Partitioning and Clustering in Electrical Networks

Applications for Reliability and Economics

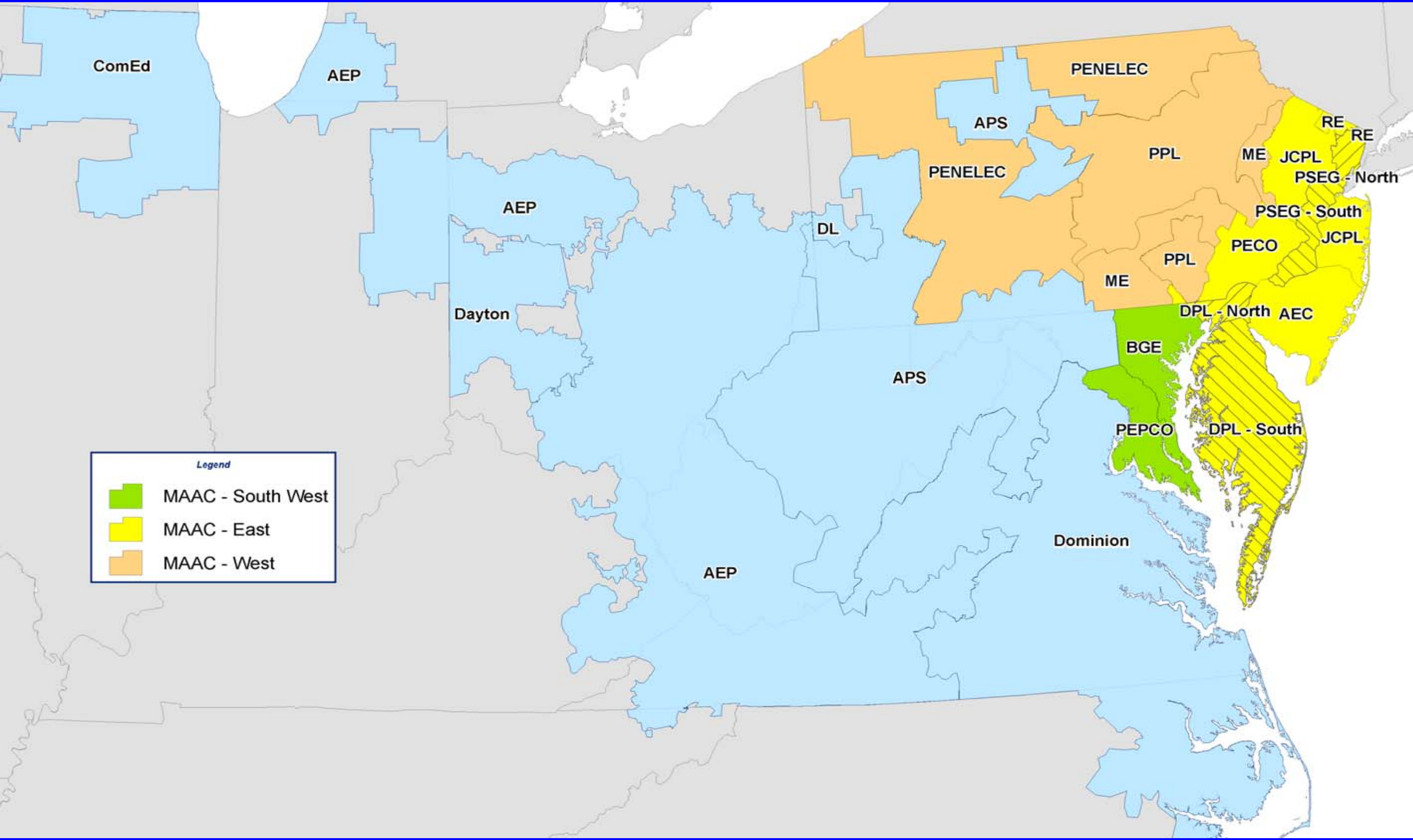
Seth Blumsack
Dept. of Energy and Mineral Engineering
Penn State University
blumsack@psu.edu

Thanks to NSF and PJM for funding support

Outline

- Partitioning the PJM Network
- Structural Metrics for Electrical Networks
  - Topological vs. Electrical Distance Concepts
- Three Clustering Techniques
  - Hierarchical, K-Means, Spectral
  - Genetic Algorithm Refinements
- Clustering Results on the PJM Network
- Further Applications: Economics of “Smart Grids”

The PJM Footprint



Zonal analysis in PJM

PJM Zones are used for multiple planning and operations analyses:

1. *Reliability and resource adequacy assessments*
2. Allocation of ancillary services costs
3. Administration of load curtailment programs
4. Identification and mitigation of market power

A Reliability Application in PJM

- Load delivery test for resource adequacy, RTEP, RPM
- For each zone within PJM:
  - CETO (transfer objective)
  - CETL (transfer capability)
  - Evaluated for seasonal peaks, transmission contingencies (but not inter-zonal congestion), generator EFOR'd

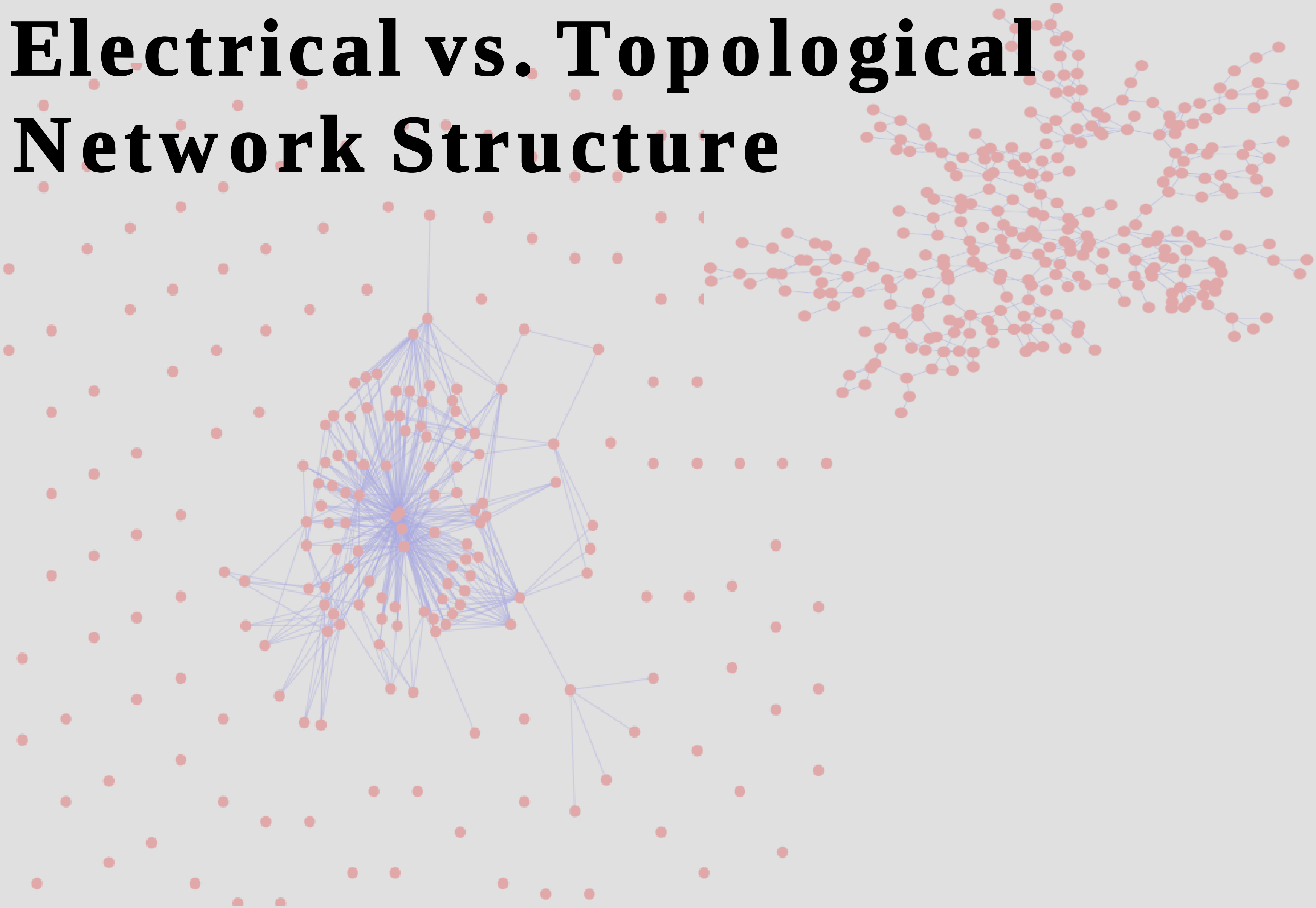
Problems with Zonal Definition

- Zones defined based on TO territories
  - Prior to restructuring, TO territories were tightly knit, few interconnections
- Now, the PJM system is regional (multi-regional?)
  - More long-distance transactions
  - “Big” transmission projects planned (NIEC, AEP 765 kV, APS projects, etc…)
  - Do the current LDA zones reflect the value of these and future projects?

Criteria for a "Good" Zone

- "Closeness"
  - Nodes within a zone should behave similarly, relative to the system outside the zone
- "Connectivity"
  - No significant bottlenecks
  - Robust to single-point contingencies
- Zones should be defined structurally, rather than by "rules"

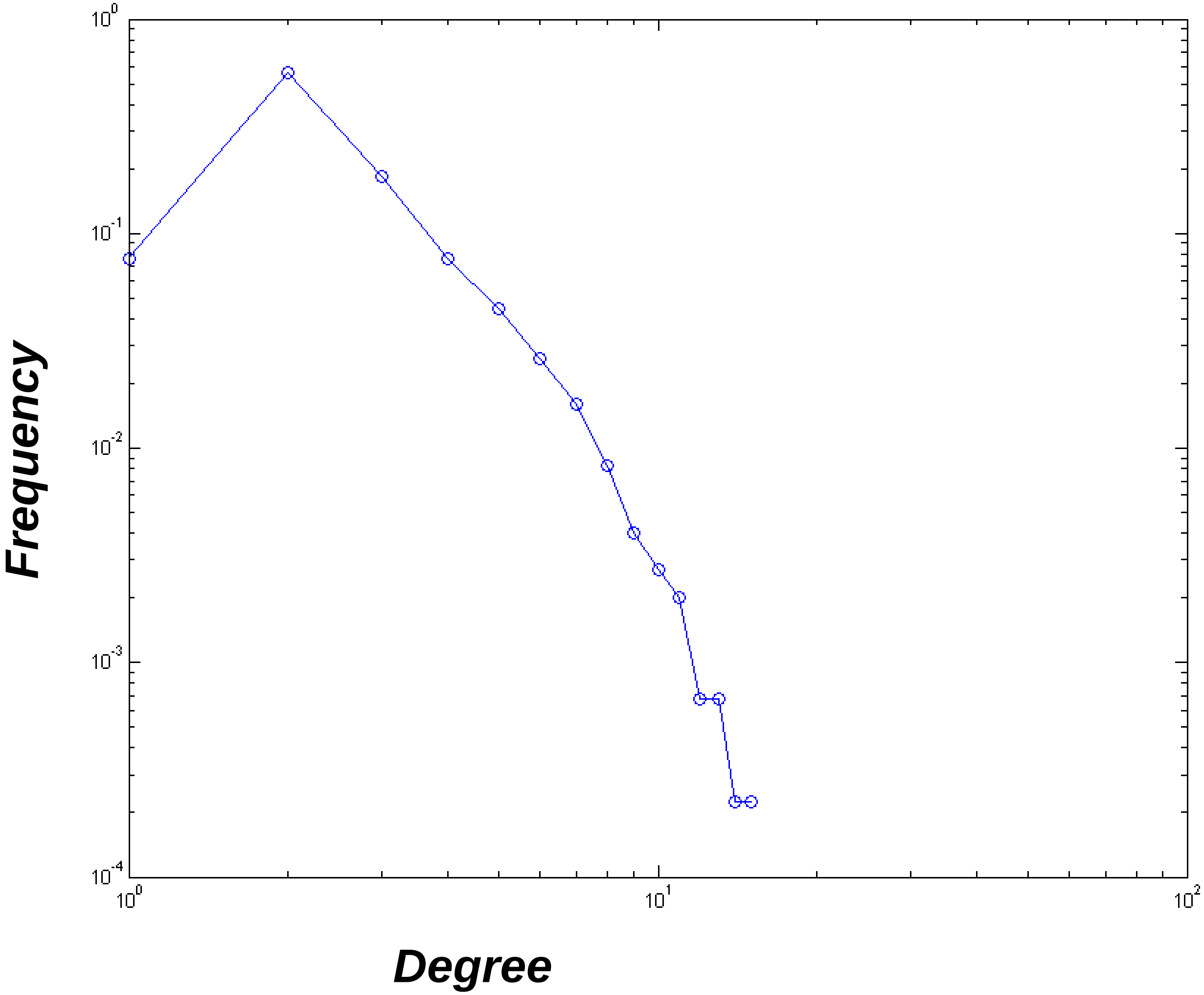
Electrical vs. Topological Network Structure



Topological Distance Metrics

- Buses should be grouped in the same zone if:
  - They are *close to one another* – short distances
  - *Highly connected* – multiple paths
- Topological metric of closeness is the *degree*
  - Degree(k) = number of other nodes that are directly connected to node *k*.
  - Distance (j,k) = number of hops required to get from node *j* to node *k*

Degree distribution for PJM



Electrical distance and centrality

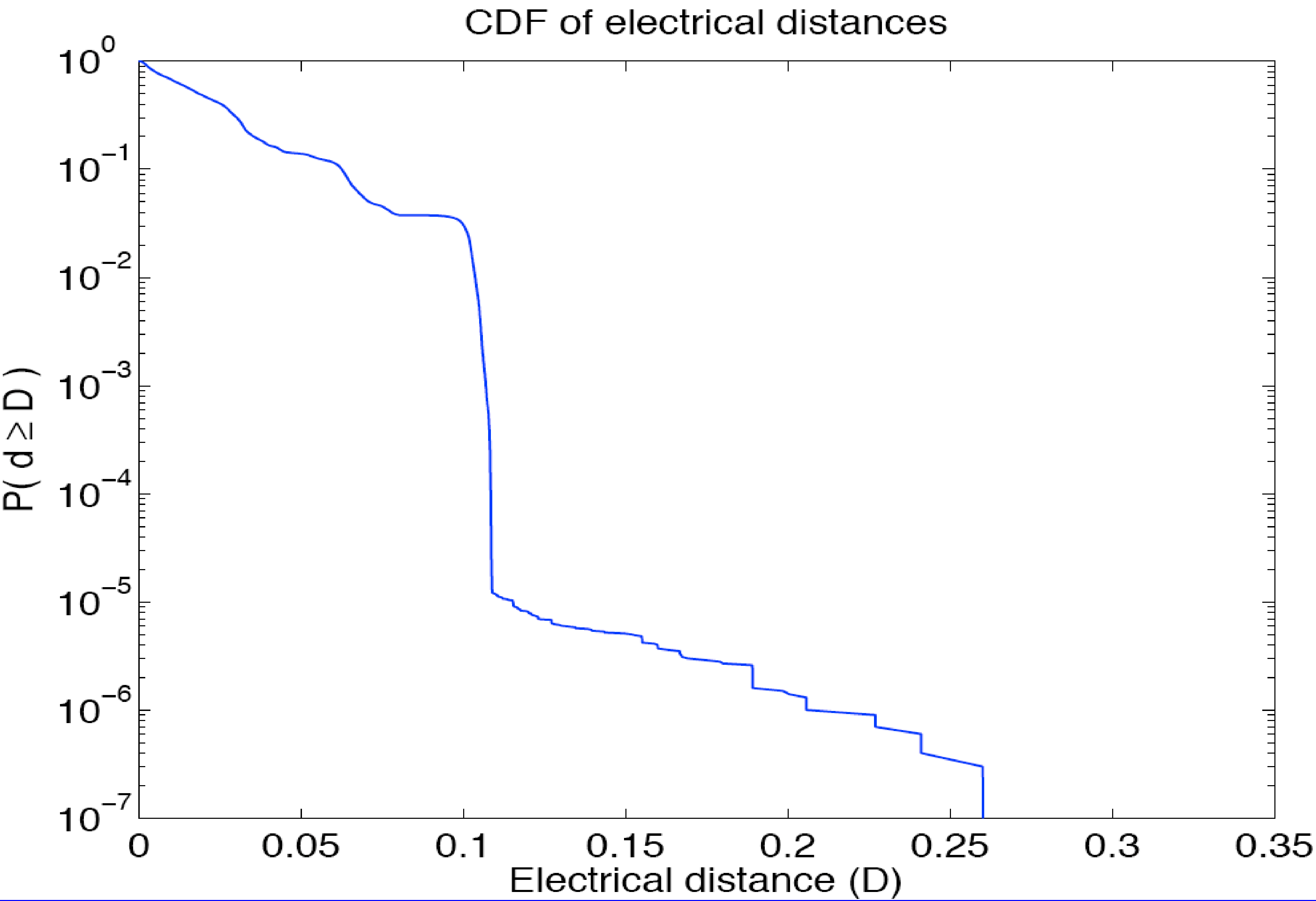
- We can get an estimate of the “distance” between two nodes using the inverse of the admittance matrix:

$$
\begin{aligned}
Z &= Y_{BUS}^{-1} \\
D_{ab} &= |Z_{ab}|
\end{aligned}
$$

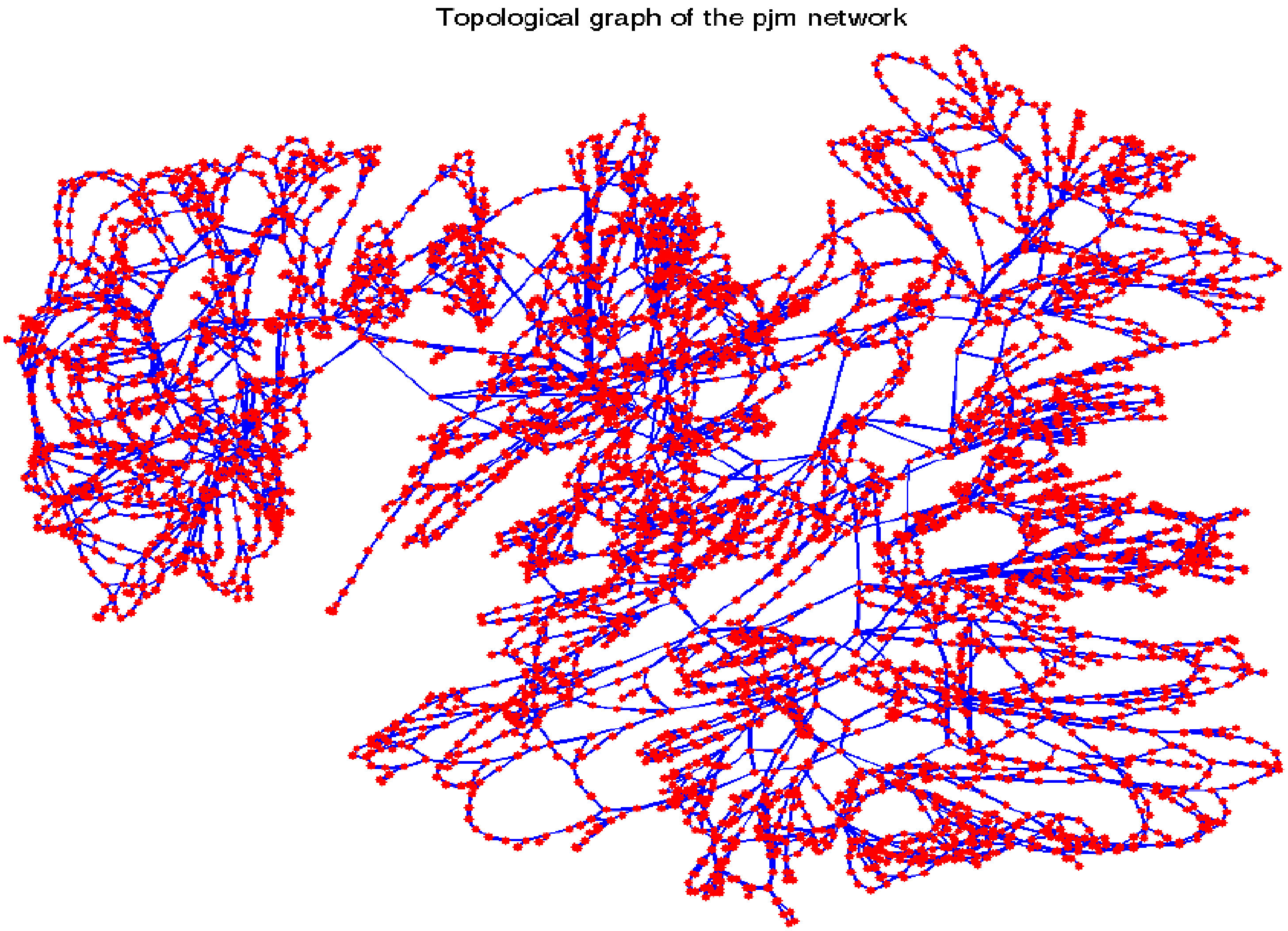
- Then we can get the total closeness (or centrality) of a node via:

$$
C_a = \sum_{b=1}^{n} \frac{1}{D_{ab}}
$$

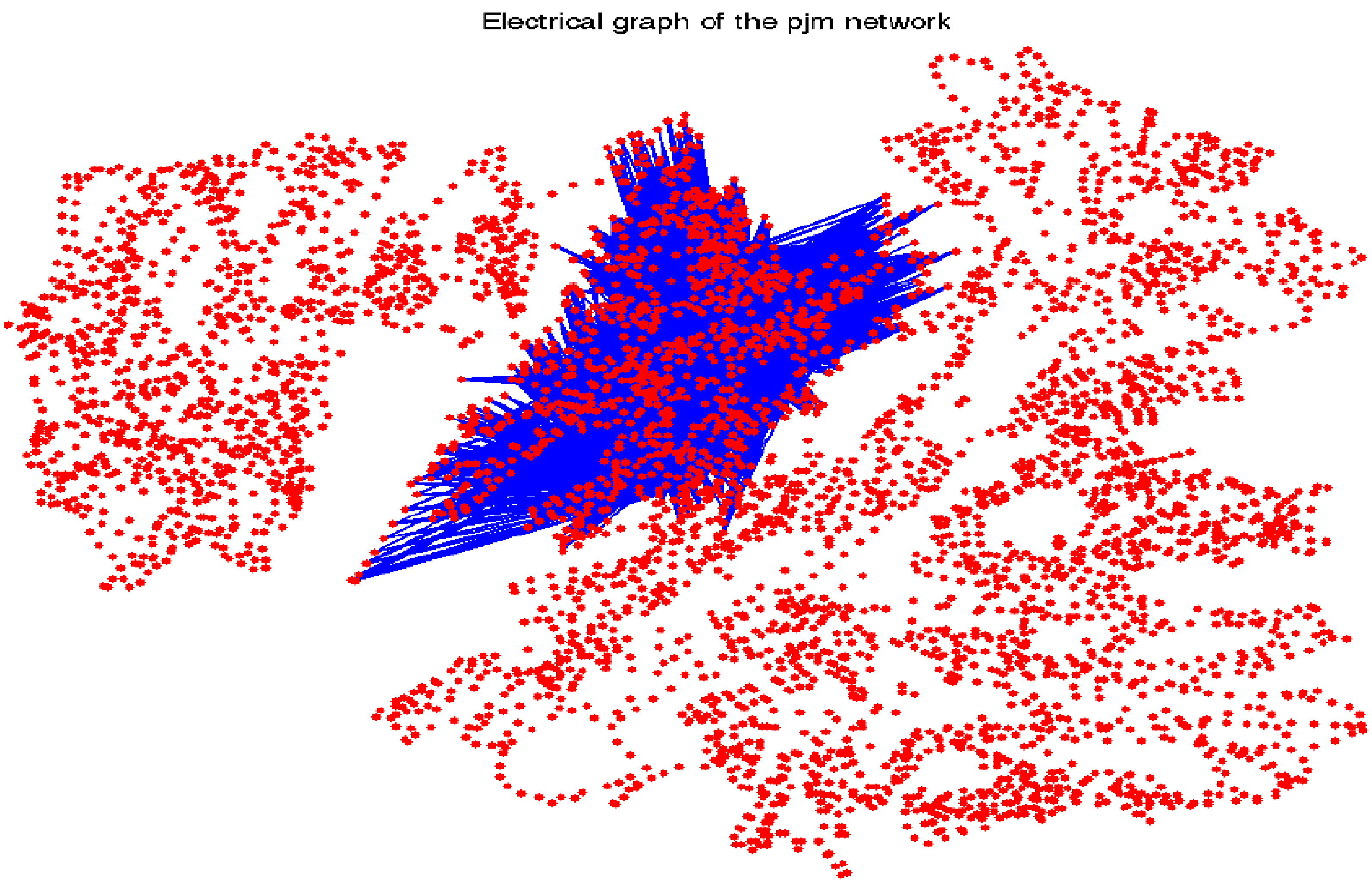
Electrical Distances in PJM



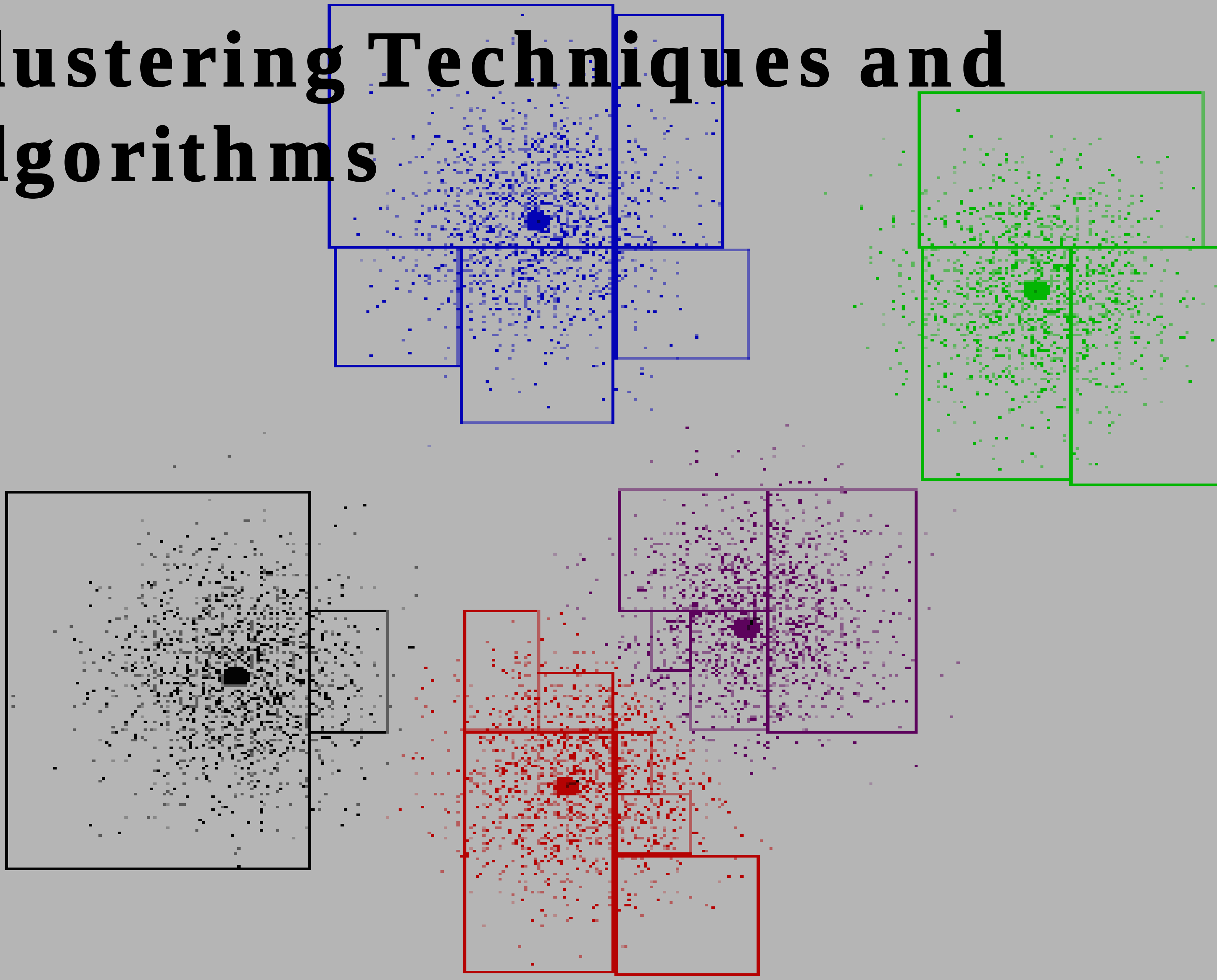
Topological representation of PJM



Electrical representation of PJM



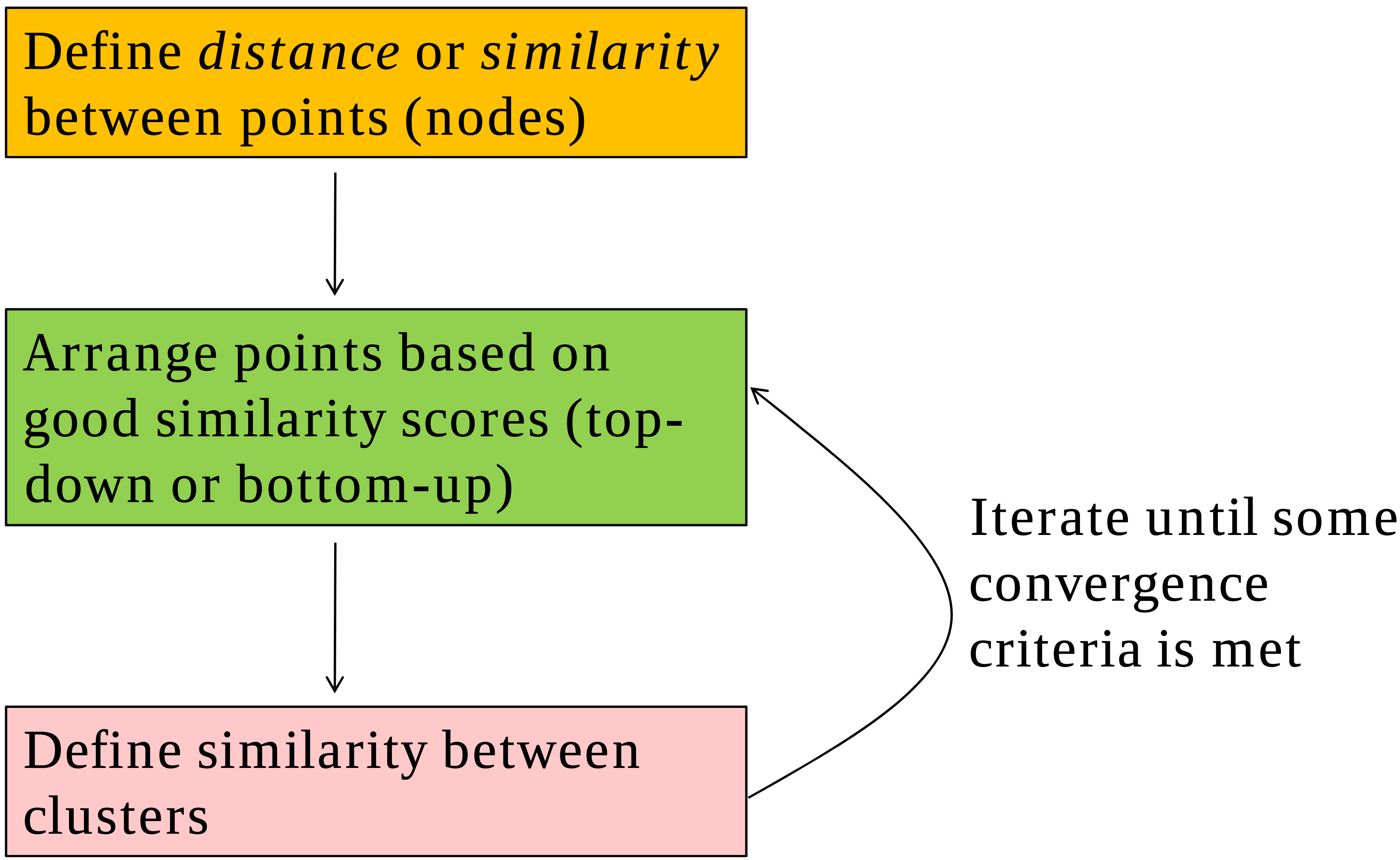
Clustering Techniques and Algorithms



Electrical centrality clustering

- We have worked on defining zones within the PJM system by clustering based on electrical distances
- Three clustering algorithms:
  - Hierarchical
  - Spectral (k-means)
  - Genetic Algorithms

Clustering algorithms



Fitness (I)

Four fitness metrics for cluster evaluation:

1. Clustering Tightness Index (CTI):

$$CTI = 1 - \frac{\sum_{a=1}^{n} \sum_{b \in M_a} d_{ab}}{\sum_{a=1}^{n} \sum_{\substack{a=1 \\ a \neq b}}^{n} d_{ab}}$$

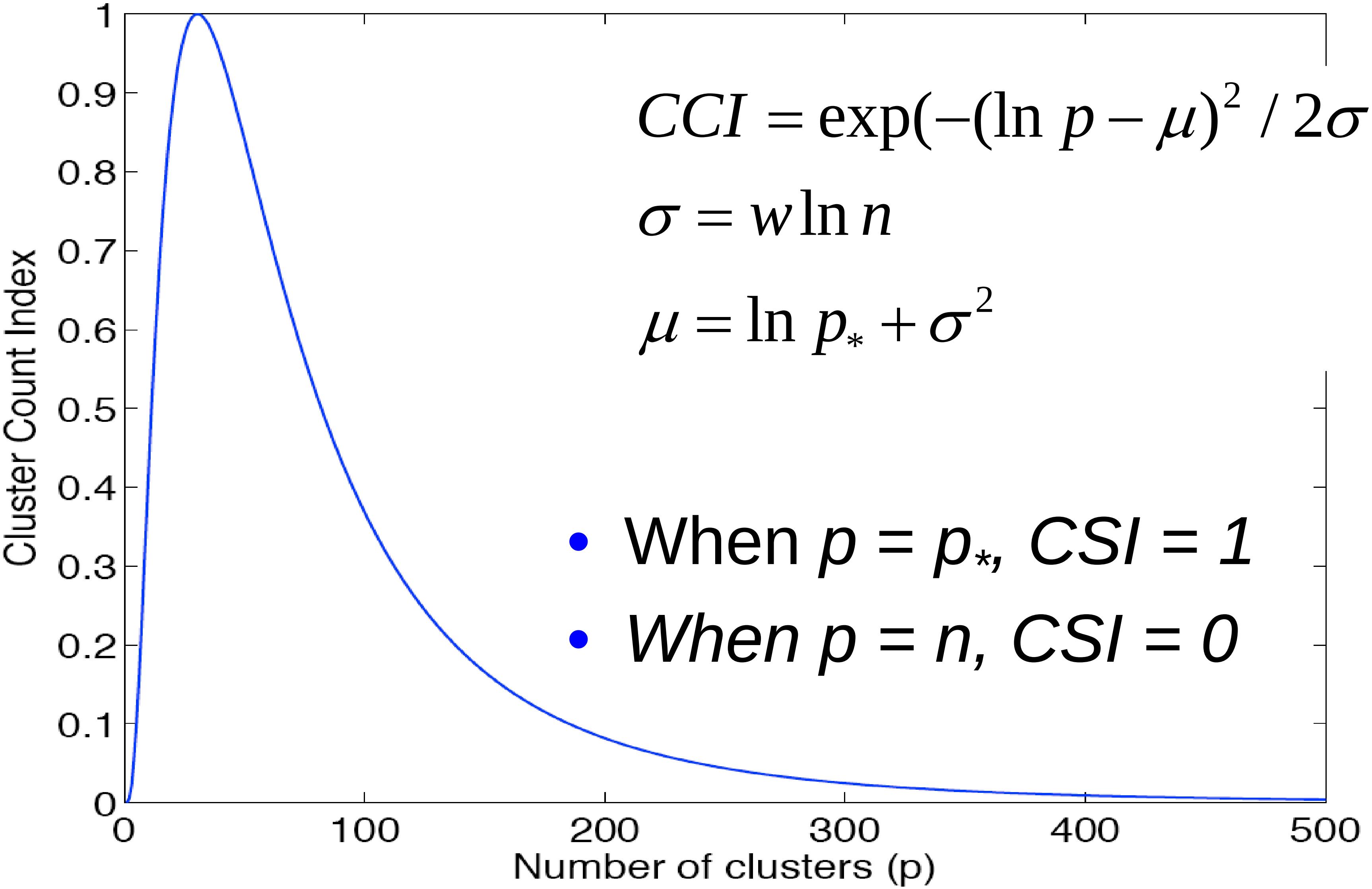
2. Cluster Count Index (CCI): p_* = desired # clusters

$$CCI = \exp(-(\ln p - \mu)^2 / 2\sigma^2)$$

$$\sigma = w \ln n$$

$$\mu = \ln p_* + \sigma^2$$

Clustering Count Index



$$CCI = \exp(-(\ln p - \mu)^2 / 2\sigma^2)$$

$$\sigma = w \ln n$$

$$\mu = \ln p_* + \sigma^2$$

- When $p = p_*$, *CSI = 1*
- *When* $p = n$, *CSI = 0*

Fitness (II)

3. Cluster Size Index (CSI): s = cluster size

$$CCI = \exp(-(\ln s - \mu)^2 / 2\sigma^2)$$

$$\sigma = w \ln n$$

$$\mu = \ln s_* + \sigma^2$$

$$s_* = n / p_*$$

4. Cluster Connectedness Test (CCT):

CCT = 1 if the cluster is connected

CCT = 0 if the cluster is unconnected

Aggregate fitness = CTI×CCI×CSI×CCT

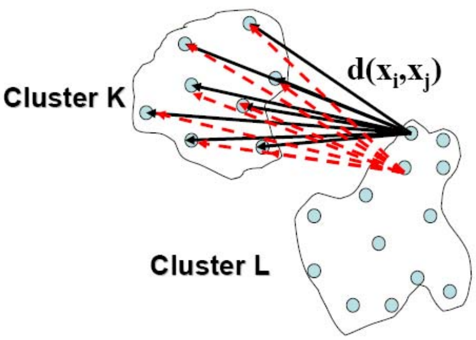
Clustering Techniques

- Hierarchical Clustering
- K-Means (Spectral) Clustering
- Genetic Algorithms

Hierarchical clustering

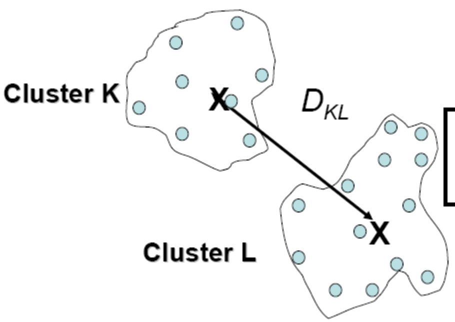
- Bottom-up clustering method

AVERAGE distance metric

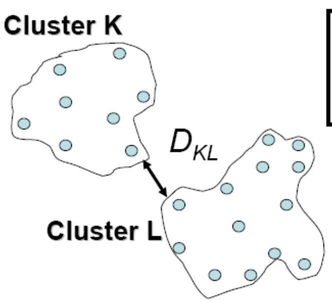


$$D_{KL} = \frac{1}{n_K n_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

CENTROID distance metric



$$D_{KL} = \|\bar{x}_K - \bar{x}_L\|^2$$



$$D_{KL} = \min_{\substack{i \in C_K \\ j \in C_L}} d(x_i, x_j)$$

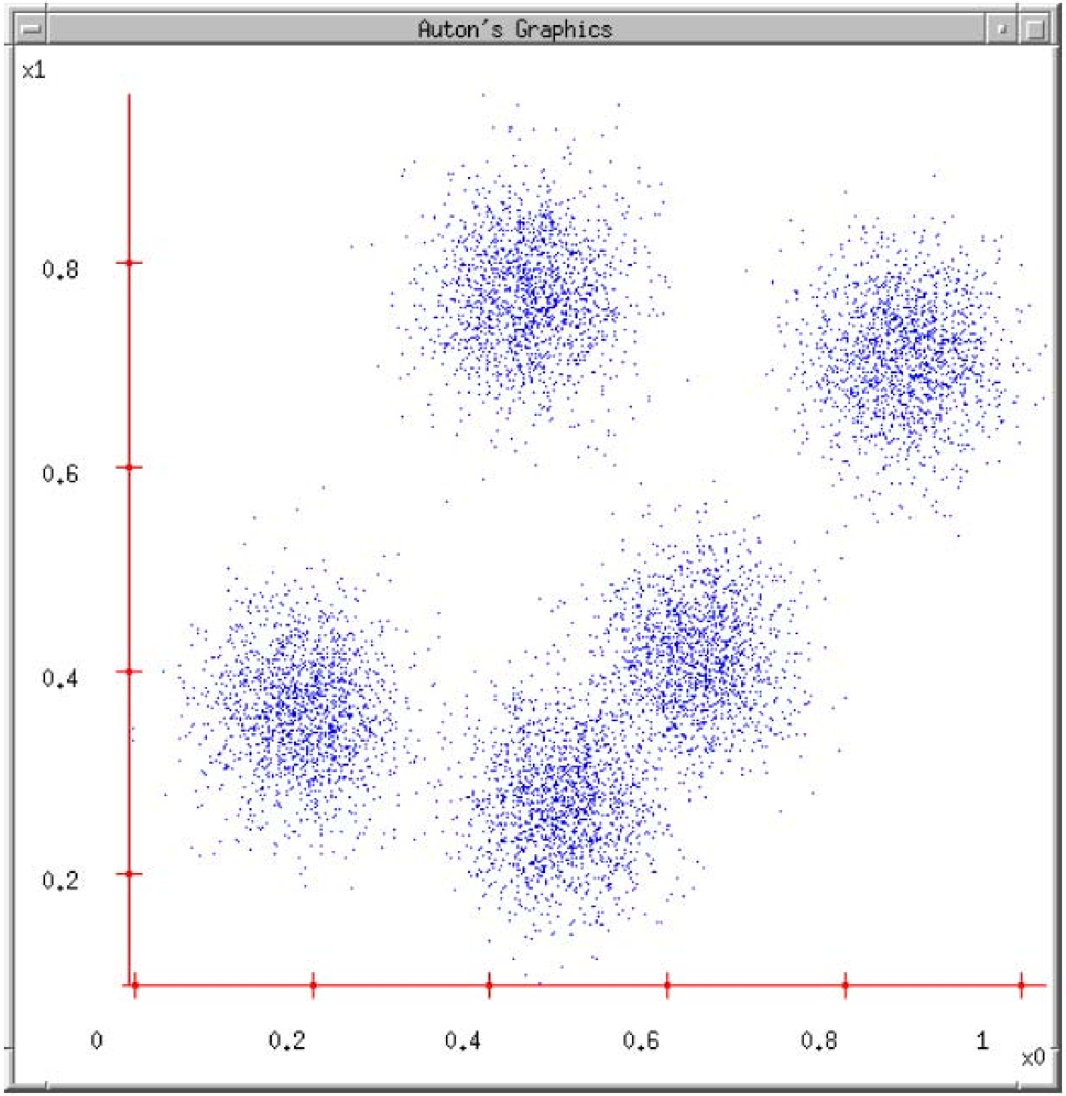
MINIMUM distance metric
(aka “single linkage”)

MAXIMUM distance metric
can also be used

K-means clustering

K-means

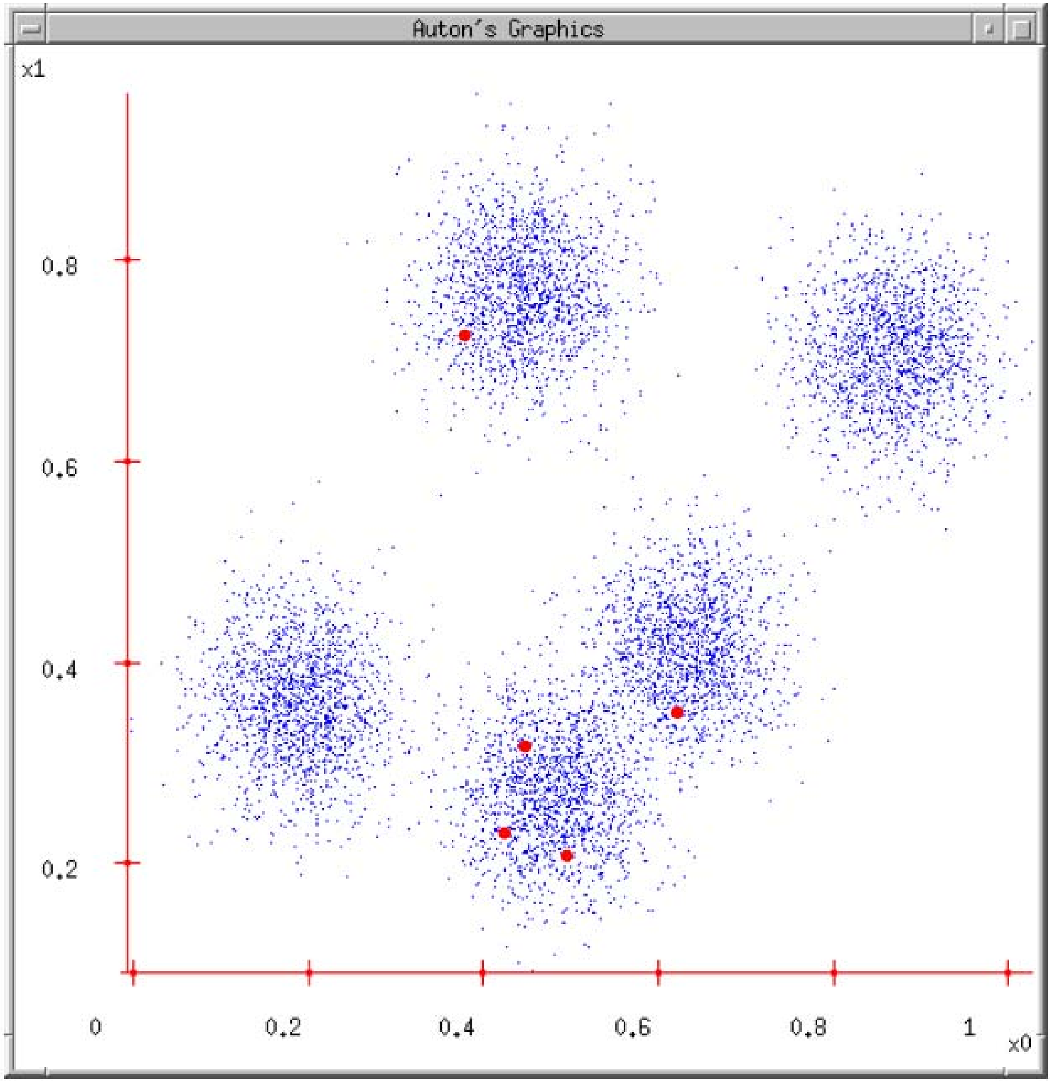
1. Ask user how many clusters they'd like. *(e.g. k=5)*



K-means clustering

K-means

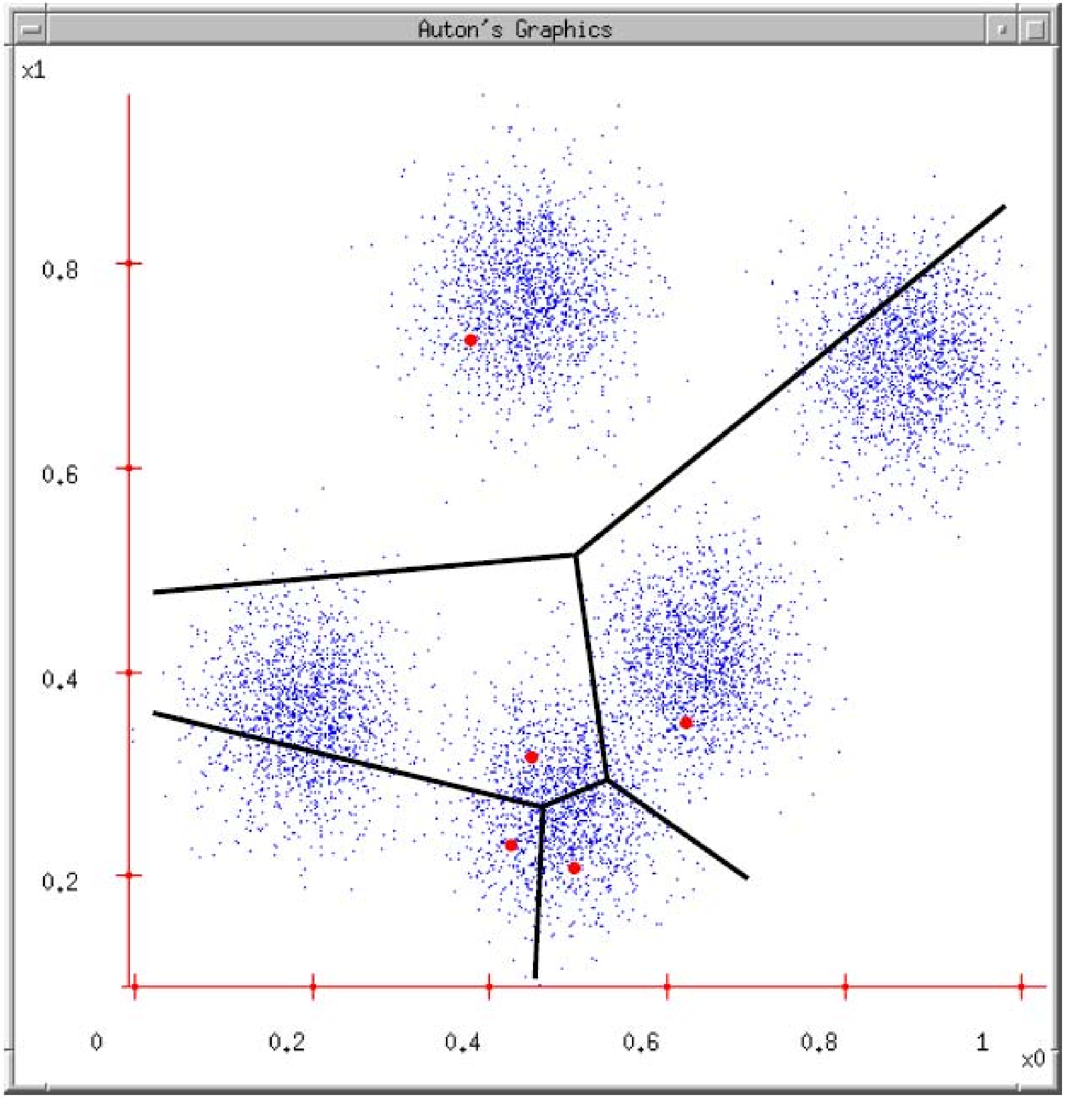
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



K-means clustering

K-means

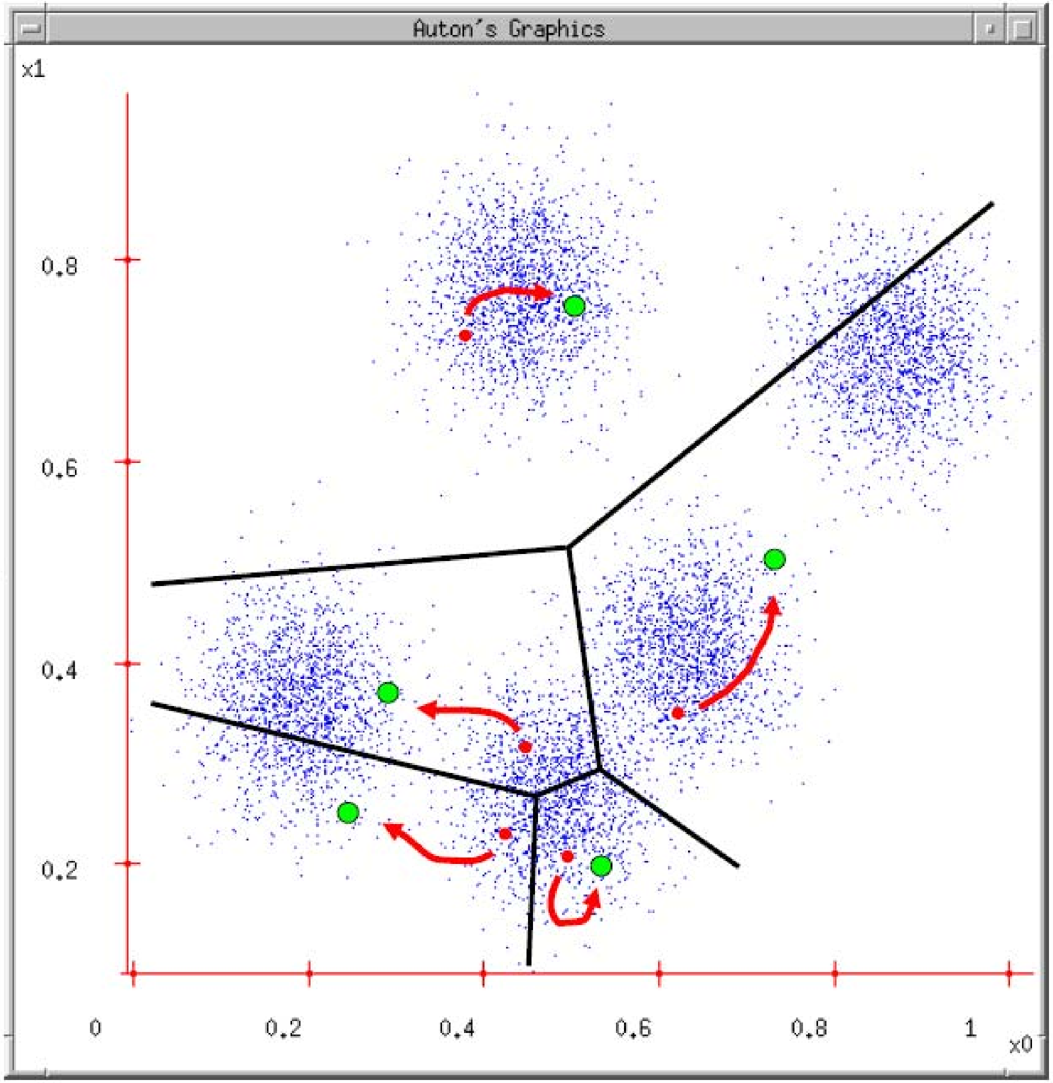
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



K-means clustering

K-means

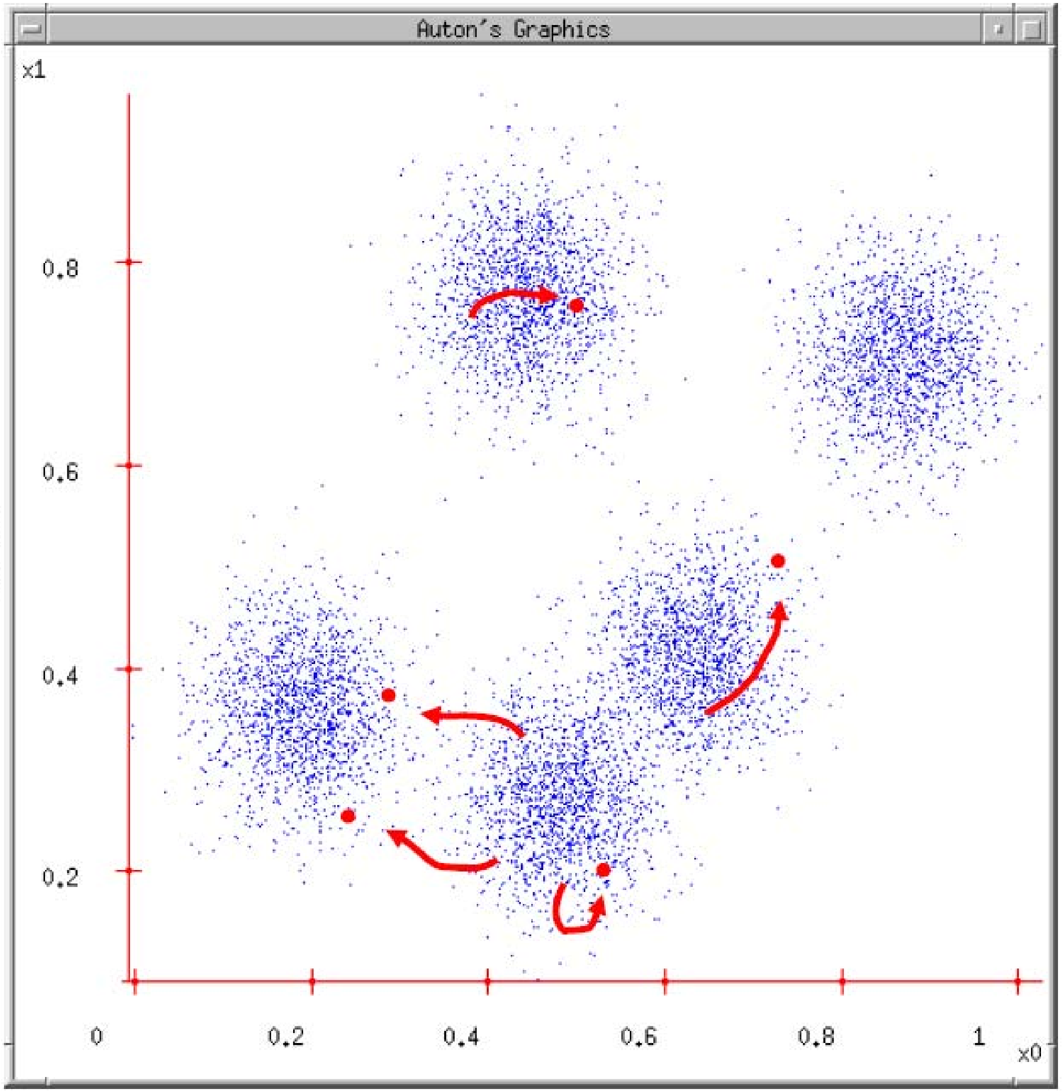
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means clustering

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!

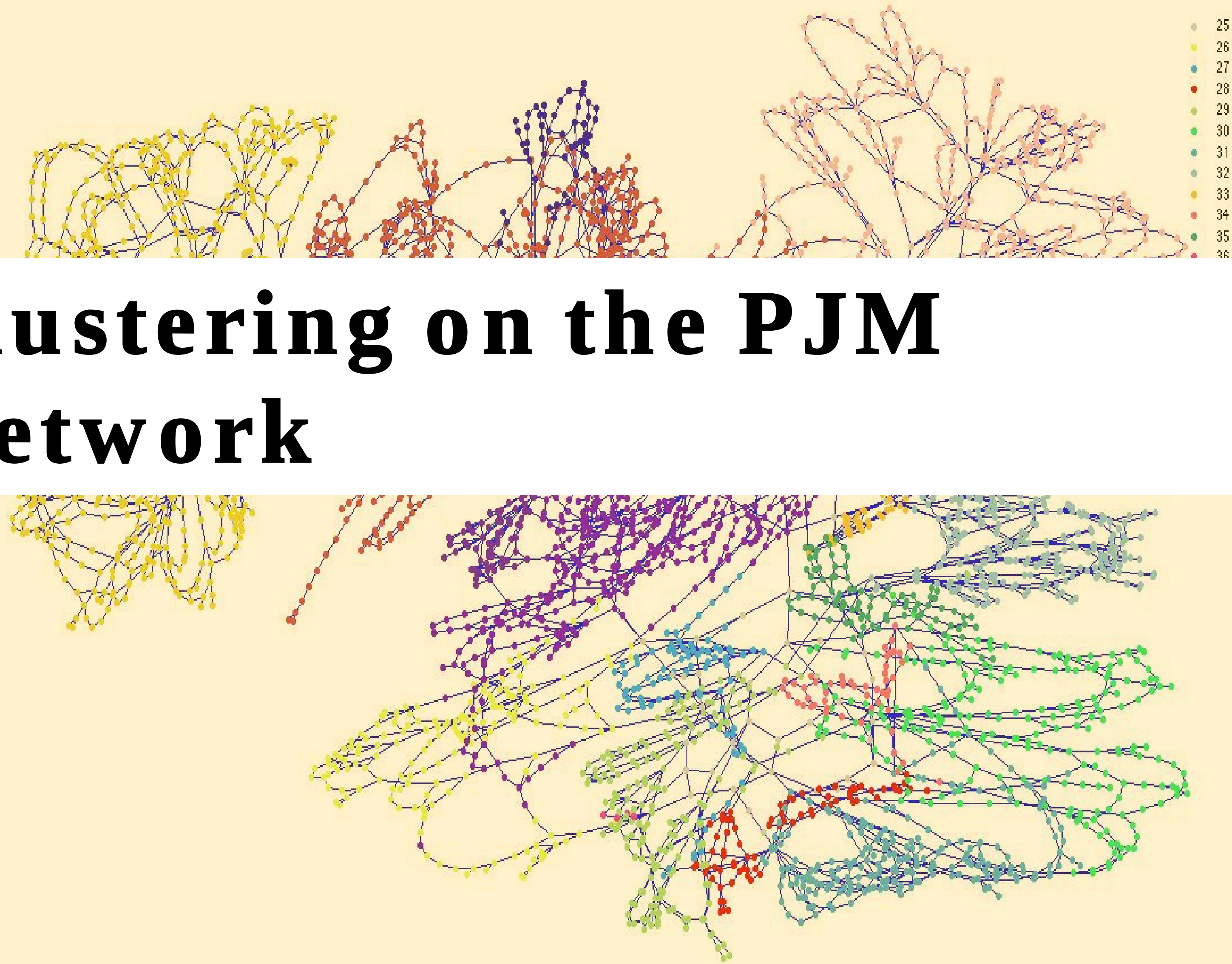


Spectral clustering

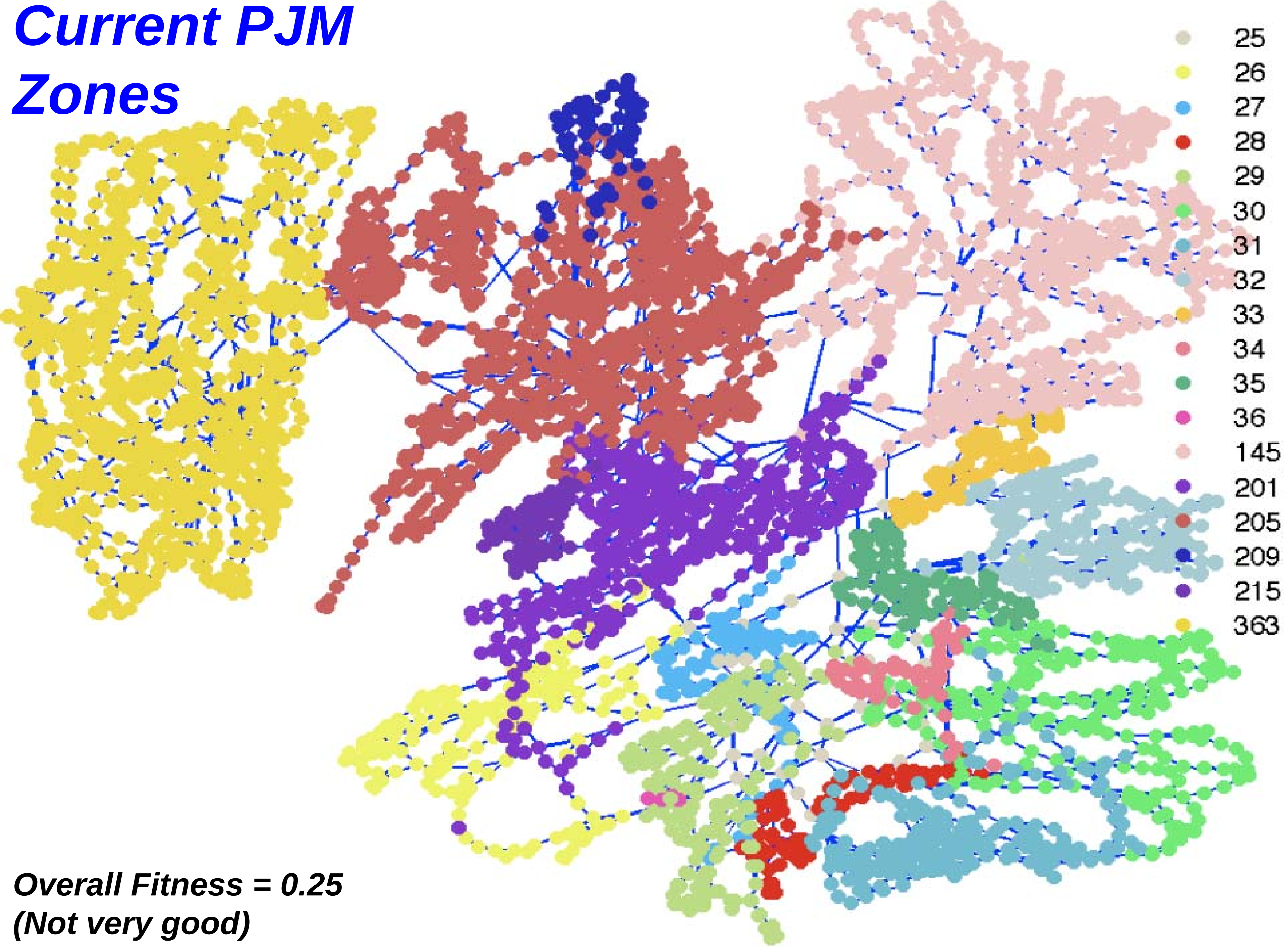
Performs k-means clustering on the spectral representation of the network

1. Calculate the graph Laplacian: $L = D - W$
2. Choose *k*, desired number of clusters
3. Calculate *k* eigenvectors associated with *k* largest eigenvalues
4. Perform k-means clustering on the eigenvectors.

Clustering on the PJM Network

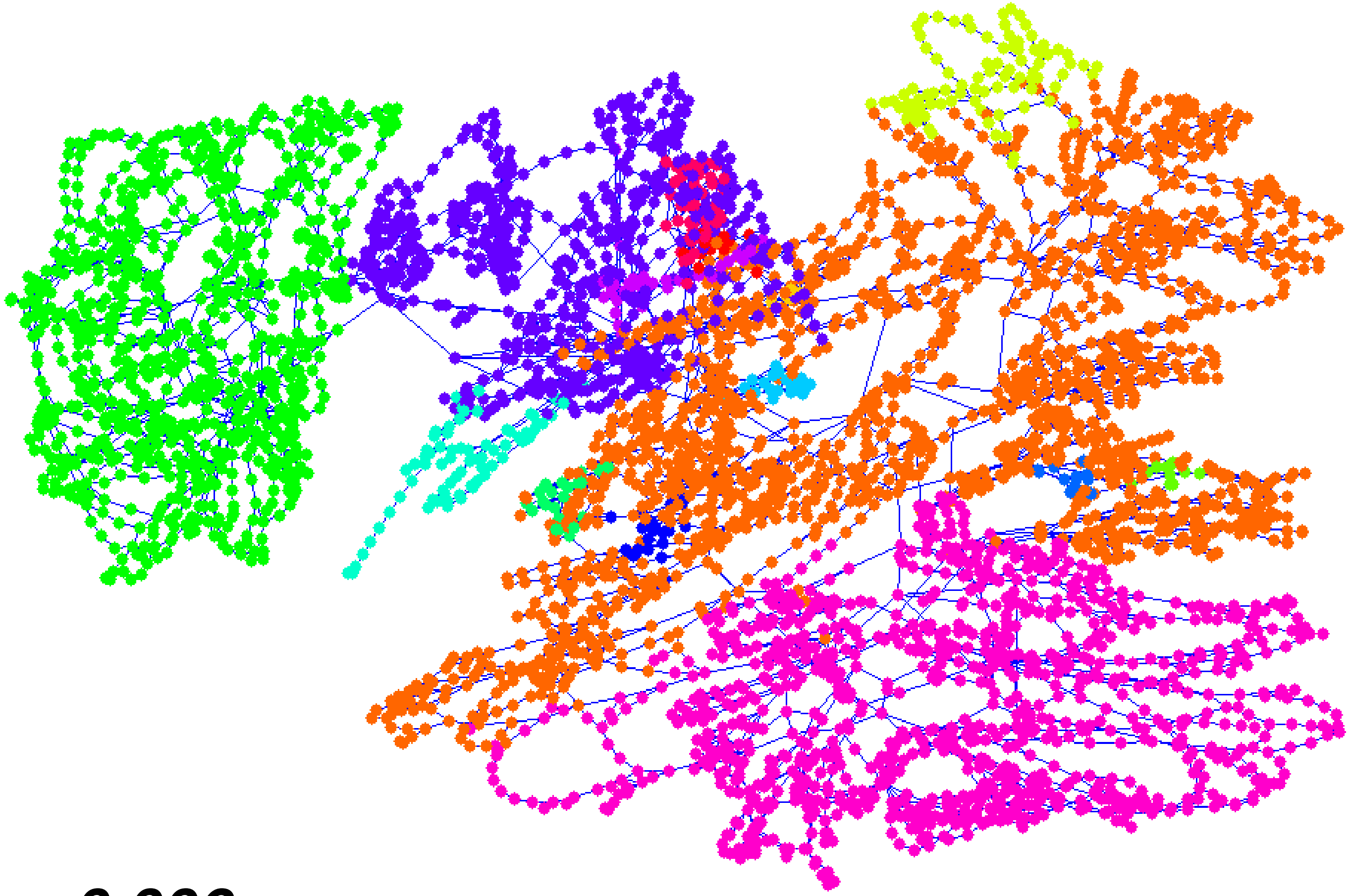


Current PJM Zones



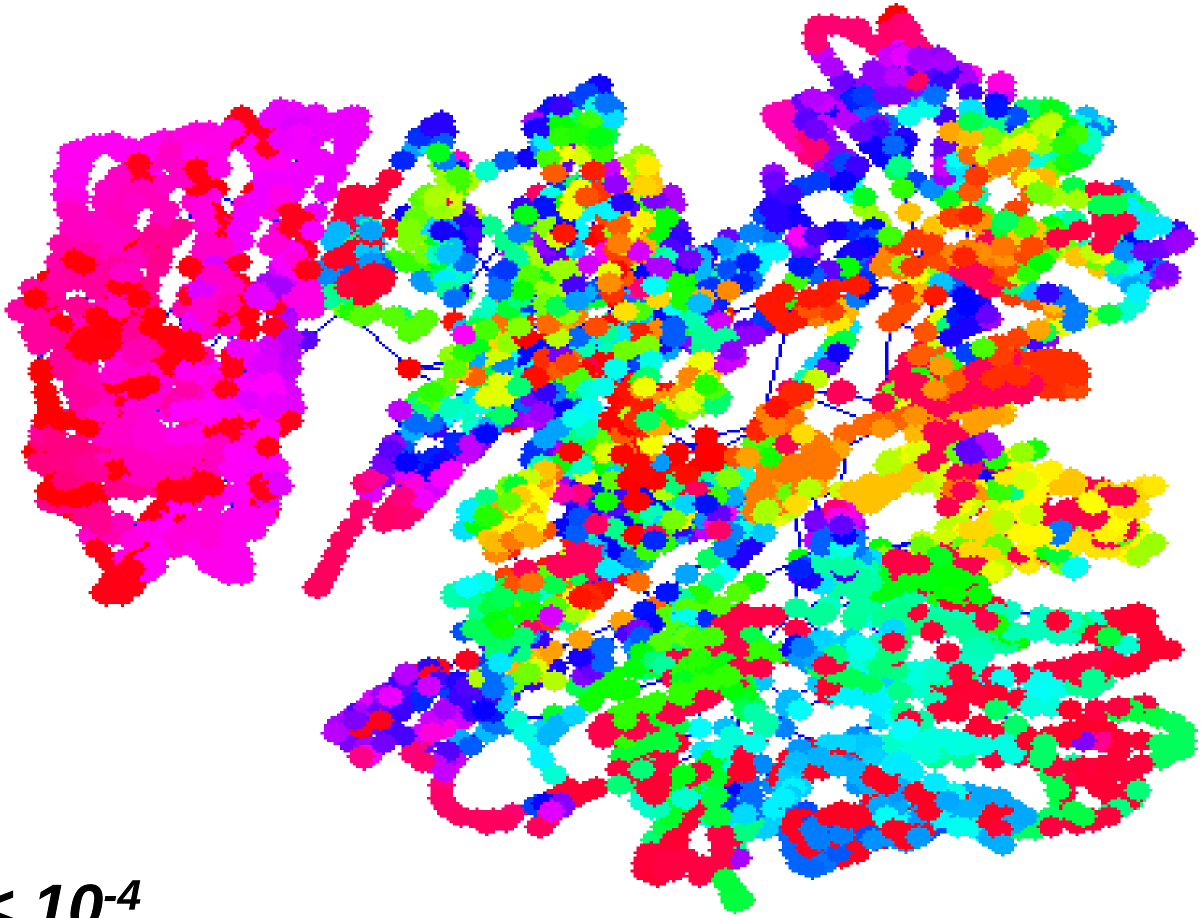
Overall Fitness = 0.25
(Not very good)

Clustering Method #1



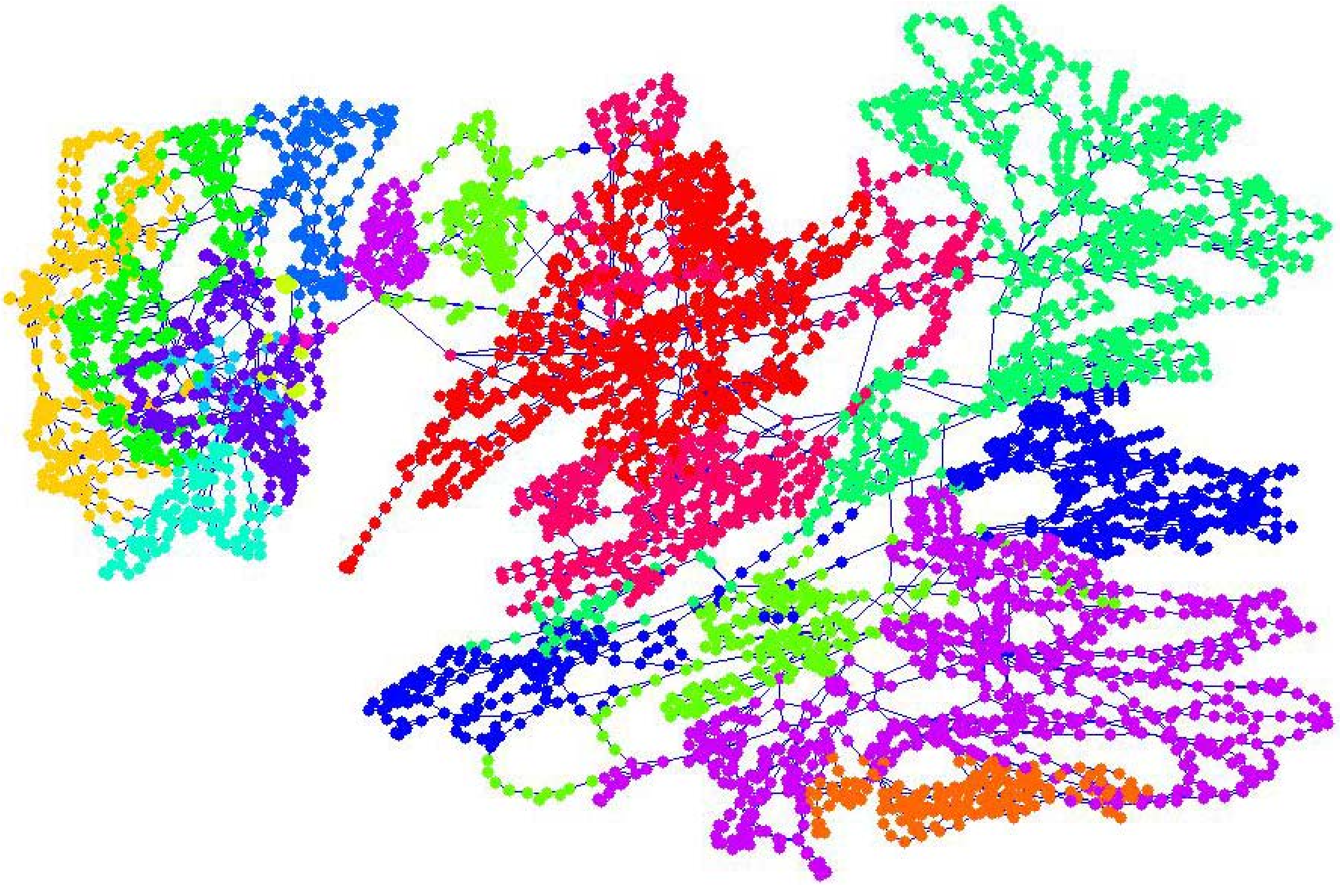
Fitness = 0.006

Clustering Method #2

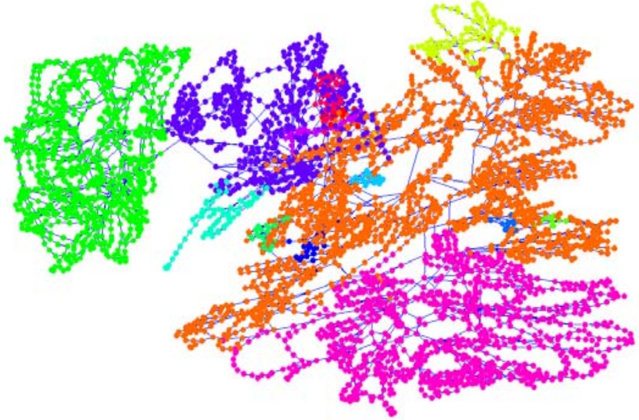


Fitness < 10^{-4}

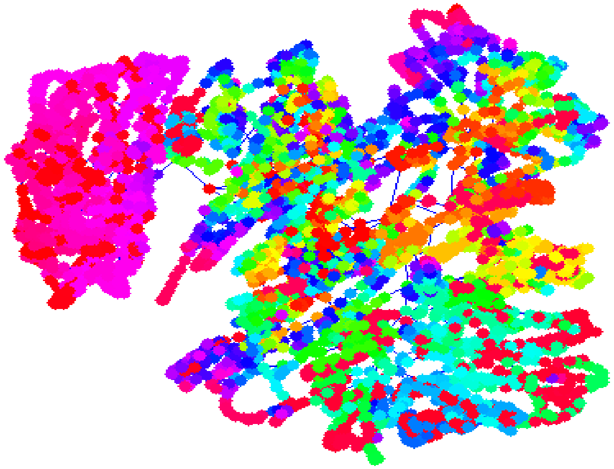
Clustering Method #3



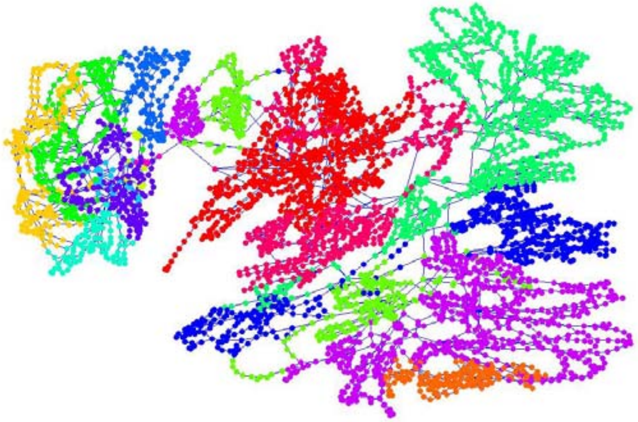
Fitness = 0.24



Hierarchical Clustering



K-Means Clustering



Spectral Clustering

Genetic algorithms

- Represent a solution (individual) as a string of bits
  - 01001011000110011010
  - Branch statuses
- Create an initial population
  - Random or from known solutions
- Evaluate fitness
- Do crossover, mutation, gen. eng, cloning
- Repeat until fitness reaches desired level

Crossover

- Choose two parents giving preference to the “most fit” individuals
- Choose a set of bits to select from each parent (adjacent strings or random)
- Swap bits:
  - 00111010101000010110
  - 01000000110111000110

    becomes
  - 00111010110111000110

Mutation

- For each bit flip a (weighted) coin.

- If it lands “heads” flop the bit
  - 0100101100011001101**0**
  - 0100101101011001101**1**

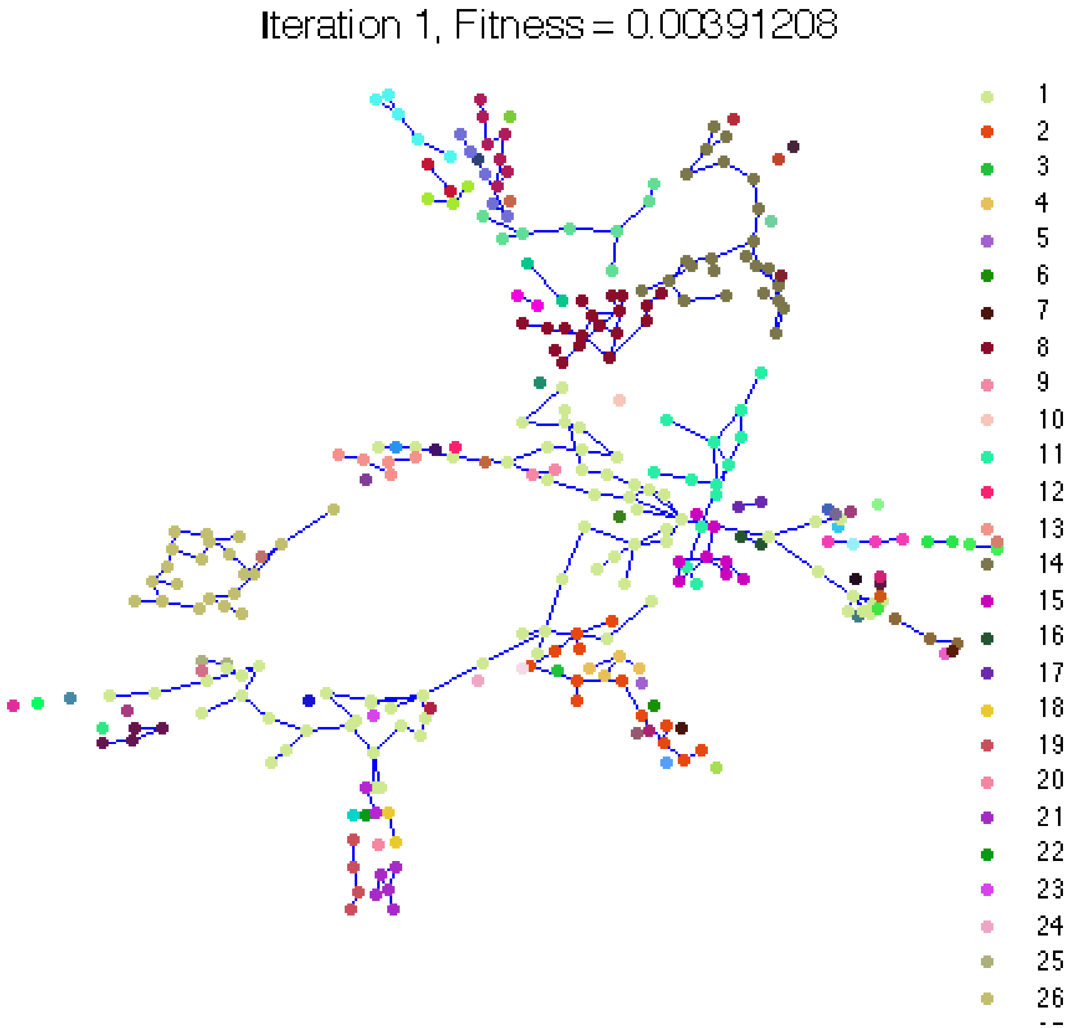
Genetic engineering

- Select an individual for GE, giving preference to more “fit” individuals
- Flip random bits to see which bit flip improves the fitness most
- Select the best modification
    - Tends to weed out “loners”

Cloning

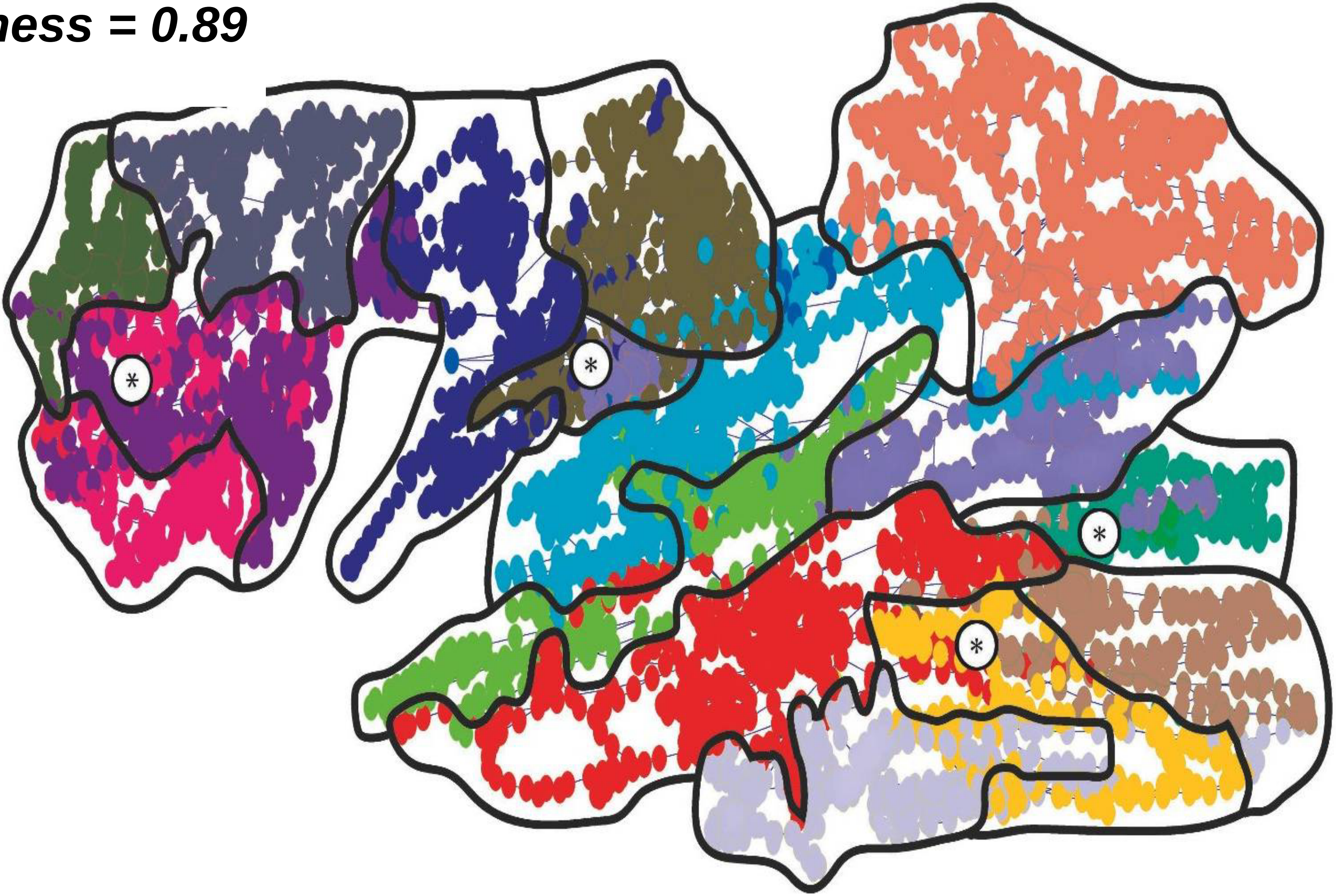
- Select several (3) of the best individuals in the previous population and copy them to the new population

Demonstration on the 300 bus network



Clustering via Genetic Algorithms

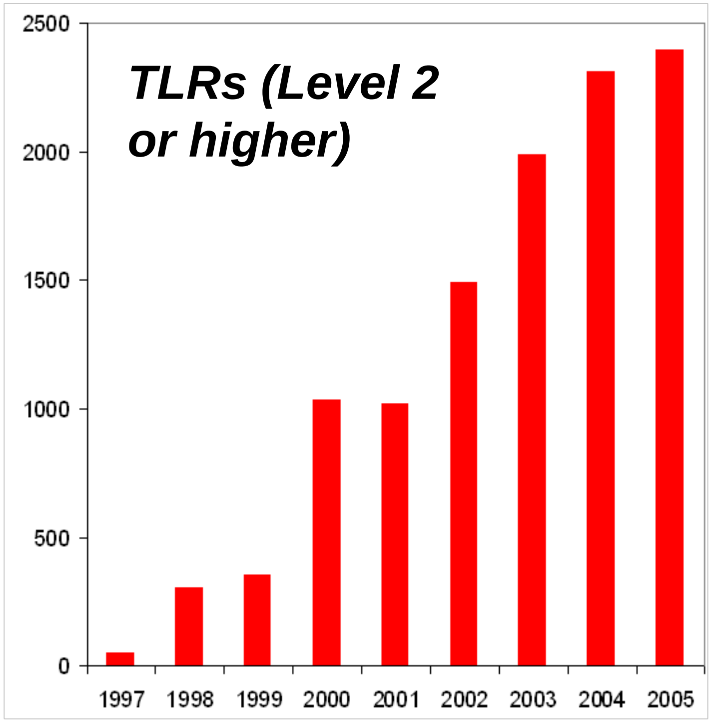
Fitness = 0.89

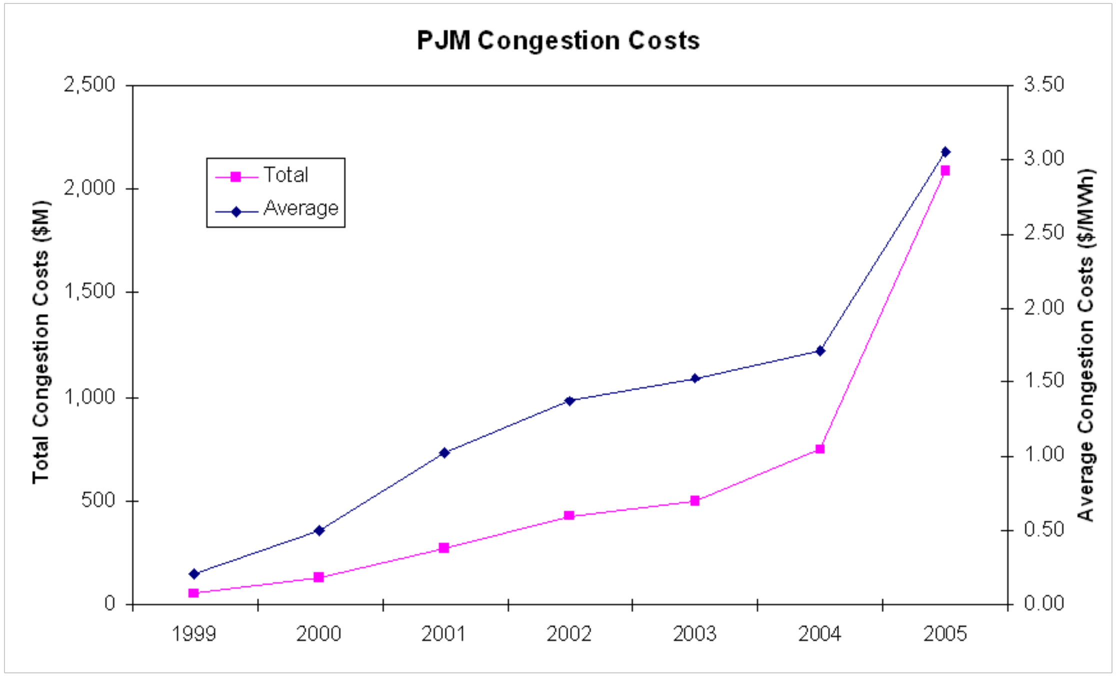


Flexible Transmission Topologies

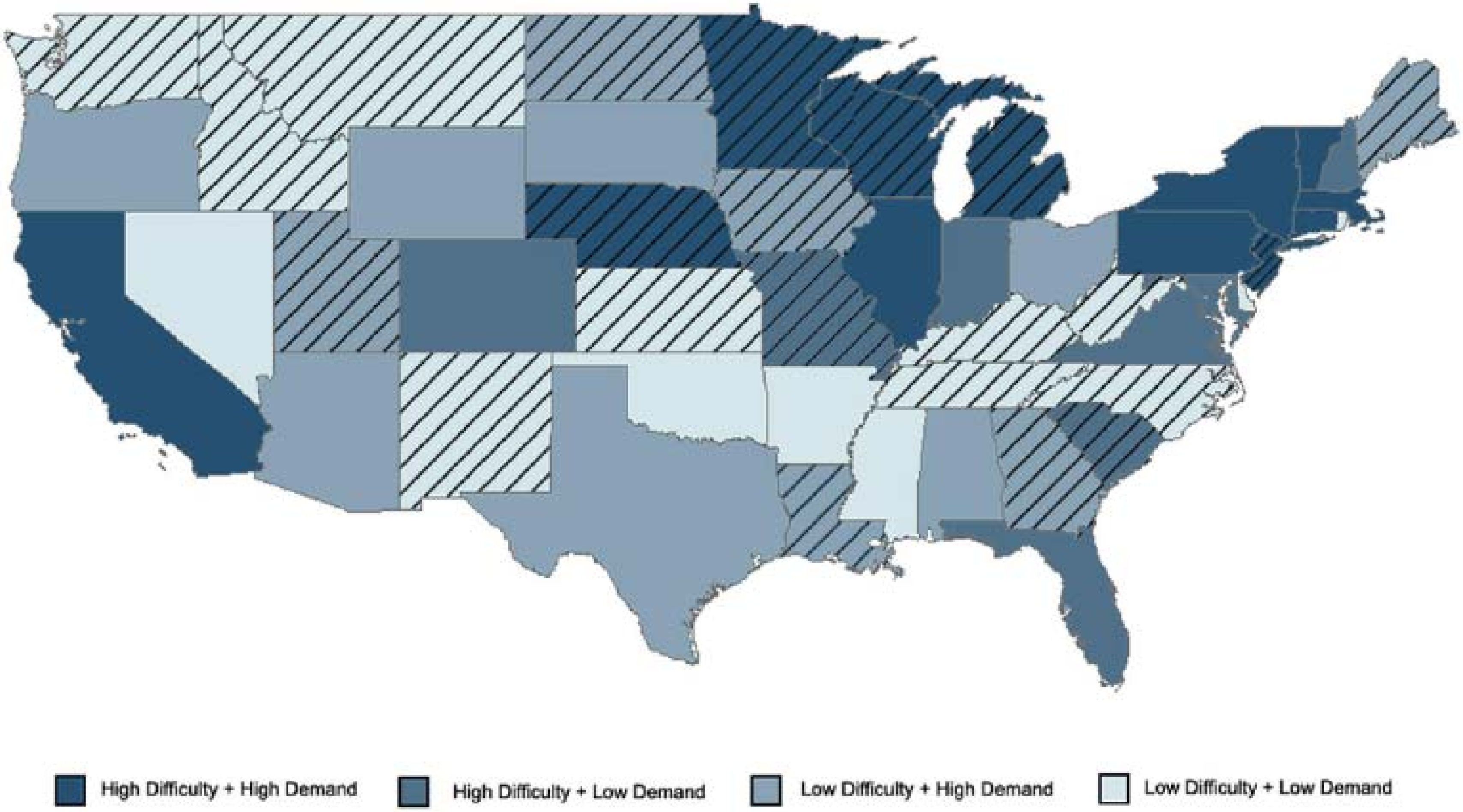
"Smart Grid" is not simply "smart meters"

The Transmission Problem (I)

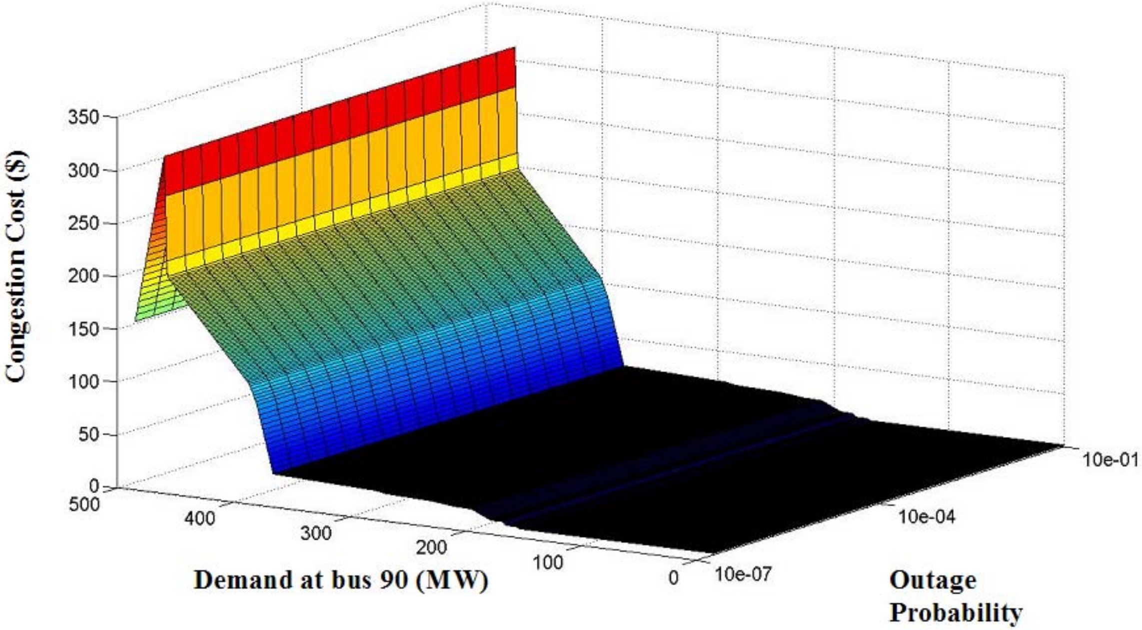


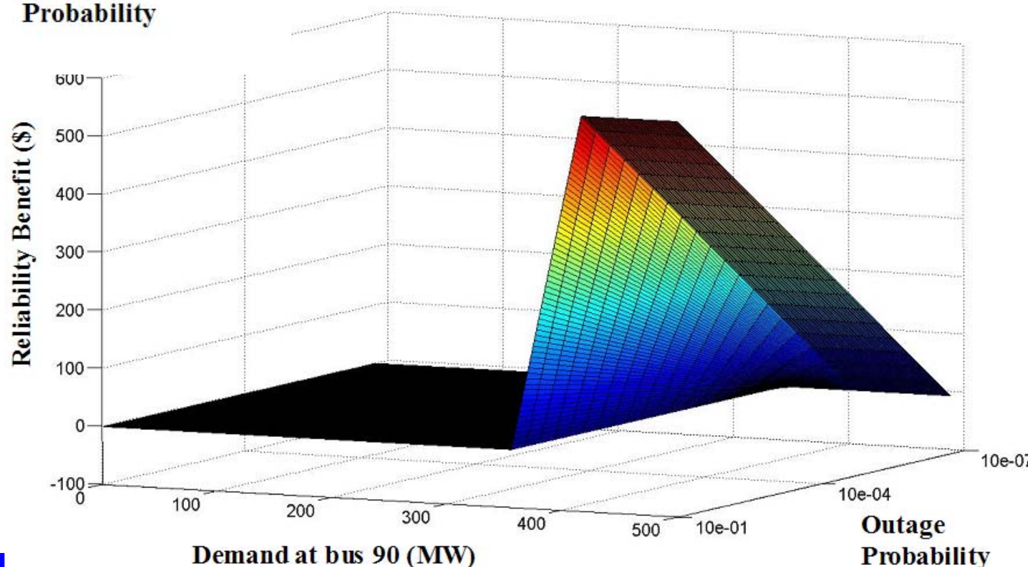


The Transmission Problem (II)



Congestion and Reliability





Flexible Transmission Topologies

- New transmission is needed to achieve many different energy goals (renewables, reliability, competition)
- Another application of “smart grid” technology is to allow the system operator to adjust the grid topology in near-real time
- FACTS (power electronics), fast switching, are a couple of examples

Problem Formulation

- Formulate an optimization problem where the status of network branches is a *decision variable* for the system operator

$$\min_{P_{Li}, P_{Gi}, u_{ij}} \sum_i \left(C_i(P_{Gi}) + VOLL_i \times UE_i \right)$$

$$s.t.$$

$$\sum_j F_{ij}^{transfer} = P_{Gi} - P_{Li} \qquad \forall i$$

$$F_{ij} = u_{ij} f(V_i, V_j, \theta_i, \theta_j)$$

$$\mathbf{g}(\mathbf{x}) = \mathbf{0}$$

$$\mathbf{h}(\mathbf{x}) \leq \mathbf{0}$$

Placement Strategies?

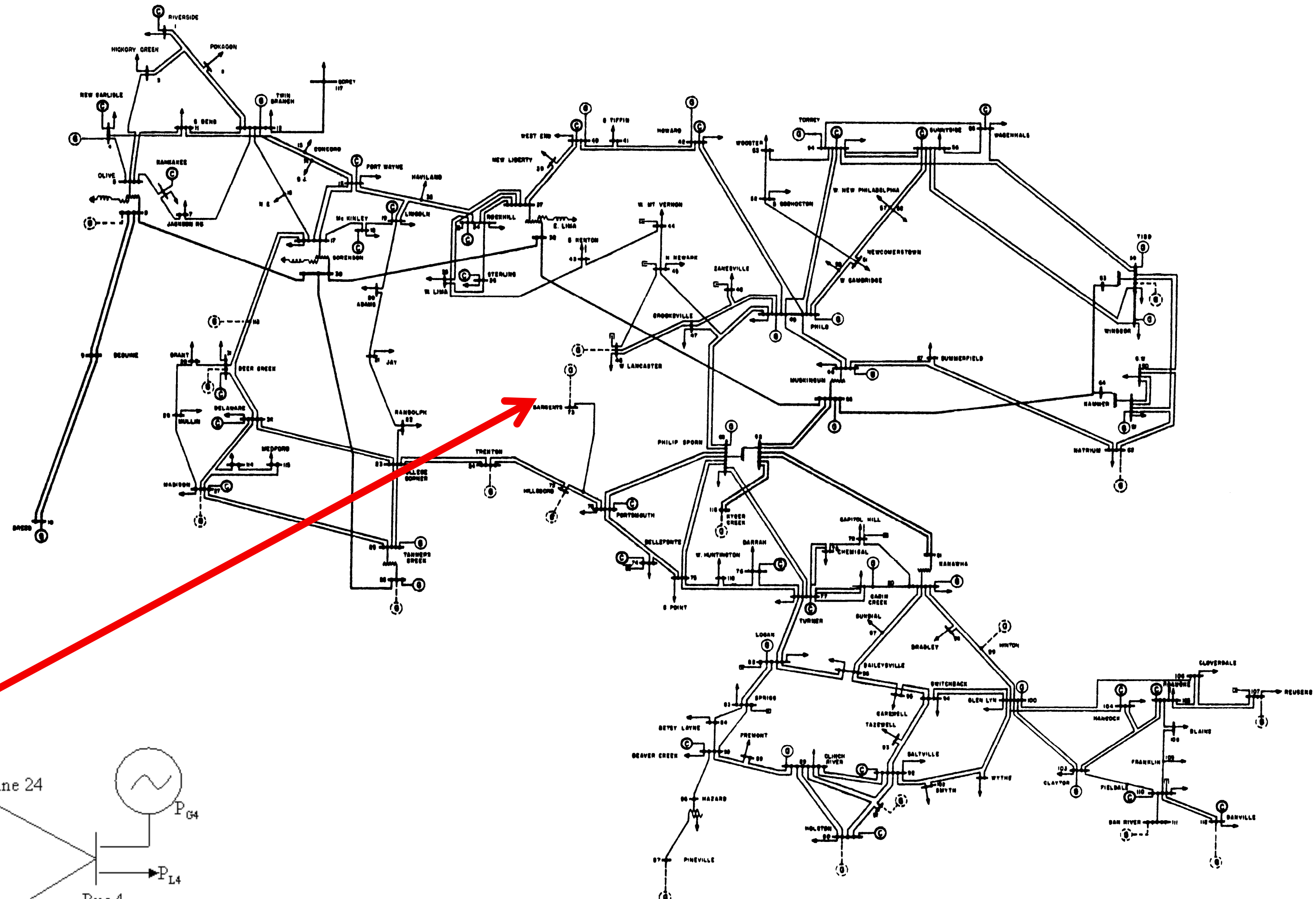
- The mixed-integer programming problem is essentially impossible to solve globally.
- Do we really need *all* transmission to be controllable by the system operator?
- Are there *diminishing returns* to controllability? (Can we get 90% of the benefit by strategically selecting and controlling 15% of the lines?)

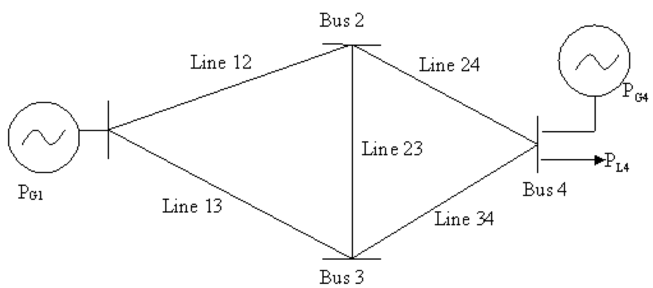
Graph-theoretic Strategies

Suppose you are given *N* controllers, where *N* << # lines in the system

- *Strategy #1*: Distribute the controllers randomly (this doesn't seem to work well)
- *Strategy #2*: Distribute the controllers in locations topologically similar to Wheatstone bridges
- *Strategy #3*: Distribute the controllers in areas with high electrical centrality

Finding Wheatstone Bridges





Thank You!

Questions?

Seth Blumsack
Dept. of Energy and Mineral Engineering
Penn State University
blumsack@psu.edu

Thanks to NSF and PJM for funding support