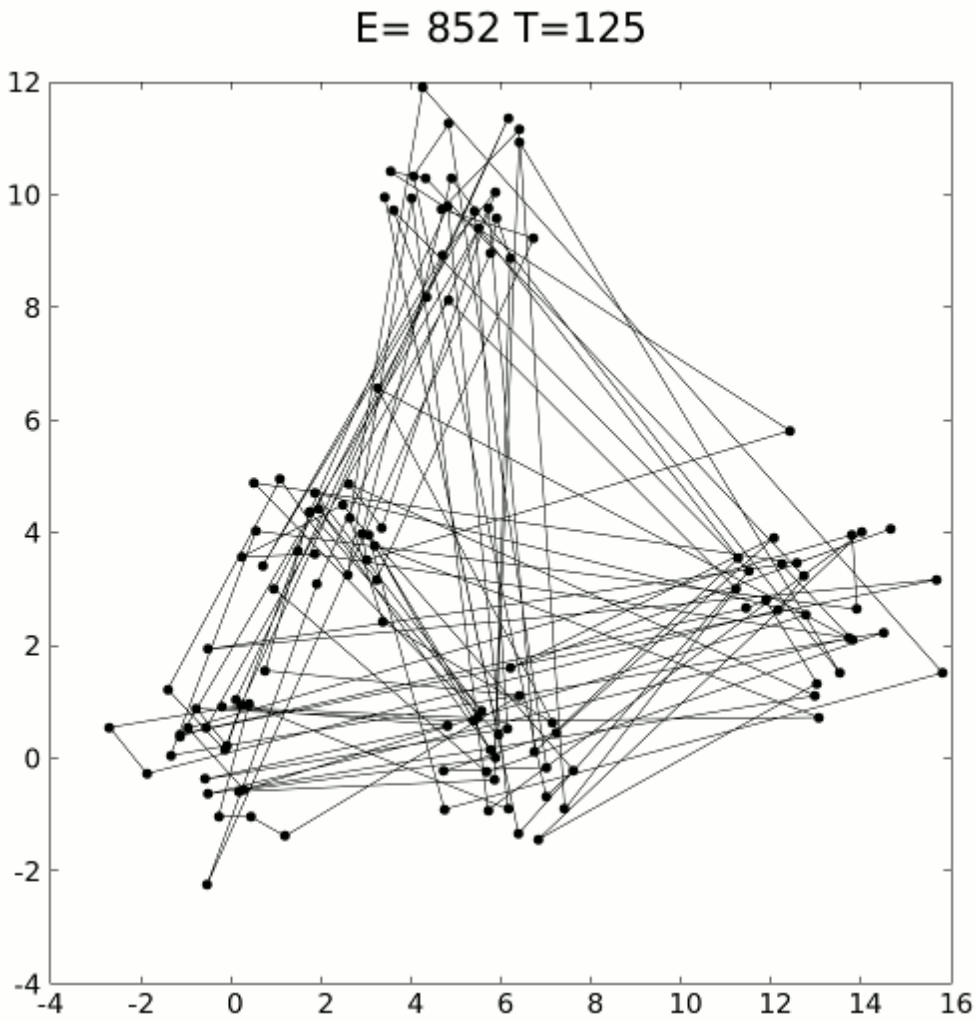


Figure from Wikimedia