

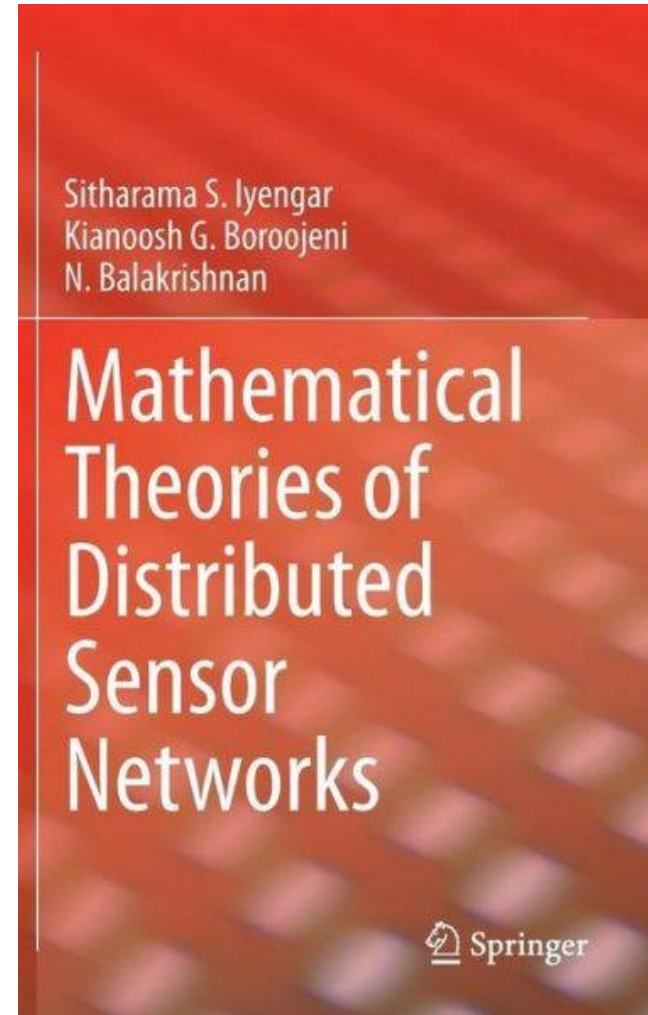
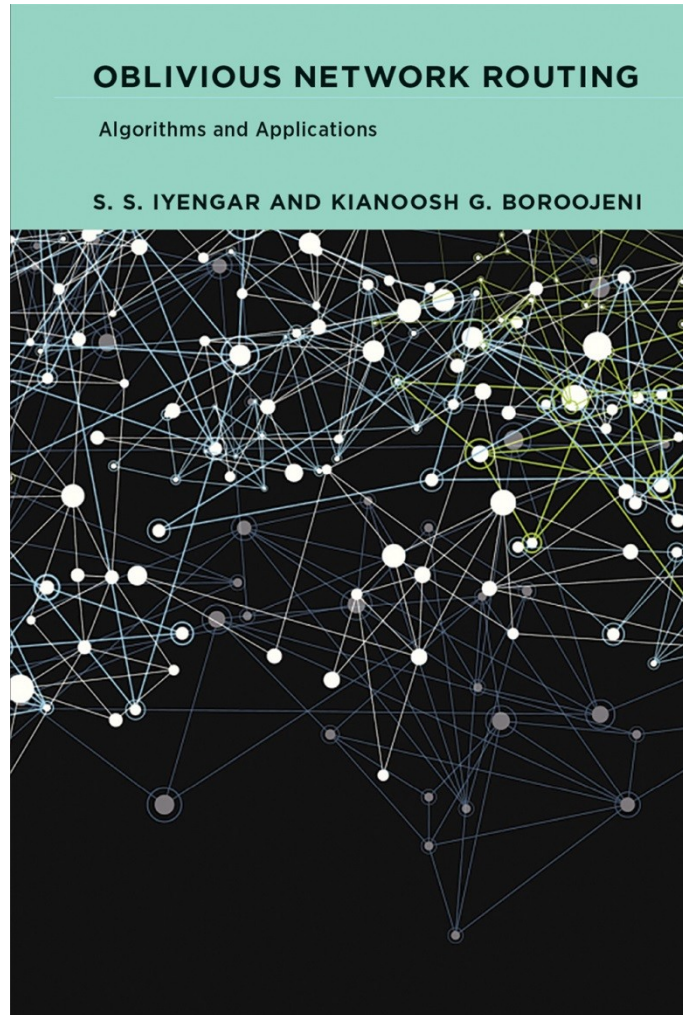
Oblivious Network Routing: Algorithms and Application

S.S. Iyengar, ACM Fellow, IEEE Fellow, AAAS Fellow, National Academy of Inventors Fellow of USA, Director and Ryder professor, and **Kianoosh G. Boroojeni**, PhD student, School of Computing and Information Sciences, FIU Miami, Florida.

Acknowledgements: Thanks to NSF CNS1158701, Srivatsan Srinivasagopalan and Konstantin Busch for their contribution
Some of their slides are incorporated here

March 10th 2016, Talk at IIT Chicago, IL

Books regarding Oblivious Network Routing

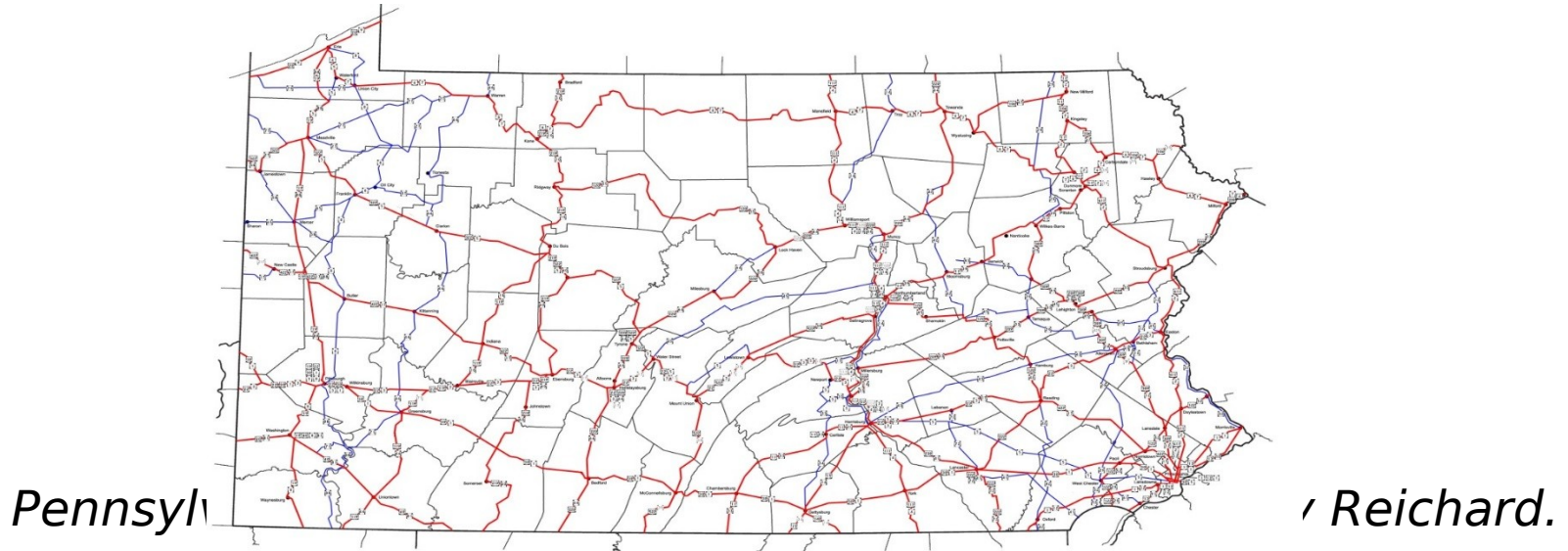


Agenda

- Introduction
 - Oblivious vs. Classic Network Routing
 - Example and illustration
- Motivation and Problem Set-Up
- Related work and our contribution
- Proof Sketches
- Other Applications
- Future Work and wrap up

Introduction

- **Network** is a common concept that is used to describe a group of interconnected things.



Network and Routing Problem

- Network is a *tool for **flowing** something from one place to another via the network connections in a step by step manner.*
- Each network flow will have the following properties: *source point, target point, flow path, flow amount, and flow cost.*
- The problem of *specifying flow paths for given source/target points* is called **network routing problem**.

Classic Network Routing Cost Optimization

- Problem of specifying flow paths through a network such that the cost incurred by the flow in specified paths gets its minimum value.
- The optimization problem is subject to:
 - Flow sources/targets
 - Flow amount
 - Cost function in each network connection (link)

Classic Network Routing: Example

- Users/clients.
- Each user has a **demand** – number of packets/bits.
- **Cost** of sending information on an edge for a set of users depends on a single parameter – the **total demand** of that set of users.

e.g. **Steiner tree**: $\text{coste}(S) = c_e$ for $S \neq \emptyset$

Buy-at-bulk ND: $\text{coste}(S) = c_e \cdot f(|S|)$, f concave

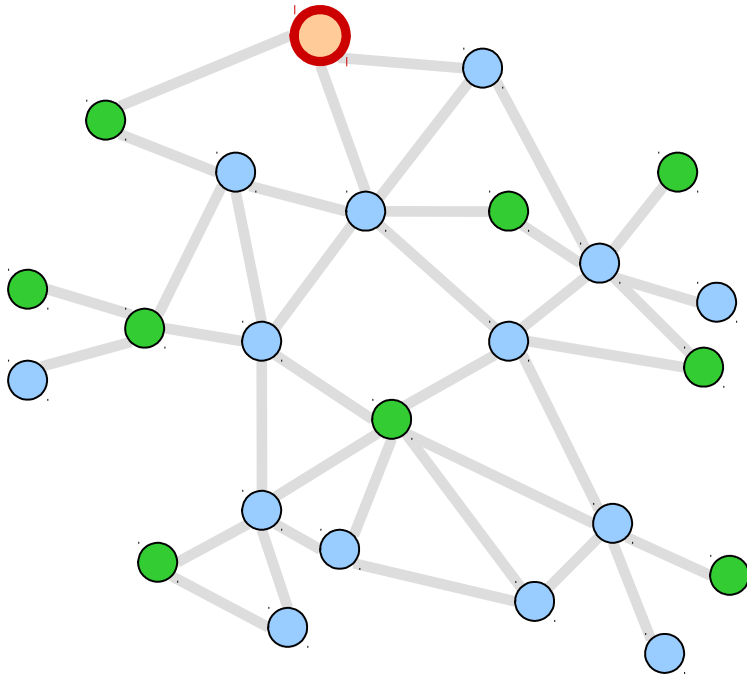
Classic Routing Cost Optimization Problem

- Single-sink problem

● : Terminal

○ : Sink

● : Node



Task: For each terminal, choose a path to sink to send information.

Goal: Minimize total cost of sending information

$$= \sum_e c_e \cdot f(A_e)$$

Edge
weight
t

fusion
function

Set of
terminals
using edge e

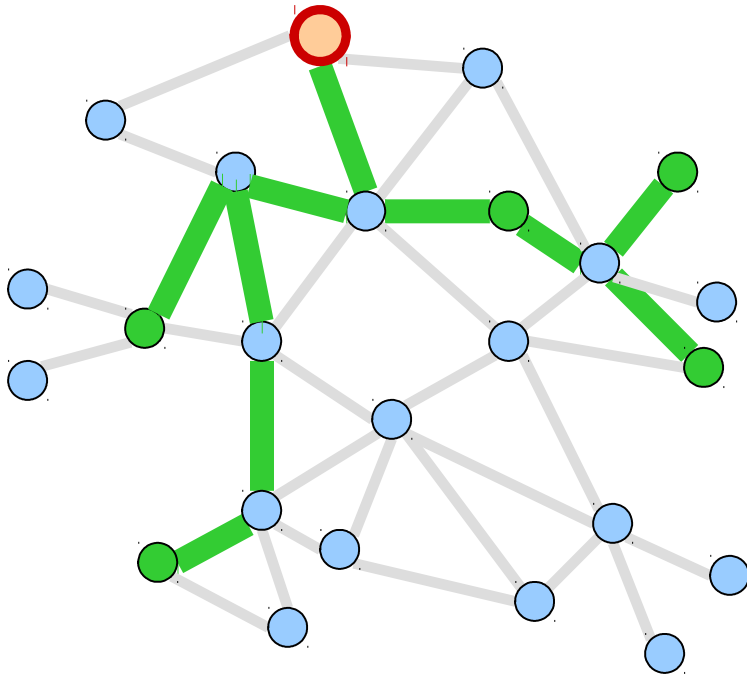
Cont.

- Single-sink problem

● : Terminal

○ : Sink

● : Node



Task: For each terminal, choose a path to sink to send information.

Goal: Minimize total cost of sending information

$$= \sum_e c_e \cdot f(A_e)$$

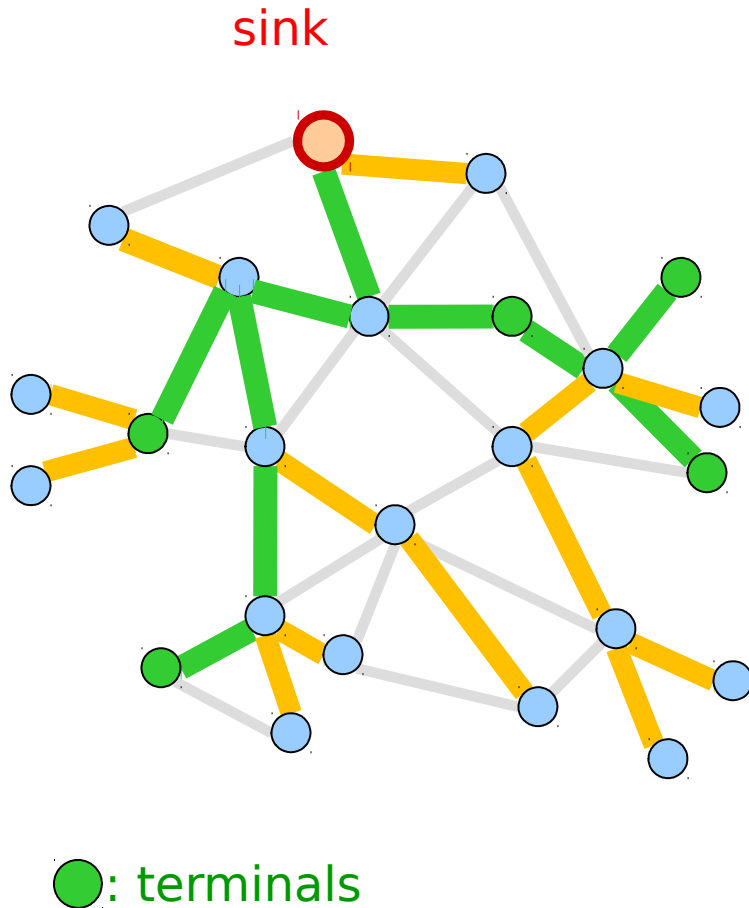
Oblivious Routing Optimization Problem

- The mentioned cost optimization problem is **oblivious** if the flow source/target/amount or cost function *are not restricted* to known values.
- Example:
 - Considering Internet at any moment, large amount of data is being exchanged between computer devices scattered all around the world.
 - The traffic flow amount in each point located in the Internet is permanently changing which eventuates in time-varying cost amount incurred by the data flows passing through it.

Main Properties of Oblivious Problems

- The low-cost flow paths through the networks are too time-consuming to be computed online (upon request).
- This is basically because of the large size of the network and/or the necessity to route very quickly.
- The practical way of flow routing in such networks is to route in an *offline manner*.
- *Pre-computing* the flow paths.

Oblivious Network Routing Optimization



Task: pre-compute paths to sink.

Goal: Minimize

$$\sum_{e \in E} w_e \cdot f(A_e)$$

For every set of terminals and for every
Canonical fusion cost function f .

Applicability

- **Sensor Networks**
 - Distributed sensor nodes send information to central node(s)
 - Information can often be well aggregated along paths, e.g., temperature readings
 - May care only about aggregate information, e.g., average temperature, humidity ...
- **Content-based publish-subscribe systems**
 - Users “publish” or “subscribe” to information
 - Information flowing through network can be aggregated
 - Idea: publisher produces web pages, **content distribution network** replicates web pages to many locations so consumers can access at higher speed
 - **MST may not be good enough!**
 - **content distribution on minimum cost tree may take a long time!**

Applicability Cont...

- **Power Grids**
 - Power distribution in grids
 - Cost of power splitting is expensive (low degree spanning trees needed)
 - Overlay Trees – A solution for fault-tolerant / survivable grids.
- **VLSI Power Circuitry**
 - Minimizes IR-Drop (reduces heat dissipation).
 - Dynamic Voltage support – Near Optimal
 - “**Class Steiner Tree Problem**” - given a connected graph with required classes of terminals, find a shortest connected subgraph that contains at least one terminal from each class.

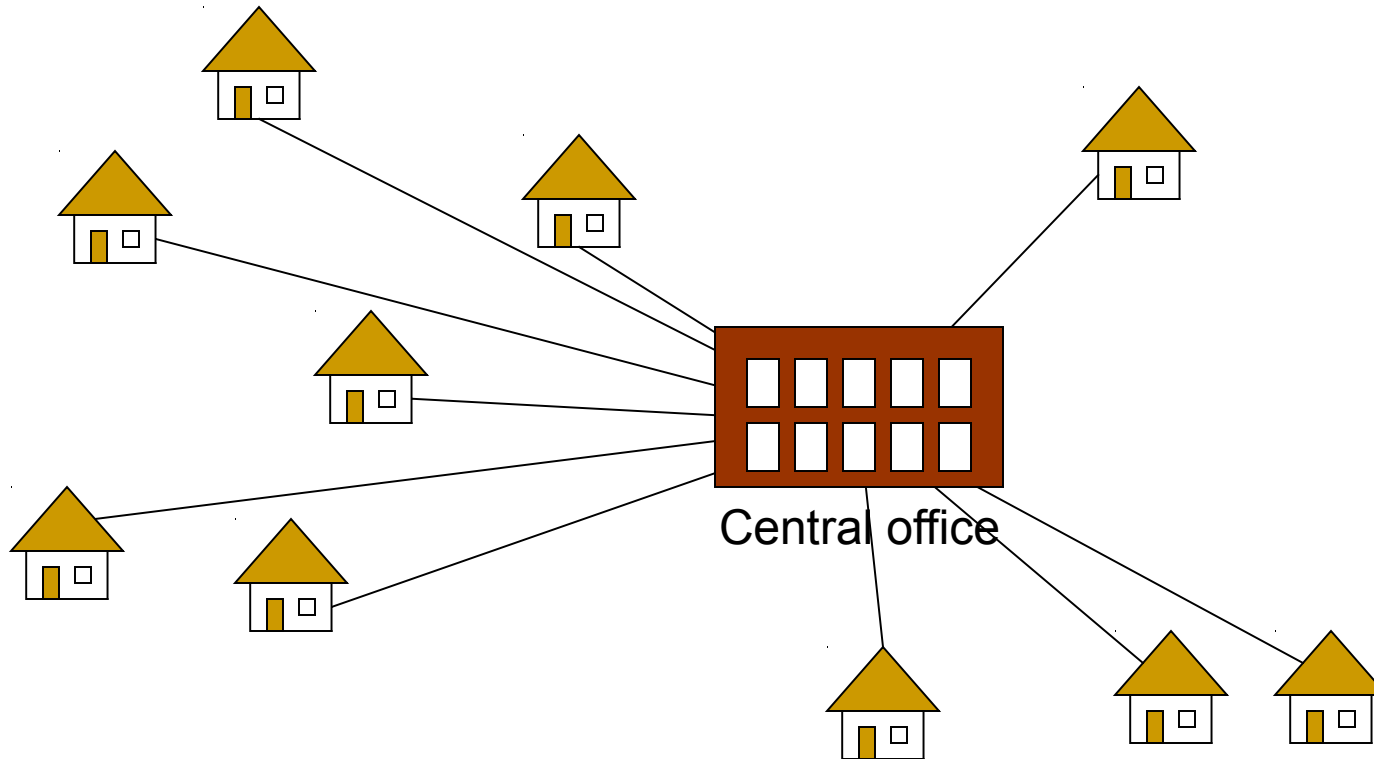
Applicability Cont...

- Pipelines (Oil, Gas, Water etc)
 - A natural application of concave cost functions
 - Cost Effective = notion of “Buy-at-Bulk”
 - Ideal Solution: Min-Cost Steiner Tree
- Railroads / Highways / Telephone Network
 - Optimal deployment of railroads – oblivious to the size/population/demands of the cities it connects to.
 - Will hold traffic when there are changes to demands.
 - *“It is to construct roads of minimum total length to interconnect several highways under the constraint that the roads can intersect each highway only at one point in a designated interval which is a line-segment.”* – [No bounded polynomial-time approximation has been found for this problem] – Courtesy “Steiner Trees in Industries”

Publications

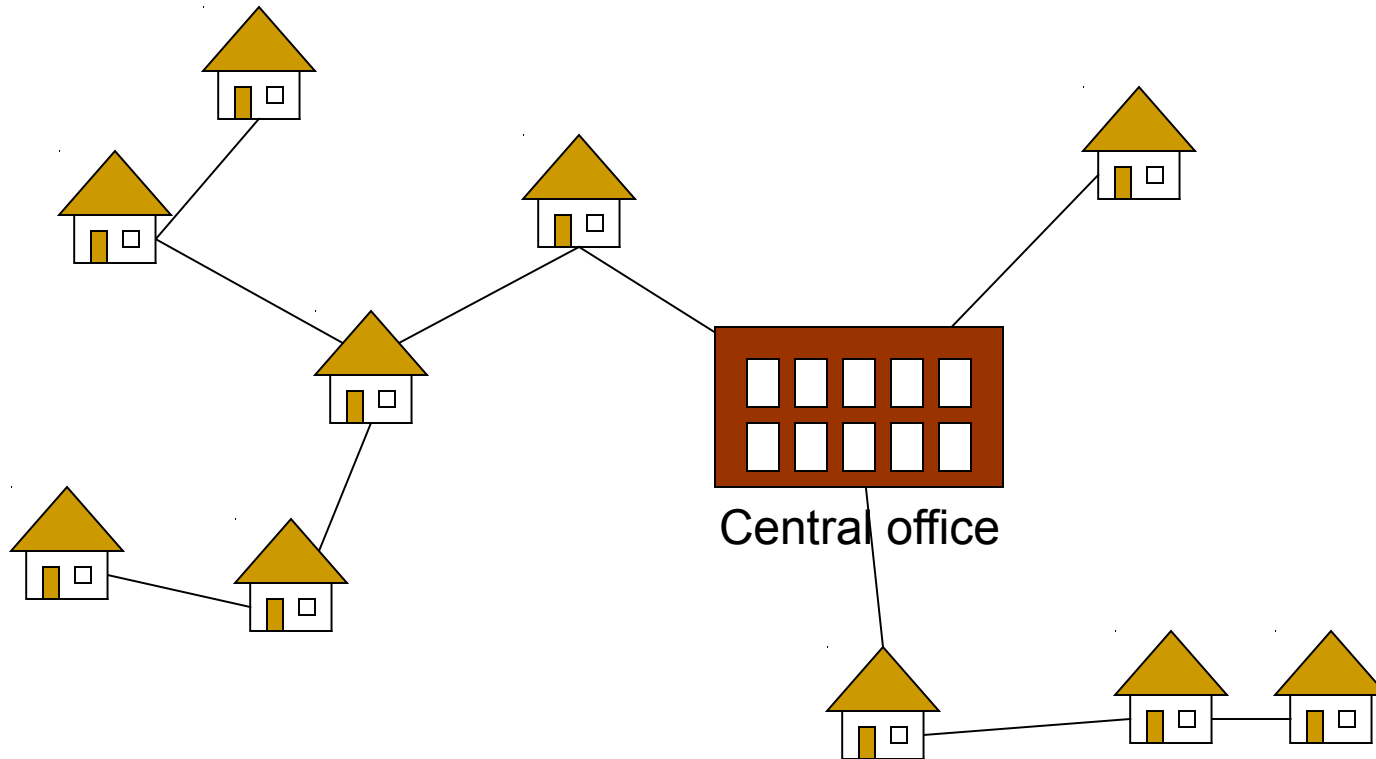
- **Srivathsan Srinivasagopalan**, Costas Busch, S.S. Iyengar, "[An Oblivious Spanning Tree for Buy-at-Bulk Network Design Problems](#)", Journal Paper in preparation.
- **Srivathsan Srinivasagopalan**, Costas Busch, S.S. Iyengar, "[Universal Data Aggregation Trees for Sensor Networks in Low Doubling Metrics](#)", in *International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, ALGOSENSORS'09, Rhodes, Greece, July 10-11, 2009. LNCS, Springer Verlag, Vol 5804/2009, pp 151-152.
- **Other:** $O(\log D \log^2 n)$
 - Oblivious Network Design in H-Minor Free Graphs
 - Working result - $O(\log^3 n)$ approximation
 - Oblivious Network Design in Arbitrary Graphs.
 - Working result - approximation.

Wiring: Naïve Approach



Expensive!

Wiring: Better Approach



Minimize the total length of wire connecting the customers

Motivations and Contribution

- **[Goel, Estrin - SODA 2003]:** Data Aggregation Tree that is simultaneously good for all canonical fusion-cost functions f which are concave and non-decreasing.
 - Hierarchical Matching Algorithm
 - Provide $(1+\log k)$ - approximation., k being the number of sources.
 - **However, every time the data sources change, the aggregation tree needs to be reconstructed**
- **[Jia, Noubir, Rajmohan, Sundaram - DCOSS 2006]:** Builds a single aggregation-tree that is independent of the number of sources.
 - Randomly distributed nodes in a 2-D plane (extended for 3-D also).
 - Provides a $O(\log n)$ - approximate aggregation cost and $O(1)$ delay.
 - **The solution does not consider fusion-cost functions.**
- **[Gupta, M.T. Hagiaghay, Racke - SODA 2006]:** Framework for oblivious network design.
 - For Agg-Function $\sum$, competitive ratio of $\log^2 n$ is presented.
 - For Agf-Function $\max$, competitive ratio of $\log^2 n \log \log n$ is given.
 - **Randomized algorithm !! Not clear about the cost of derandomization.**

Some Observations (in sensor networks)

- Sensed data (usually) traverses a multi-hop path to the destination.
- We do not know how many sensors send their data at a particular time.
- We do not have prior information about the cost of data fusion in the intermediate nodes.
 - Can assume that the fusion-functions are concave and non-decreasing.
 - We assume that the aggregation cost follows economies of scale, that is, the incremental cost of a new user is less if the set of users already served is larger.

This leads to the question :

“Is there a single data structure for data-aggregation that provides near-optimal total aggregation-cost when the number of data-sources and the fusion functions (at intermediate nodes) are unknown?”

Formal Problem Statement

- Given a network $G = (V, E)$, a set of sources s_i and a sink s , find a route for each of the sources s_i to the sink.
- Let D_e be the demand i routed through an edge e and let $x_e = \sum_{i \in D_e} D_i$ be the total demand routed through edge e .
- Total Cost required for bandwidth on edge e is $f_e(x_e)c_e$ where f_e is a concave cost function and c_e is the cost of the edge.

Problem Statement

- Total cost of the solution is $\sum_e f_e(x_e)c_e$

- **[GOAL]** Competitive Ratio:

$$\text{Minimize } \left\{ \frac{OBL_{agg}(f, x)}{OPT_{agg}(f, x)} \right\}$$

Obivious Algorithm

Optimum value

For all concave cost functions f and set of demands x .

Work on Doubling Dimension Graphs

Work on Doubling Dimension Graphs

Contributions

[Doubling Dimension Graph]

Build a *universal* Aggregation tree that is independent of the number of data sources and the fusion-cost functions.

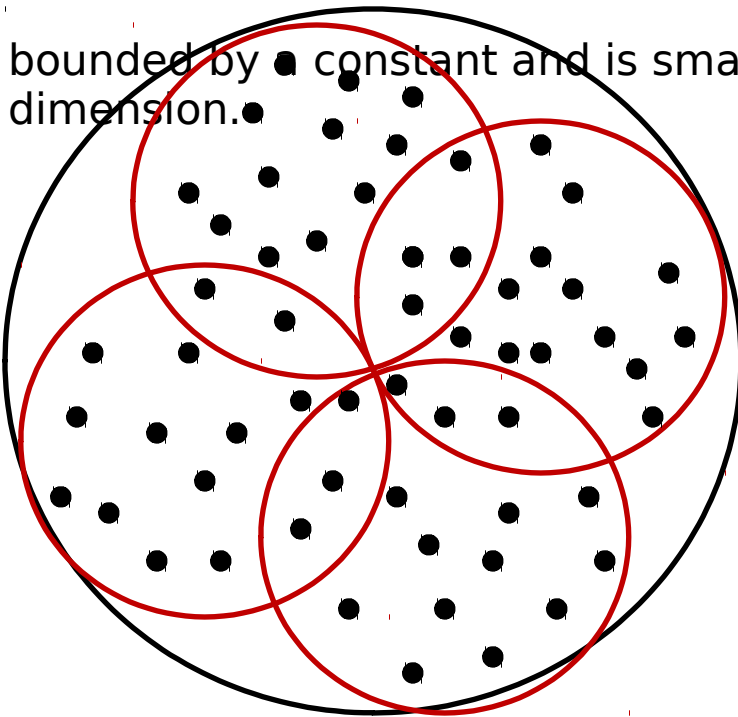
- On Low Doubling Metrics
- Provides a $O(\log^3 n)$ -approximation to optimal aggregation cost (measured as total involved aggregation cost).

Definition: Doubling Dimension Graphs

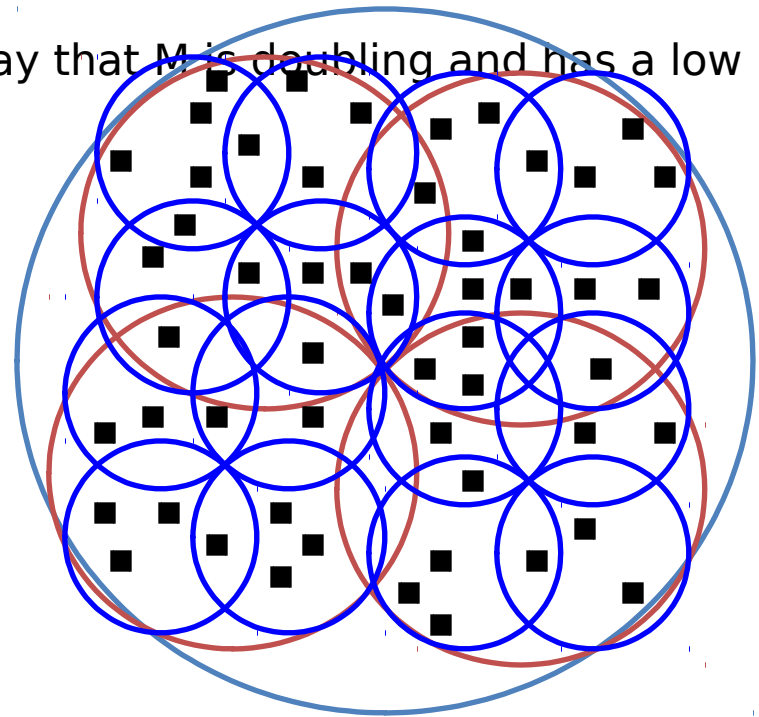
Definition: Let $M = (X, d)$ be a metric space. The doubling dimension of M is the smallest k such

that every ball of radius r can be completely covered by at most 2^k balls of radius $r/2$. If it is

bounded by a constant and is small, we say that M is doubling and has a low dimension.



All the points can be covered by 2^2 balls of radius $r = R/2$, R being the radius of the larger ball.



Each of 4 (red) balls can be covered by 2^2 balls of radius $r/2$, r being the radius of the larger ball.

Definition: Universal Steiner tree and Universal Aggregation tree

Universal Steiner Tree: A single Minimum Steiner Tree that holds for all instances of  that needs to be connected.

Universal Aggregation Tree: A Universal Steiner with fusion cost function  at every vertex.

- Minimum Steiner tree problem is NP-Complete [2].
- Polynomial-time constant factor approximation heuristics exist [3].

Universal with respect to data sources and cost function.

[2] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness.

W. H. Freeman & Co., New York, NY, USA, 1990.

[3] Vijay V. Vazirani. Approximation Algorithms. Springer. March 2004

Problem Hardness

- Lowest cost aggregation tree  Steiner Tree problem (when $f(x) = 1$)
Weighted Steiner Tree problem (when $f(x) > 1$).
- **Rectilinear Steiner Tree Problem (RSP)**
 - Variation of Steiner Tree problem.
 - Has only vertical and horizontal lines that interconnect all points.
 - Proved to be NP-Complete due to Garey and Johnson [1,2].
- **Minimum weight tree structure in a 2-D grid topology**
 - Doubling dimension graph
 - Equivalent to RSP
 - Hence, NP-Complete
- **We are looking for theoretically guaranteed bounds, not empirical bounds**

[1] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is np- complete. SIAM Journal on Applied Mathematics, 32(4):826{834, 1977.

[2] Michael R. Garey and David S. Johnson. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA, 1990.

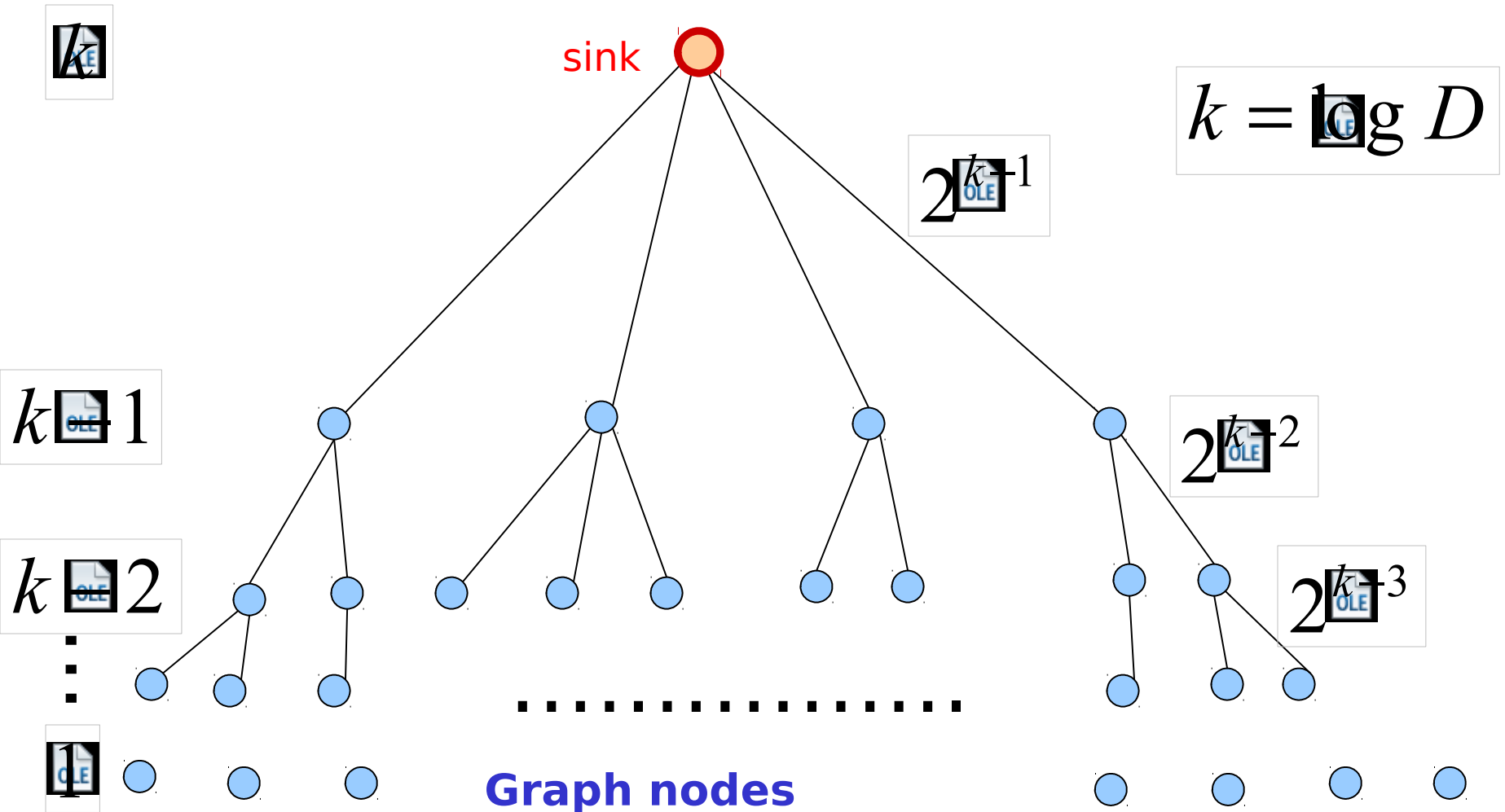
Technique Used (rough sketch)

Hierarchical partition of the graph

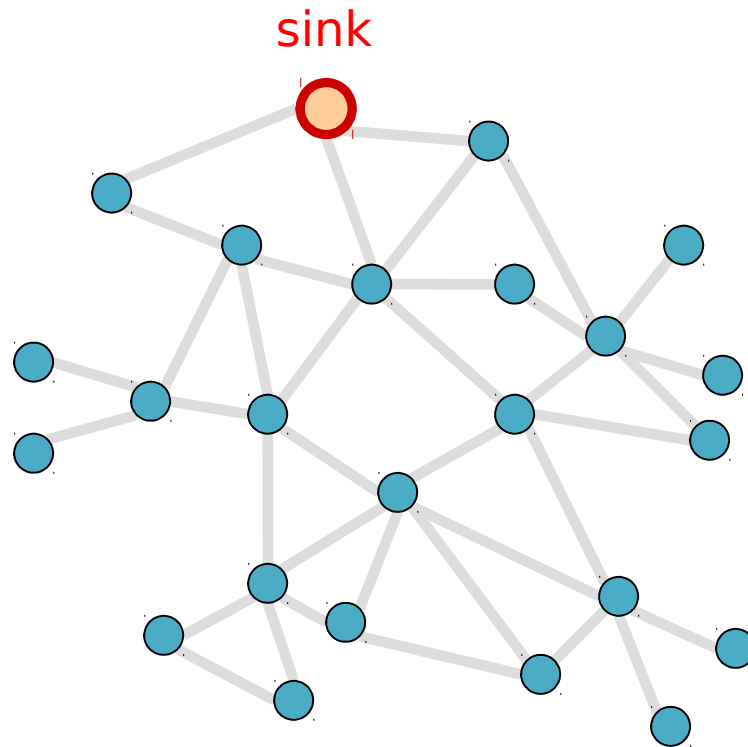
- Maximal Independent Set at each level.
- Potential leaders are the MIS Nodes at each level.
- Connect leaders of successive levels by shortest path.
 - Leads to an **overlay** tree
- Build a **real spanning tree** from overlay tree
 - Pruning process
 - Election of alternate leaders

An Overlay Tree

Layers of independent sets of leaders

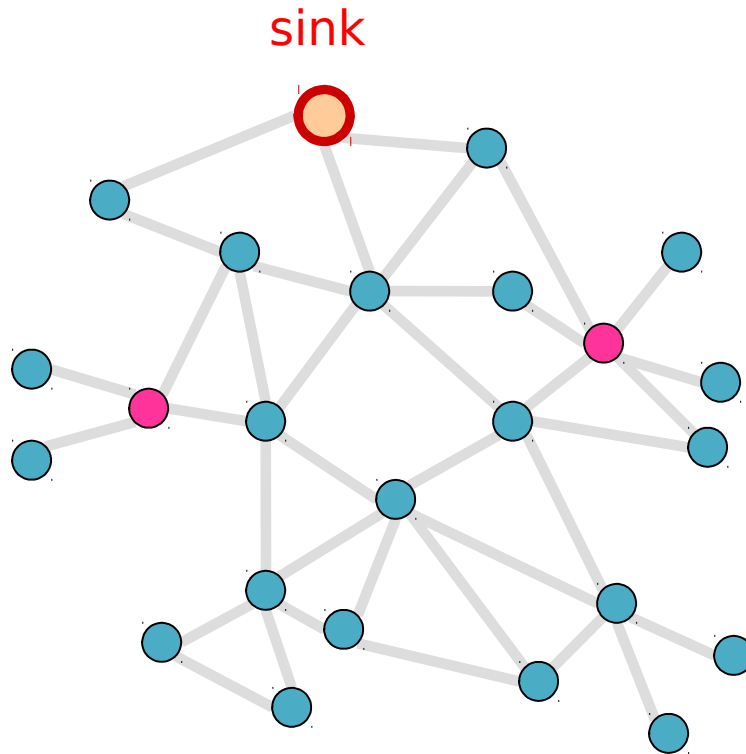


Layer k



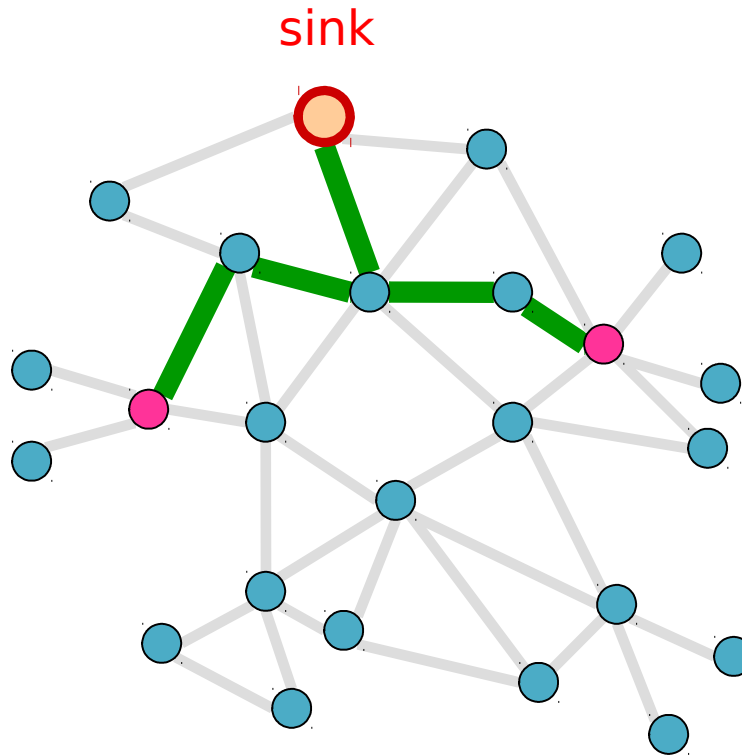
Project Overlay Tree to Graph

Layer k-1



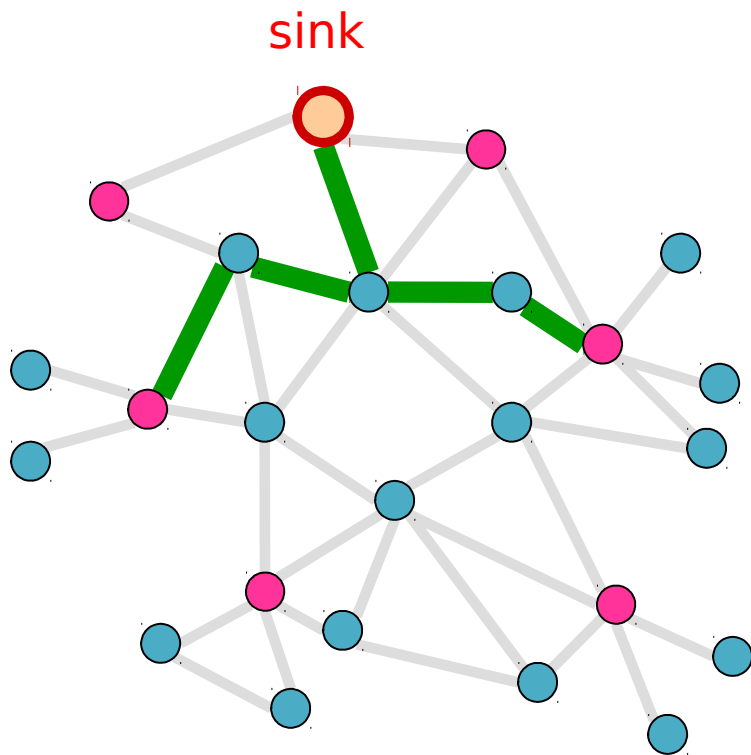
2^{k-1} distance between leaders

Layer k-1



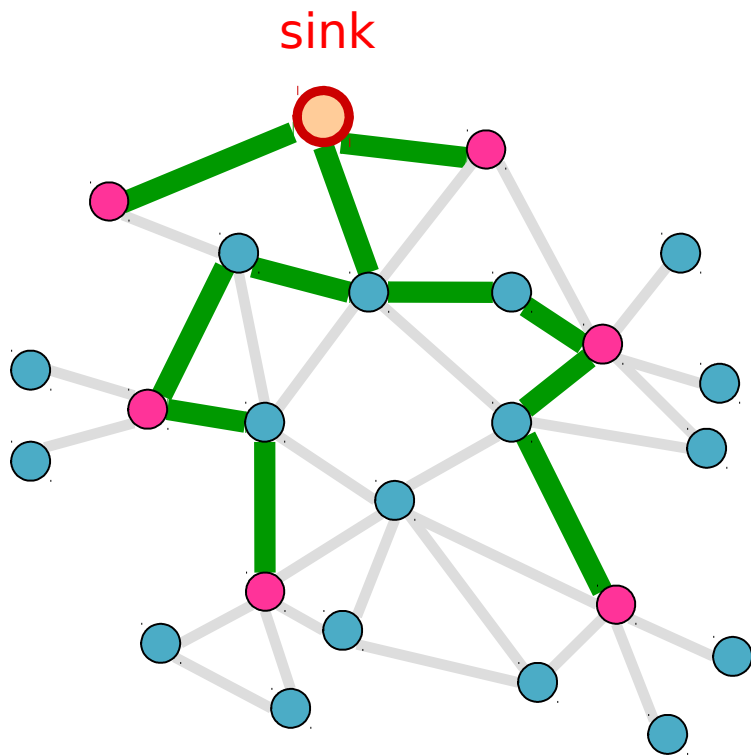
**Shortest path to nearest
higher layer leader**

Layer k-2



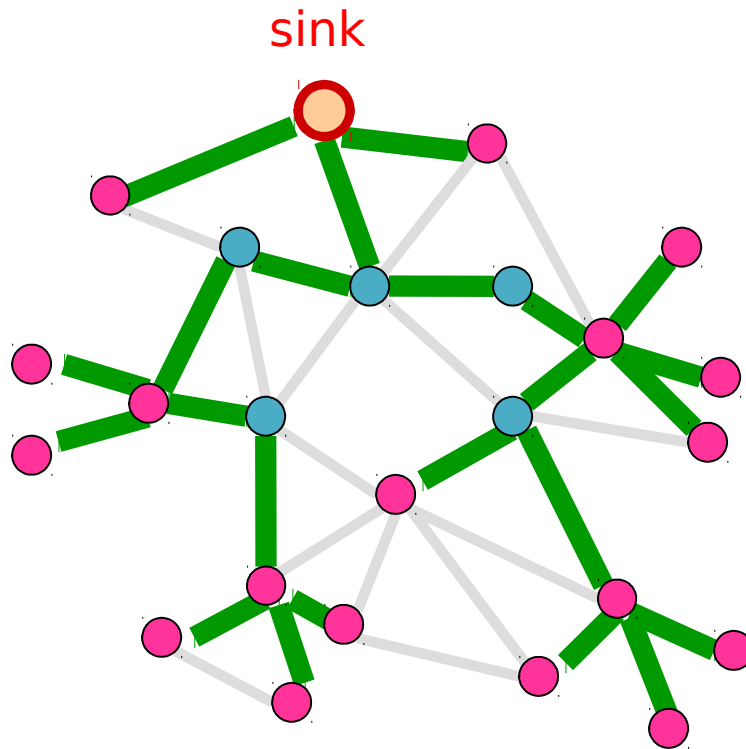
2^{k-1} distance between leaders

Layer k-2



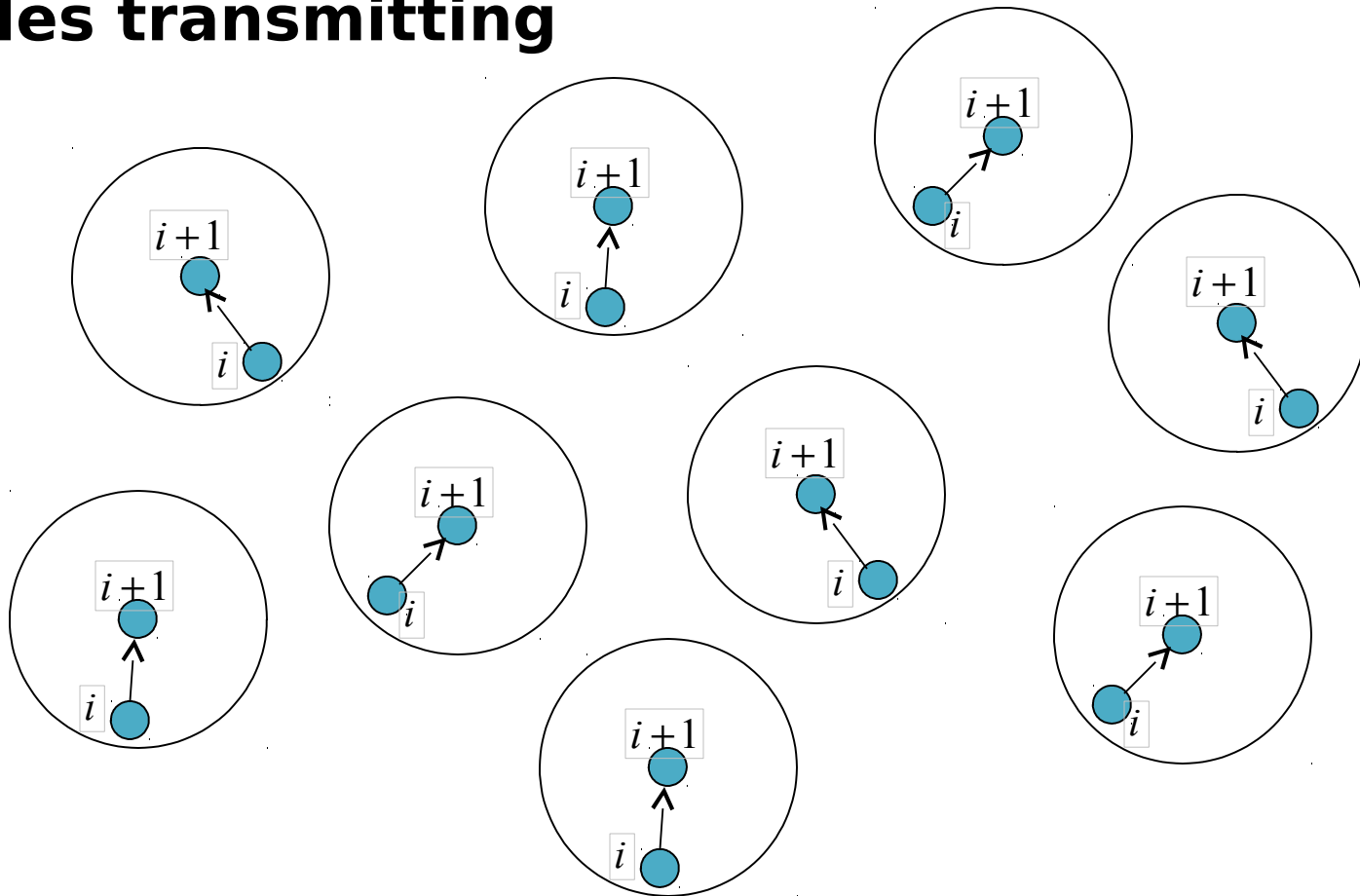
**Shortest path to nearest
higher layer leader**

Layer k-3



Analysis

Nodes transmitting

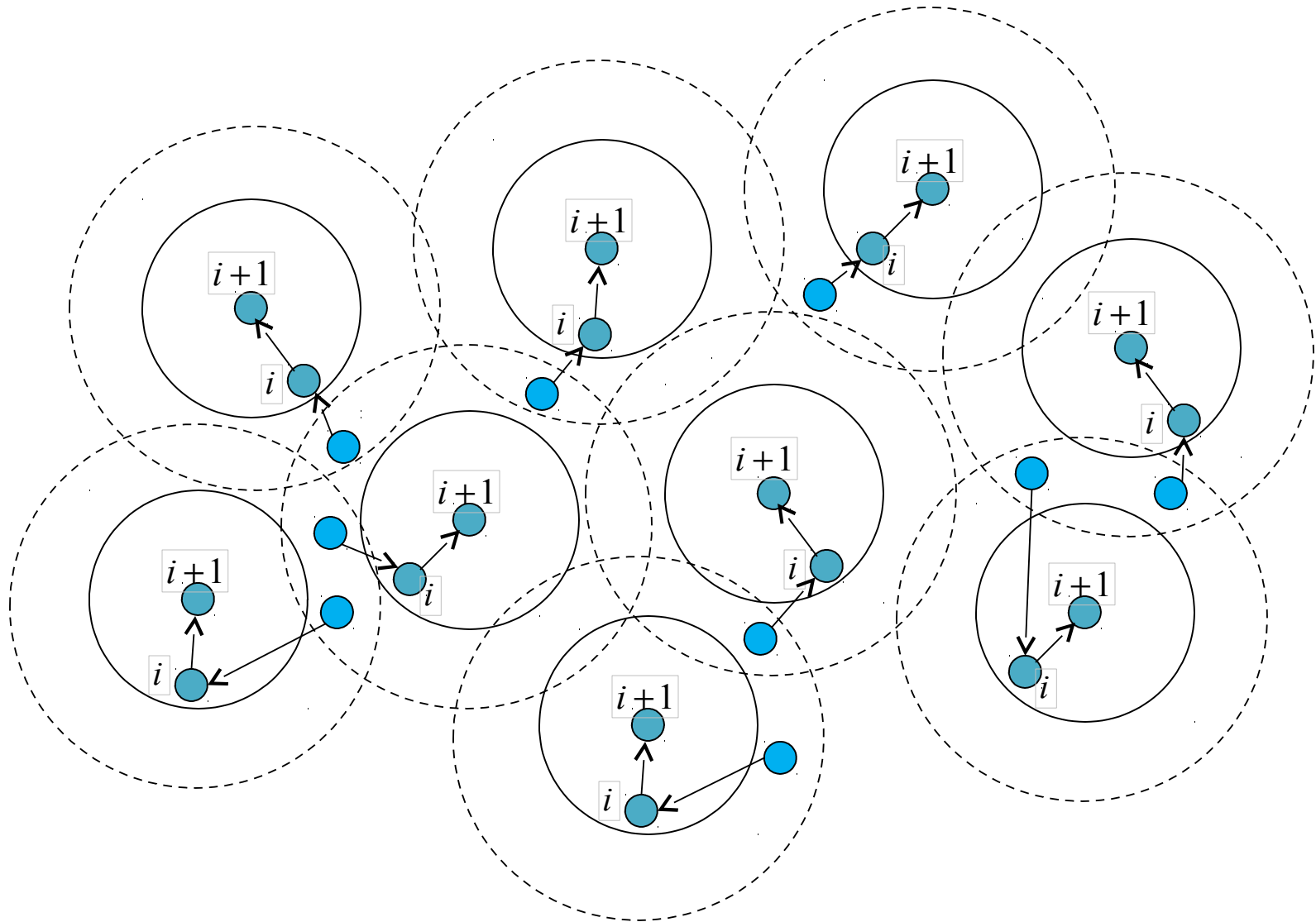


Layer
Cost:

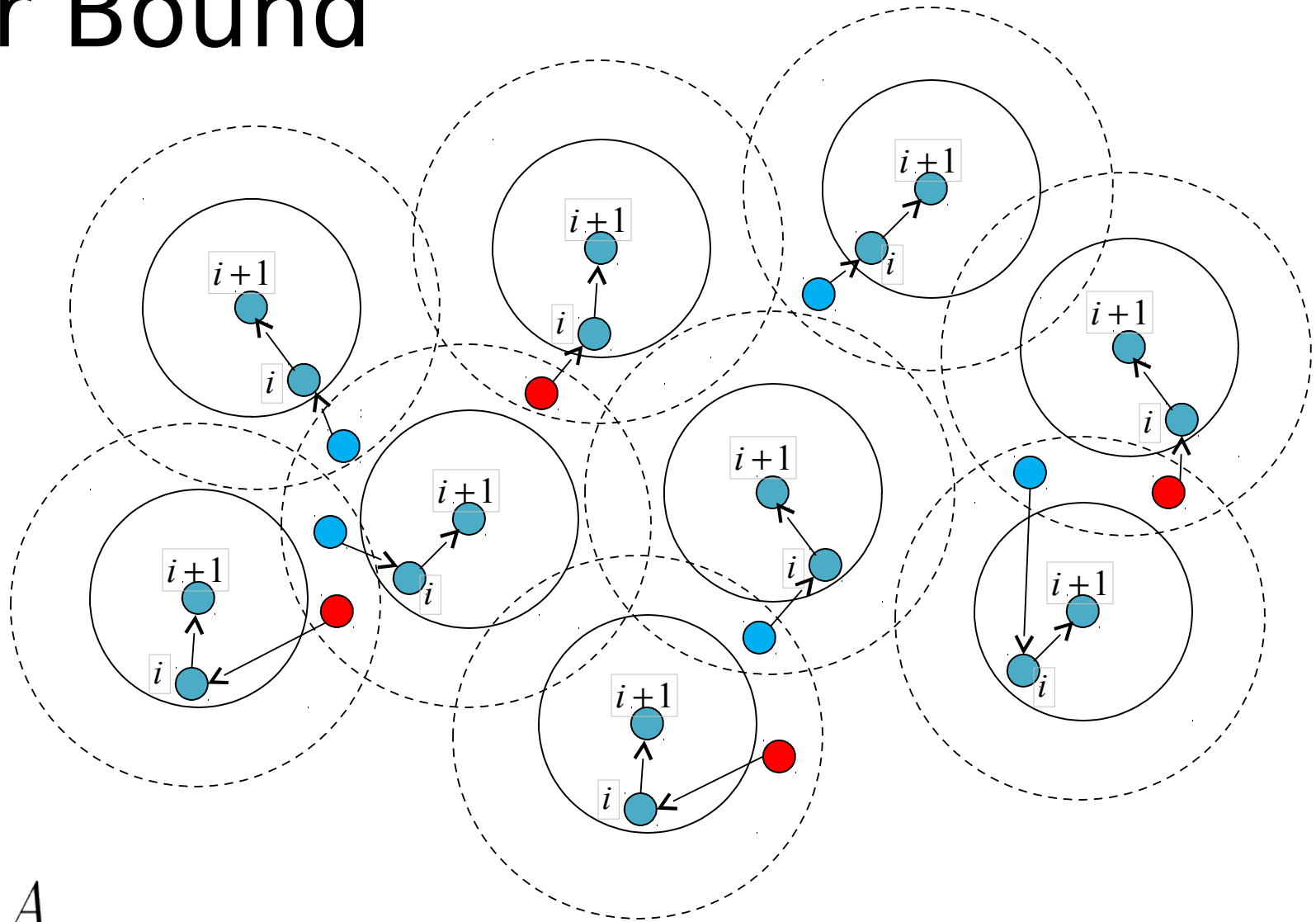
$$2^i \times |T|$$

$$f(x) = 1$$

Lower Bound



Lower Bound



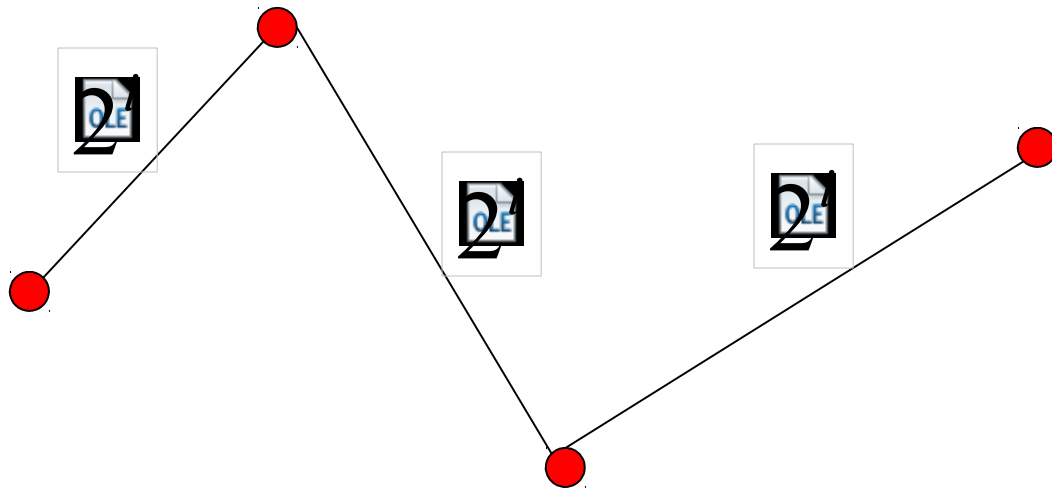
$$T = c \cdot A$$

Majority of the real
sources

$A \subseteq T$ are at distance 2

Lower Bound

Steiner tree cost



At least: $T \times 2^i$

$$f(x) = 1$$

Layer Cost: $C^*(T) \leq |T| \cdot 2^i$

Lower Bound: $C^*(A) \geq |A| \cdot 2^i$ $[T = c \cdot A]$

$$C^*(A) \geq C^*(T) \geq |A| \cdot 2^i$$

Competitive Ratio: $\frac{\text{Layer Cost}}{\text{Lower Bound}} =$ 

Considering all layers:

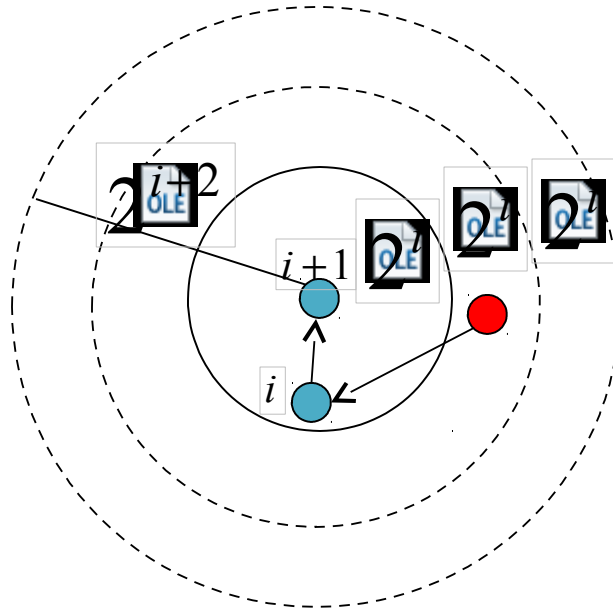
$$c \cdot \log n$$

**Conversion of overlay tree to a
spanning tree**

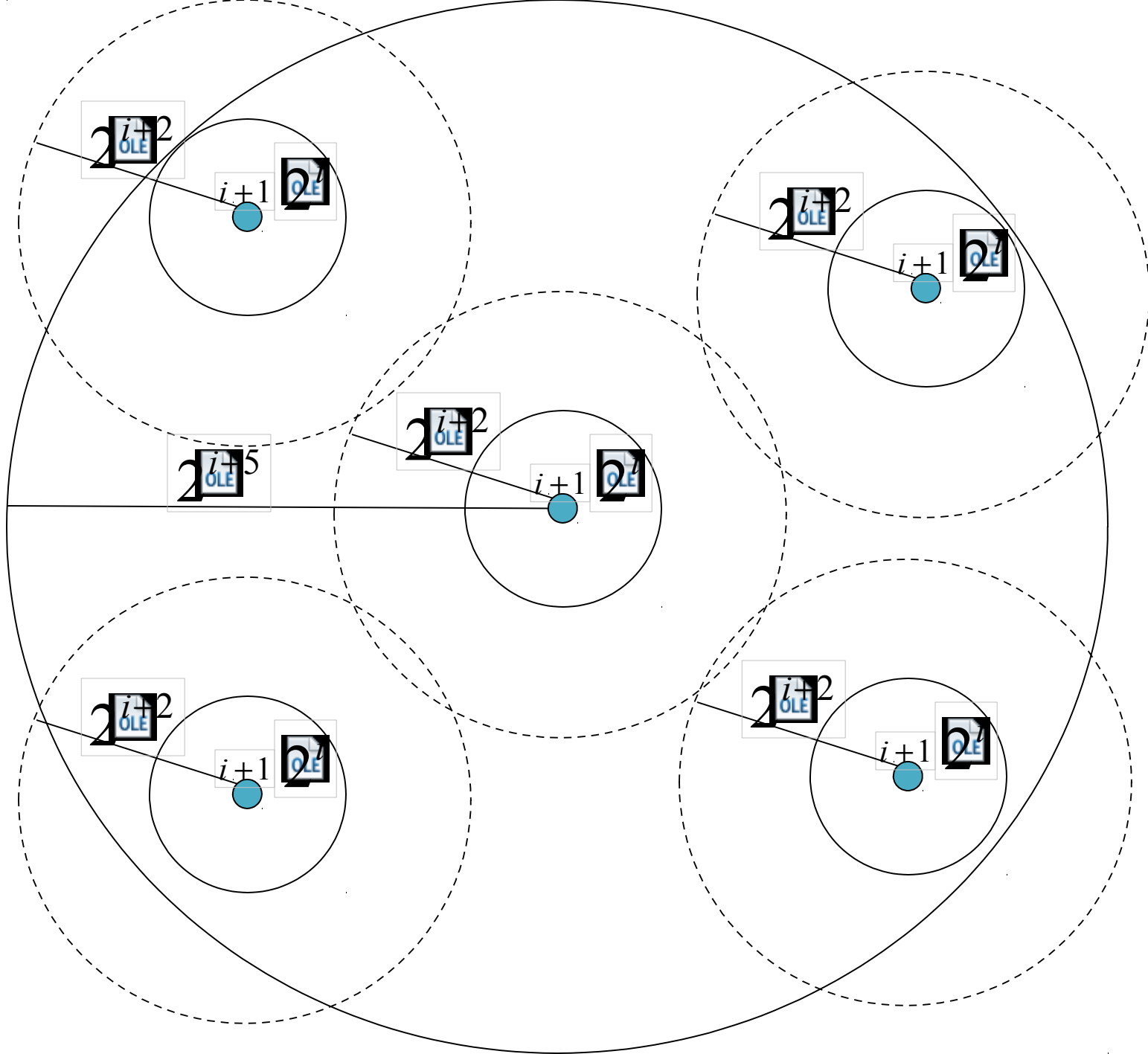
Spanning tree:

$$c \cdot \log^3 n$$

Computation of C :



There is an upper-bound on the number of data sources with a ball of certain radius.
This property holds true for Doubling-Dimension Graphs.



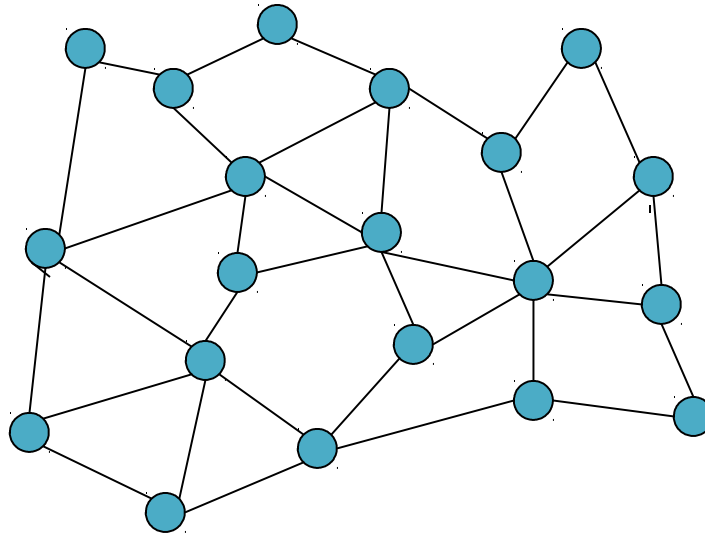
How many balls of radius 2^i

fit in an area of radius 2^{i+5} ?

Observation 2.3. In any R -neighborhood of doubling dimension graphs, there are at most m $R/(2^j)$ -neighborhoods where $m = 2^{p \log(\frac{R}{R/(2^j)})}$, $j \geq 0$.

$$m = 2^{p \log\left(\frac{2^{i+5}}{2^i}\right)} = 2^{p \log 2^5} = 2^{5p}$$

Neighborhood Graph



nodes

Max degree:

$$m \leq 2^{5p}$$

An edge between nodes i and j



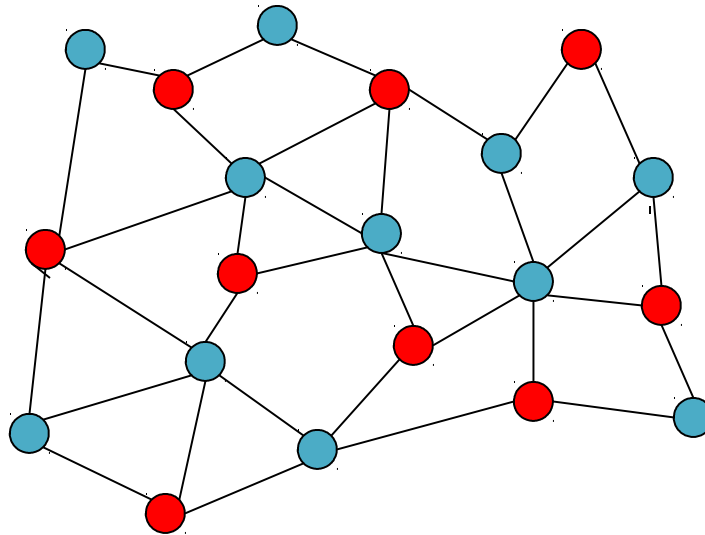
discs overlap.

We show that there are significant number of source nodes that are spaced apart.

Compute MIS of the above neighborhood graph (a linear time algorithm using

Greedy Approach with inputs in the form of adjacency list).

Neighborhood Graph



14 nodes

Max degree:

$$m \leq 2^{5p}$$

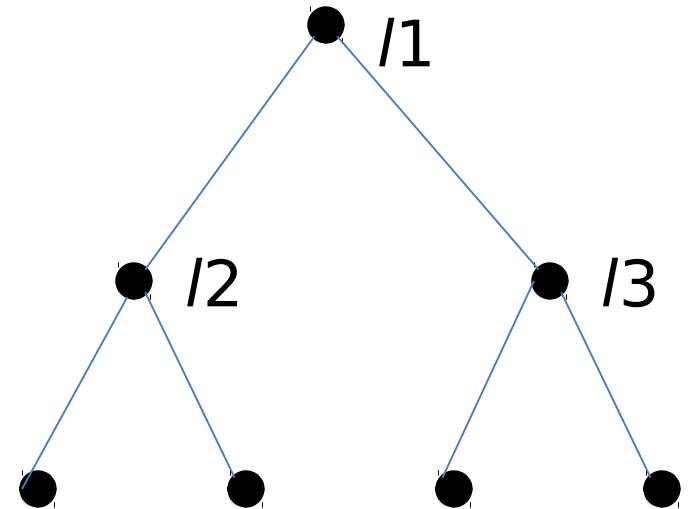
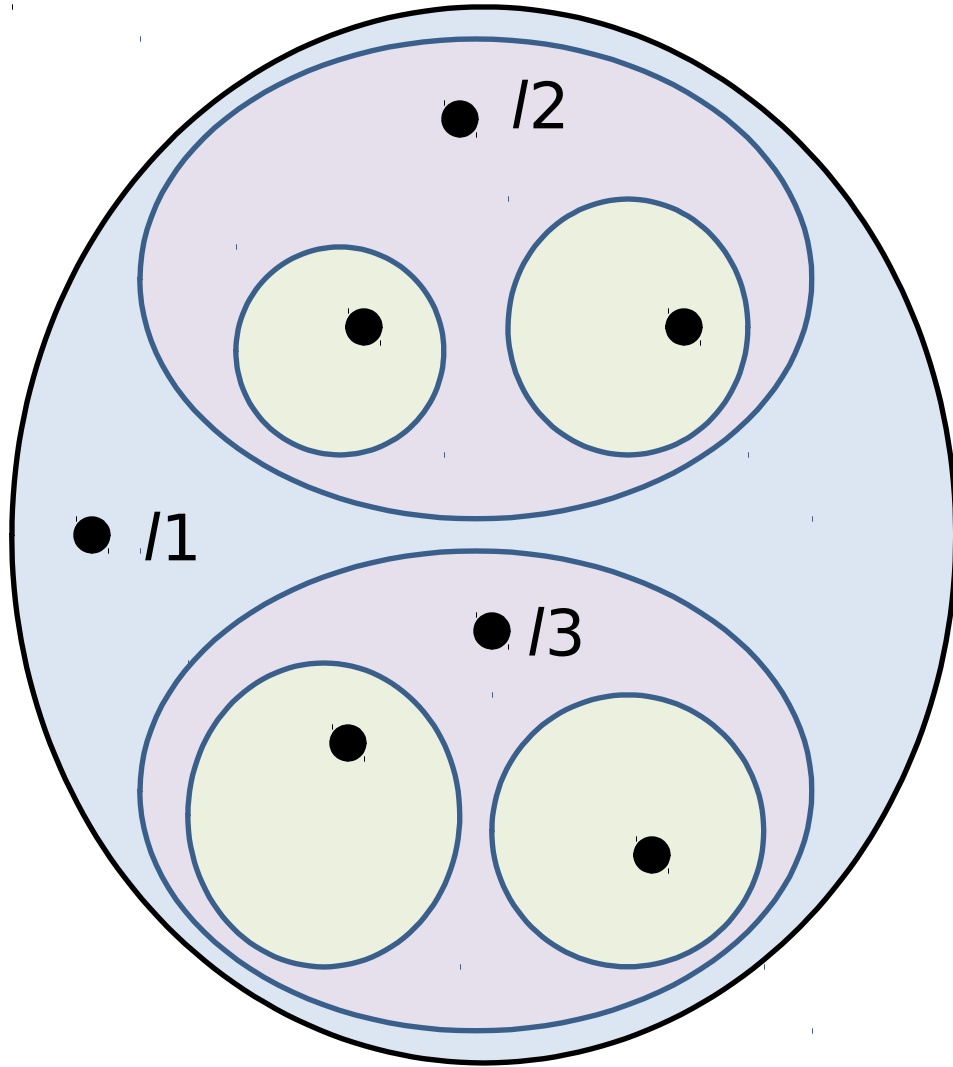
edge between nodes if 2^{i+2} discs overlap

accepts $m+1$ coloring

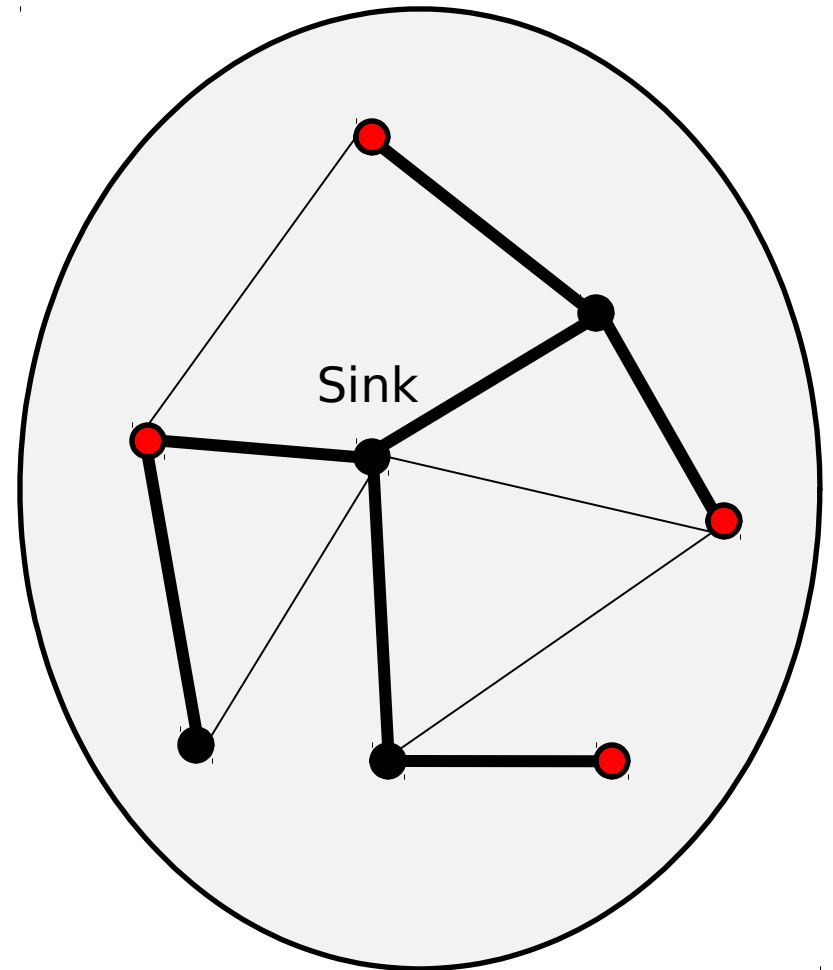
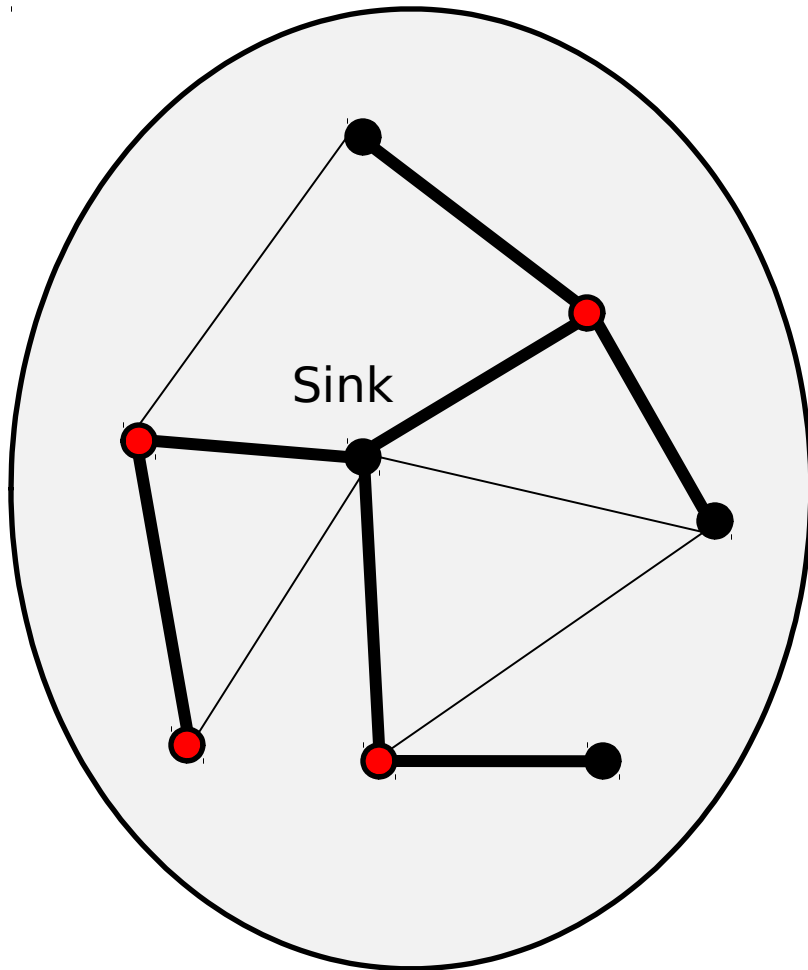
Contains independent set of size

$$\frac{|A|}{m+1} \leq \frac{|A|}{c}$$

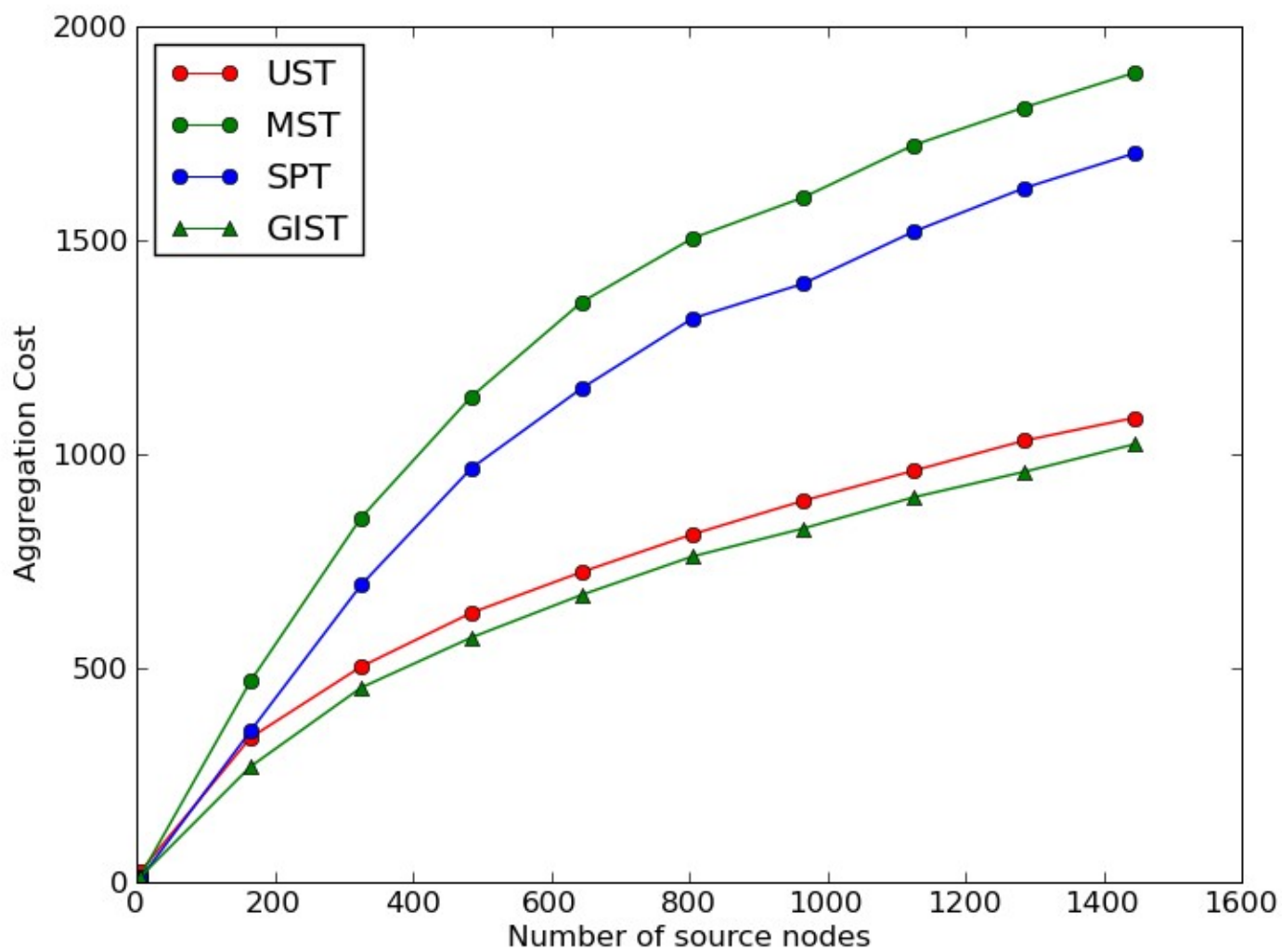
Hierarchical Clustering

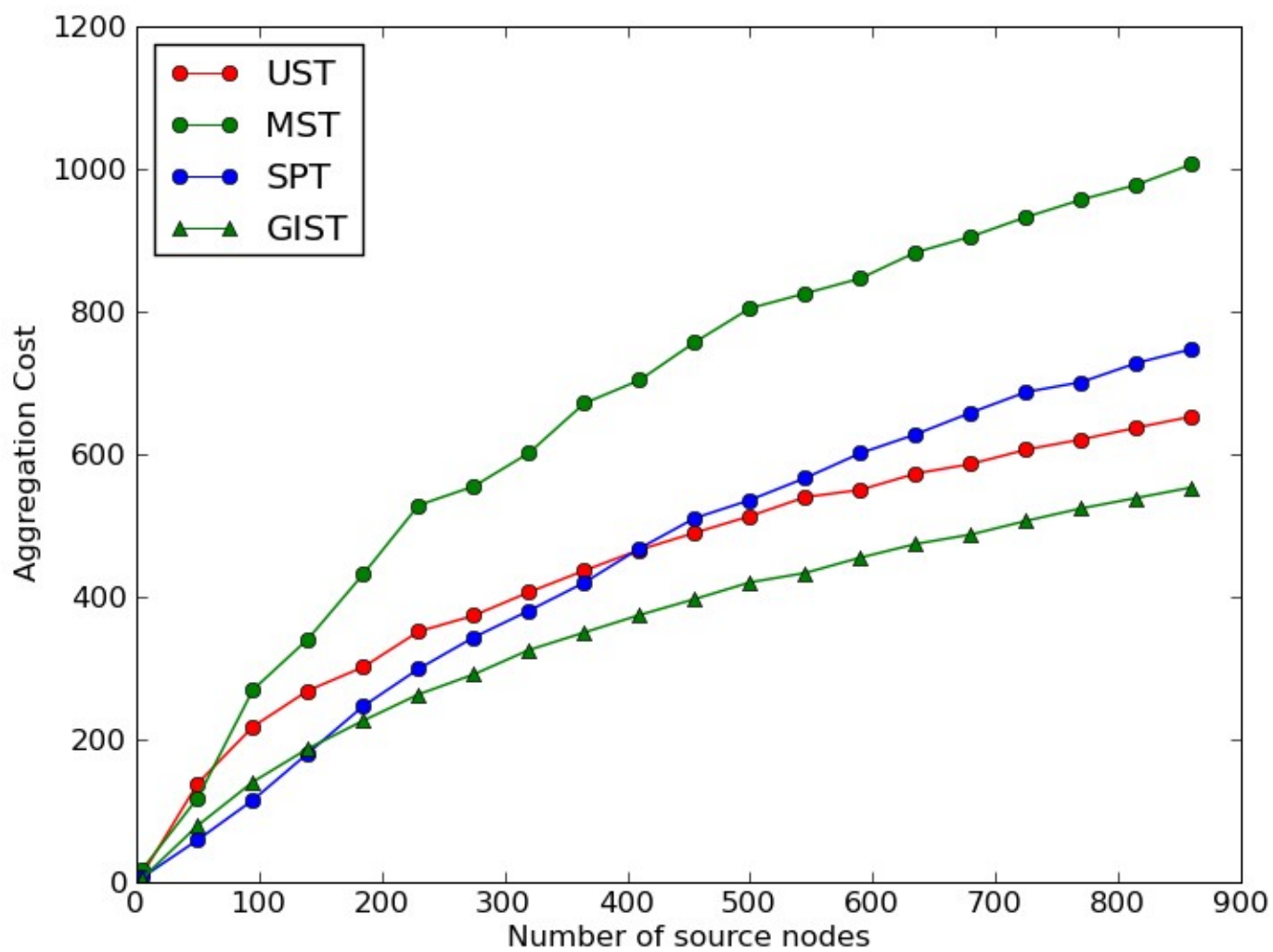


Universal Tree



Different Data-Source Sets





Work on Planar Graphs

Contributions

[Planar Graph]

2) Oblivious Network Design that is independent of the sources and aggregation cost function. We provide a deterministic algorithm for constructing an overlay tree.

- On planar graphs.
- Provides a $O(\log D \log^2 n)$ -approximation to optimal aggregation cost (measured as total involved aggregation cost).
- **Conversion/mapping to a spanning tree is currently being done.**

Technique Used

- Low-Color partitions
- Partition Types
 - Weak-Diameter : Overlay Trees
 - Strong-Diameter: Spanning Trees
- Use of Path Separators for partitioning

Low Color Partitions

- Decomposition of a graph into several components (disjoint).
- Properties of this partition:
 - The components have bounded diameter γ

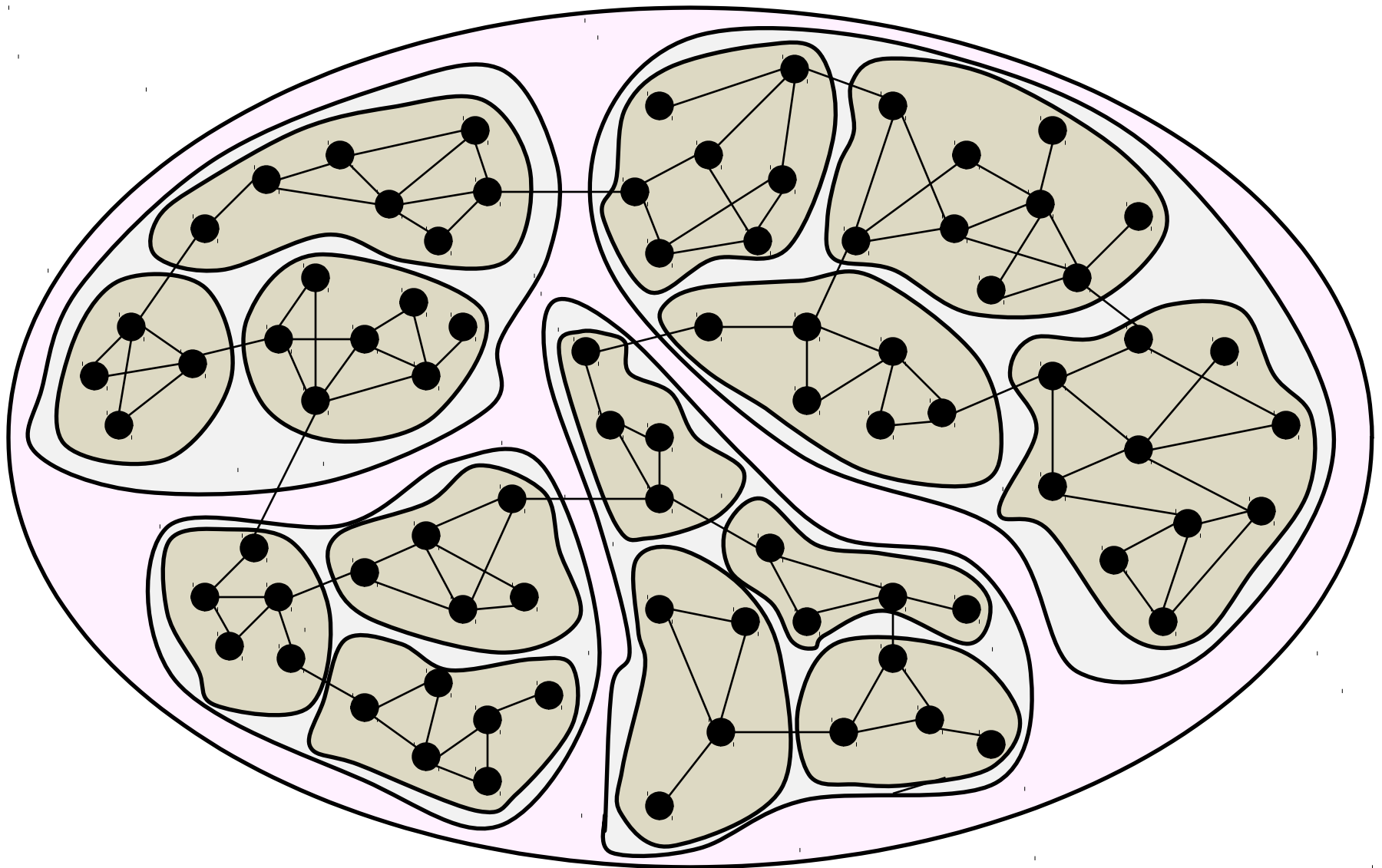
Coloring:

- Components that are “close” to each other cannot have the same color. Parameter

$$f(\gamma)$$

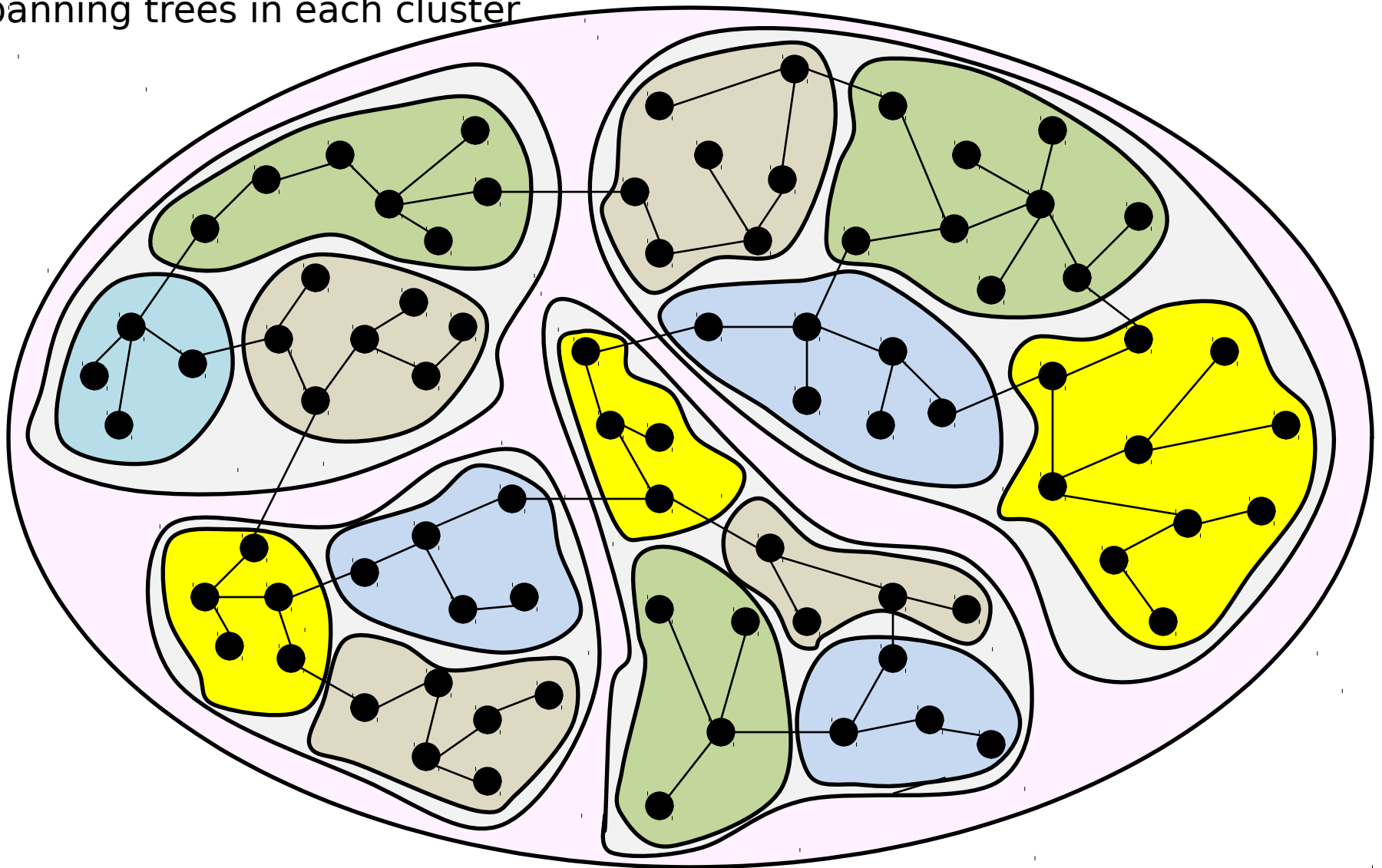
- Color the partition (at each level) with minimal # of colors.

Laminar Sets



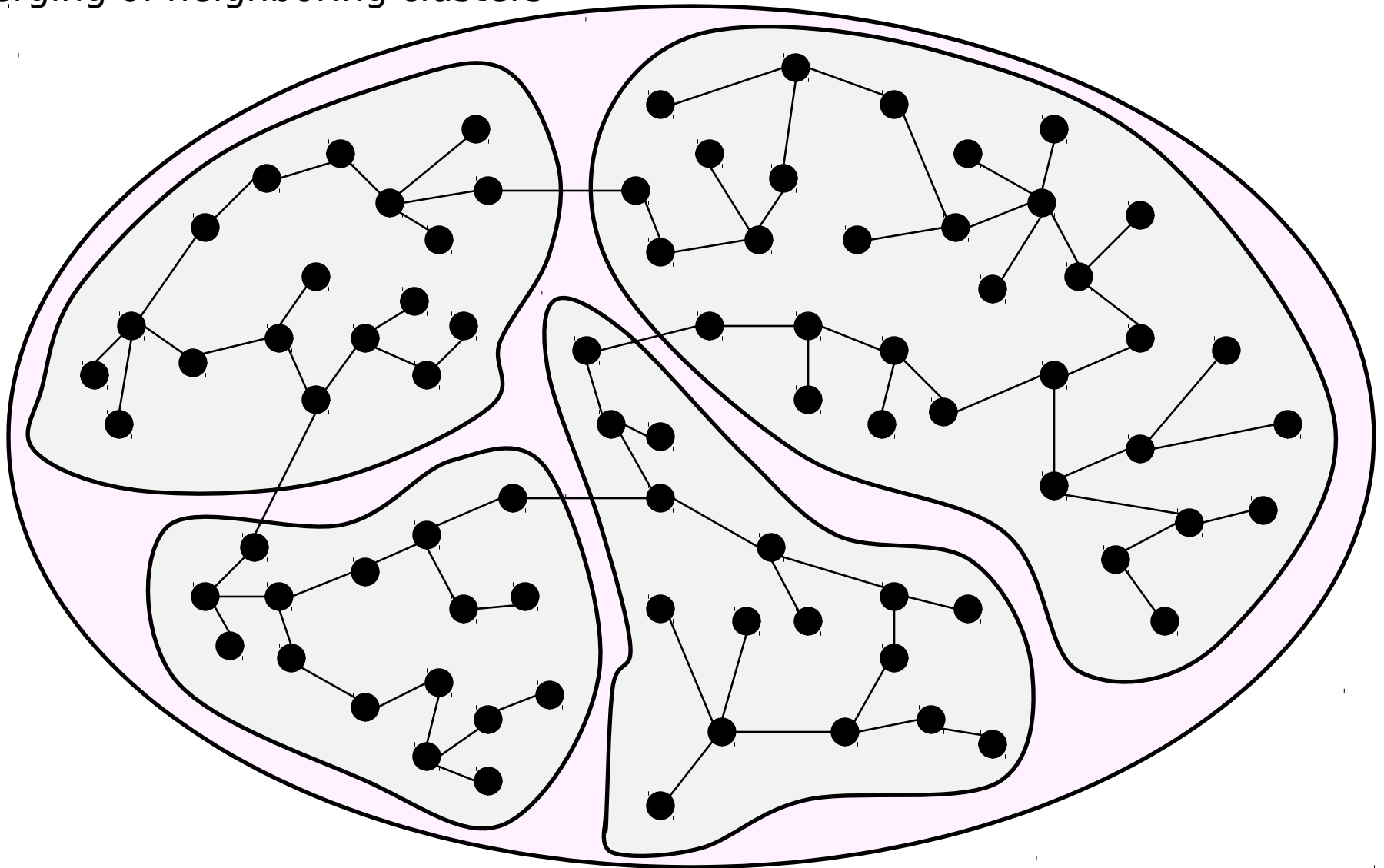
Laminar Sets

Coloring of lower-level clusters
Spanning trees in each cluster



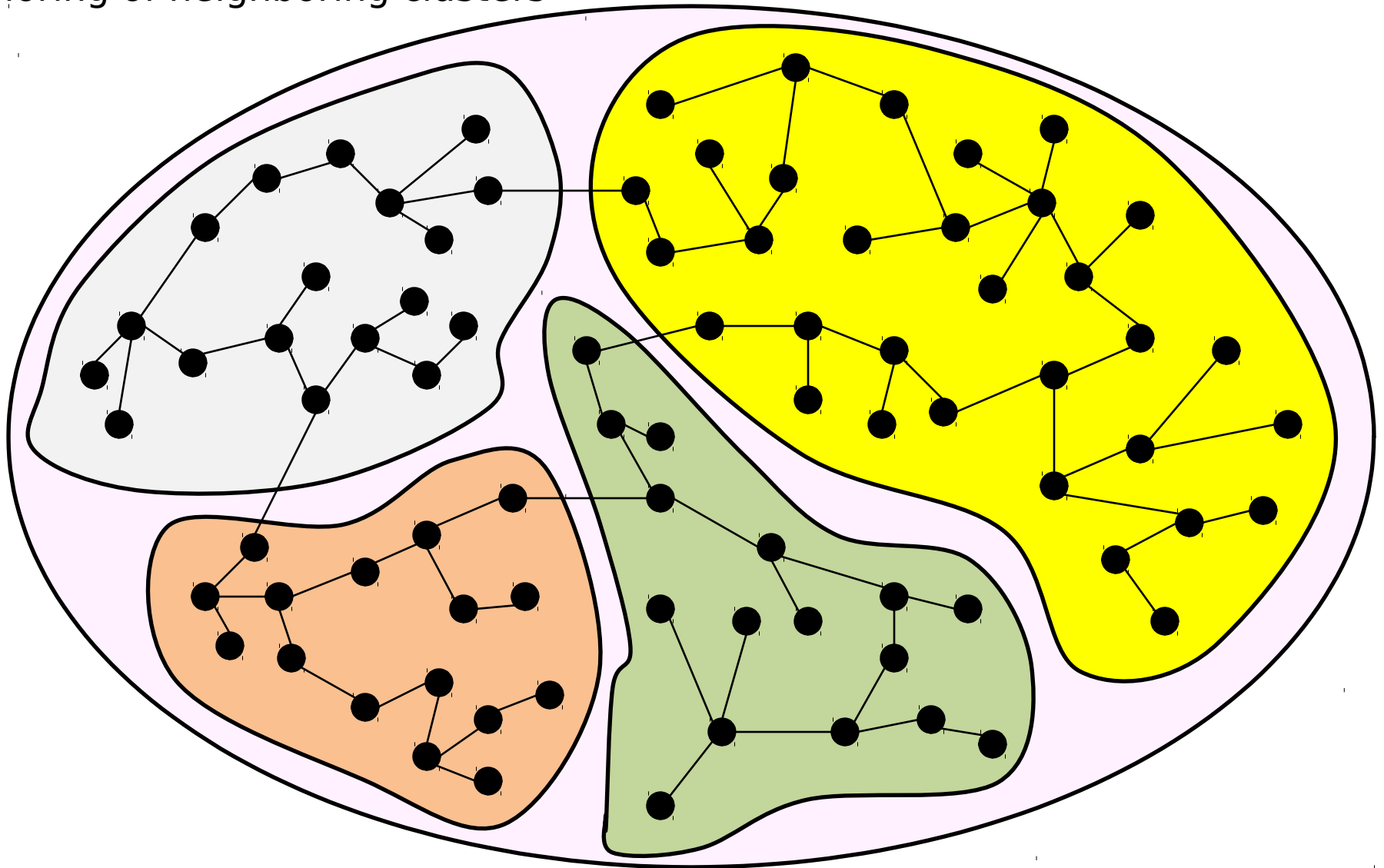
Laminar Sets

Merging of neighboring clusters



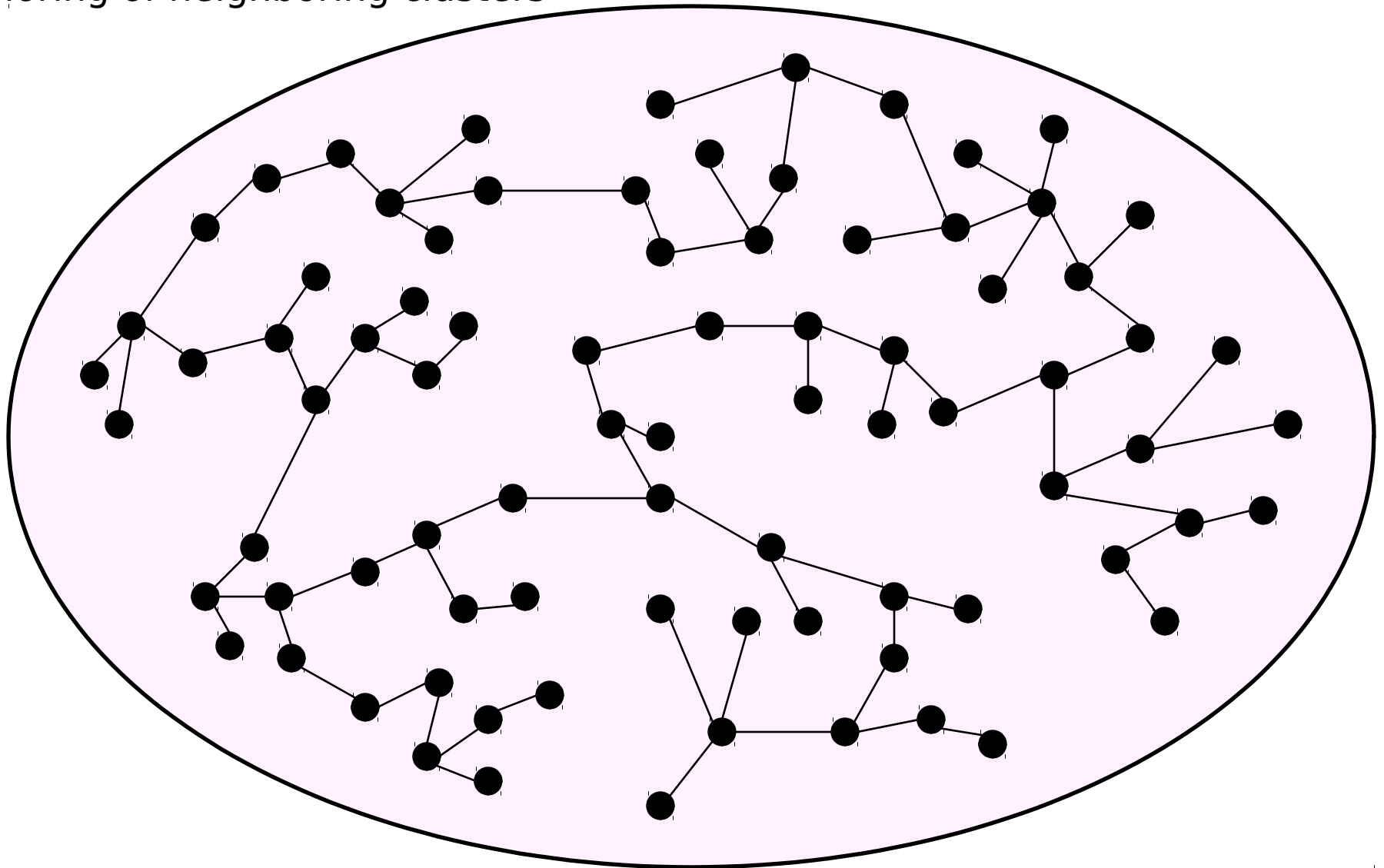
Laminar Sets

Coloring of neighboring clusters



Laminar Sets

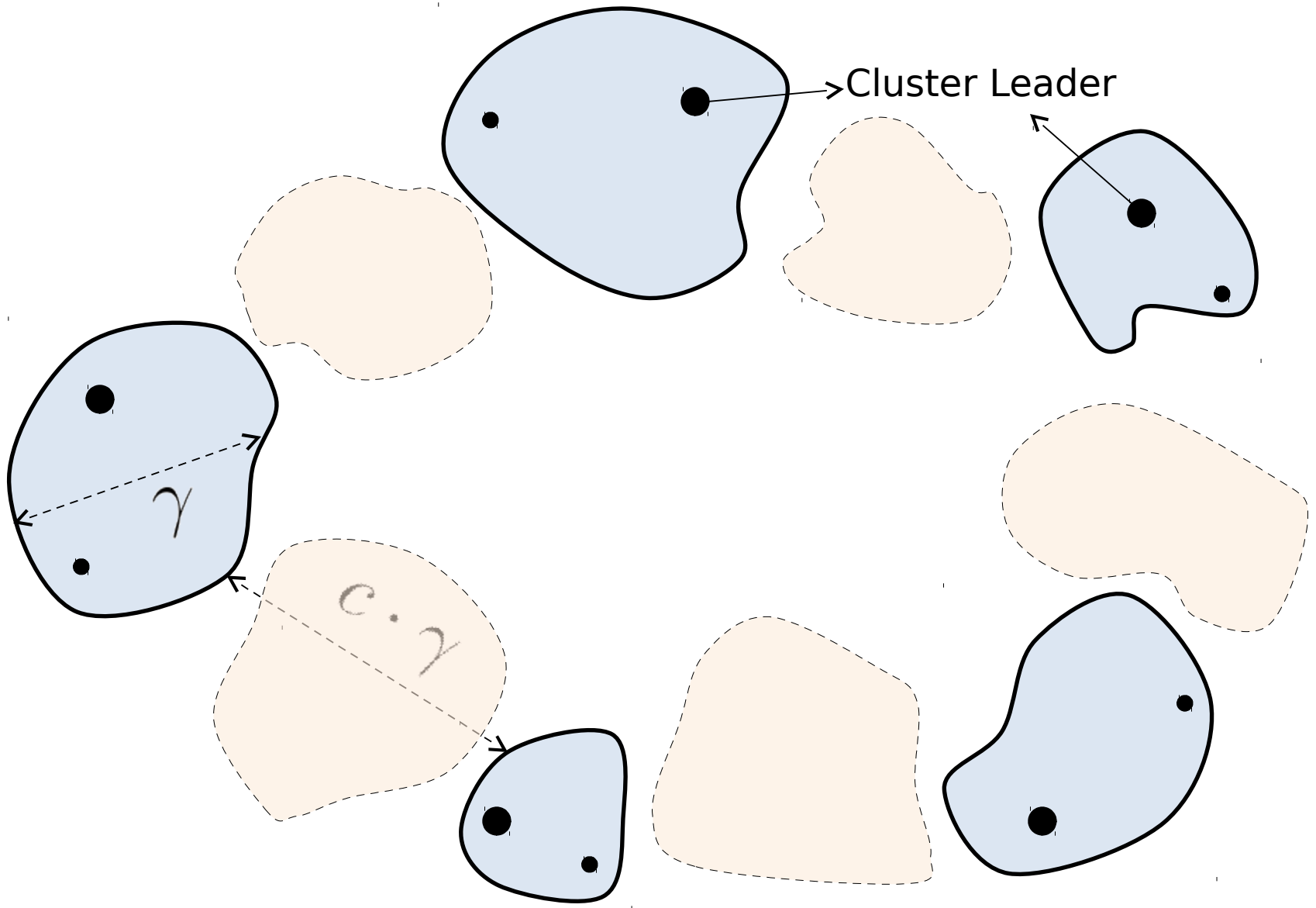
Coloring of neighboring clusters



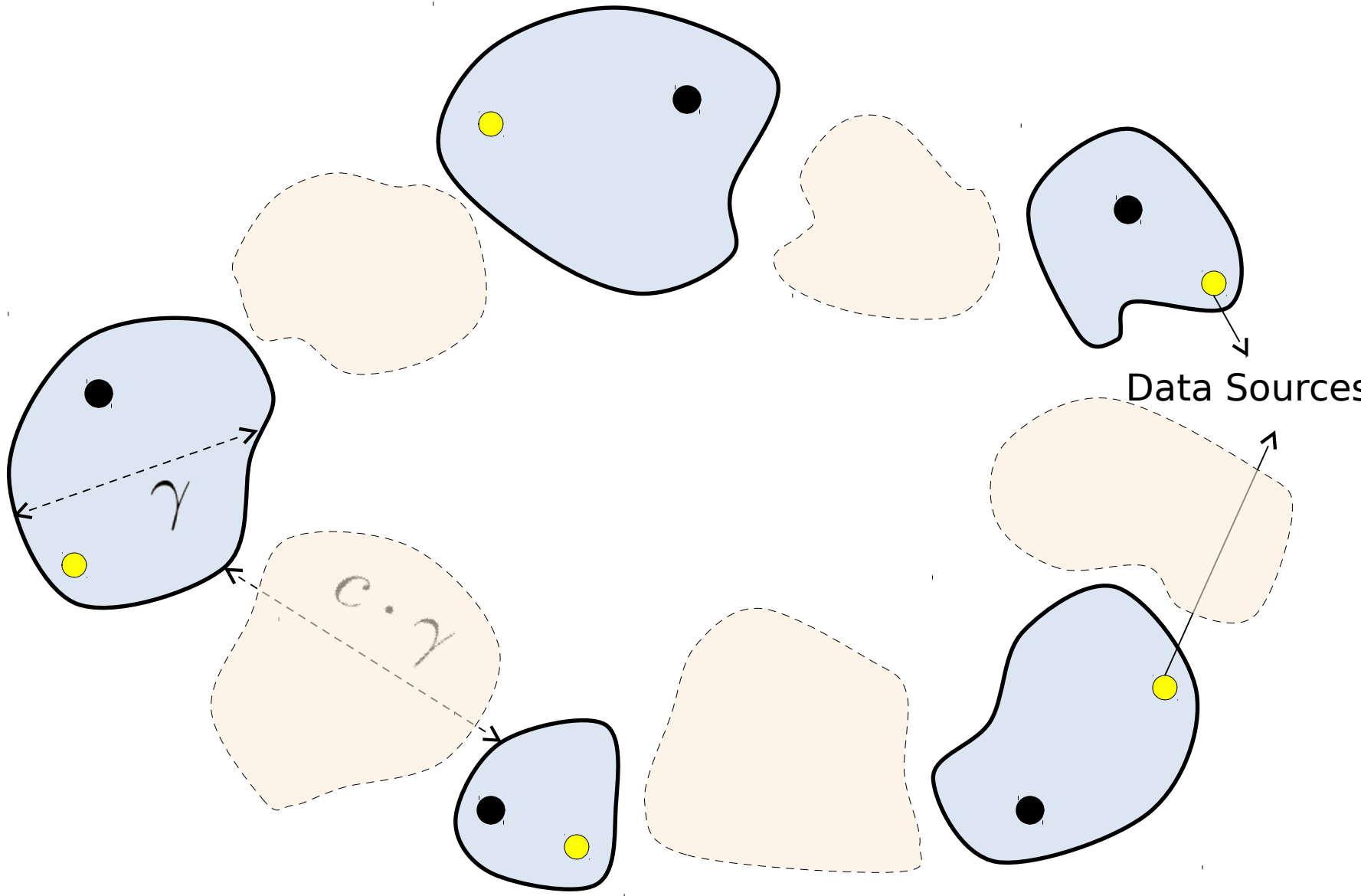
Why Low-Color Partitions?

- Clusters of same color are far away from each other.
- Leaders of these clusters are mutually far off.
- The real data sources that feed those leaders will also be mutually far away.
- The number of such real data sources that are mutually far away are significant (compared to those that are closeby).

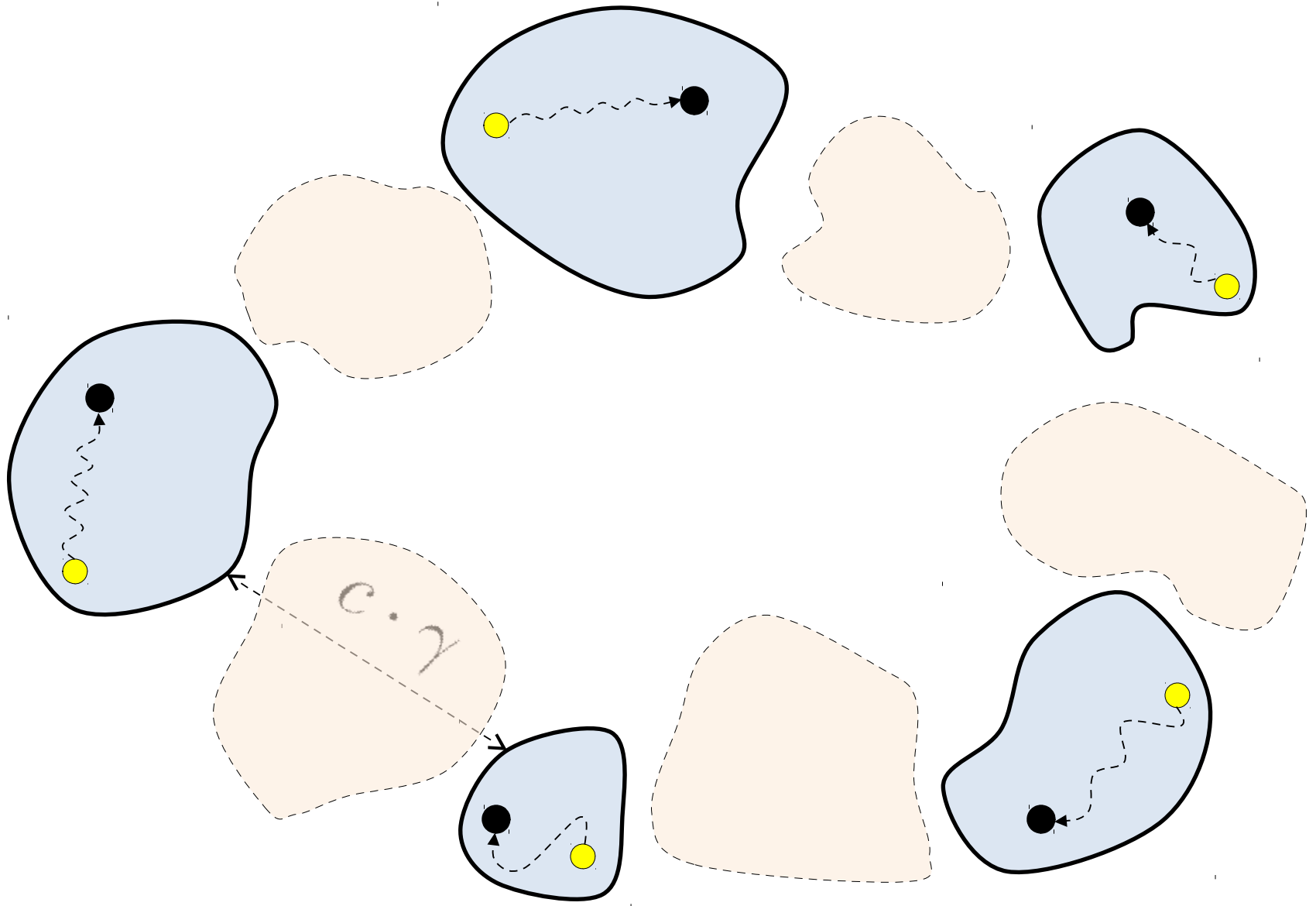
Benefit of Low-Color Partitions



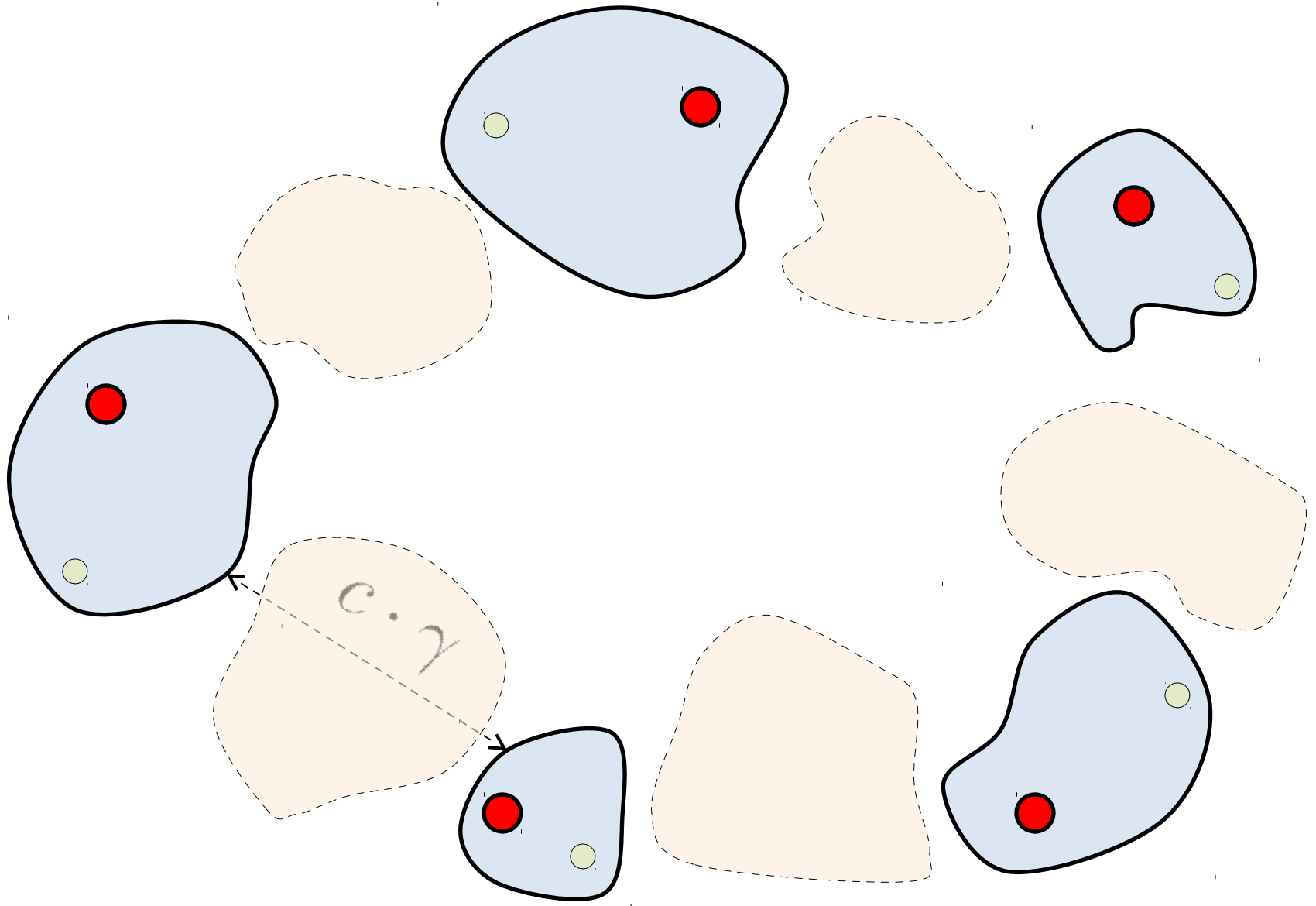
Benefit of Low-Color Partitions



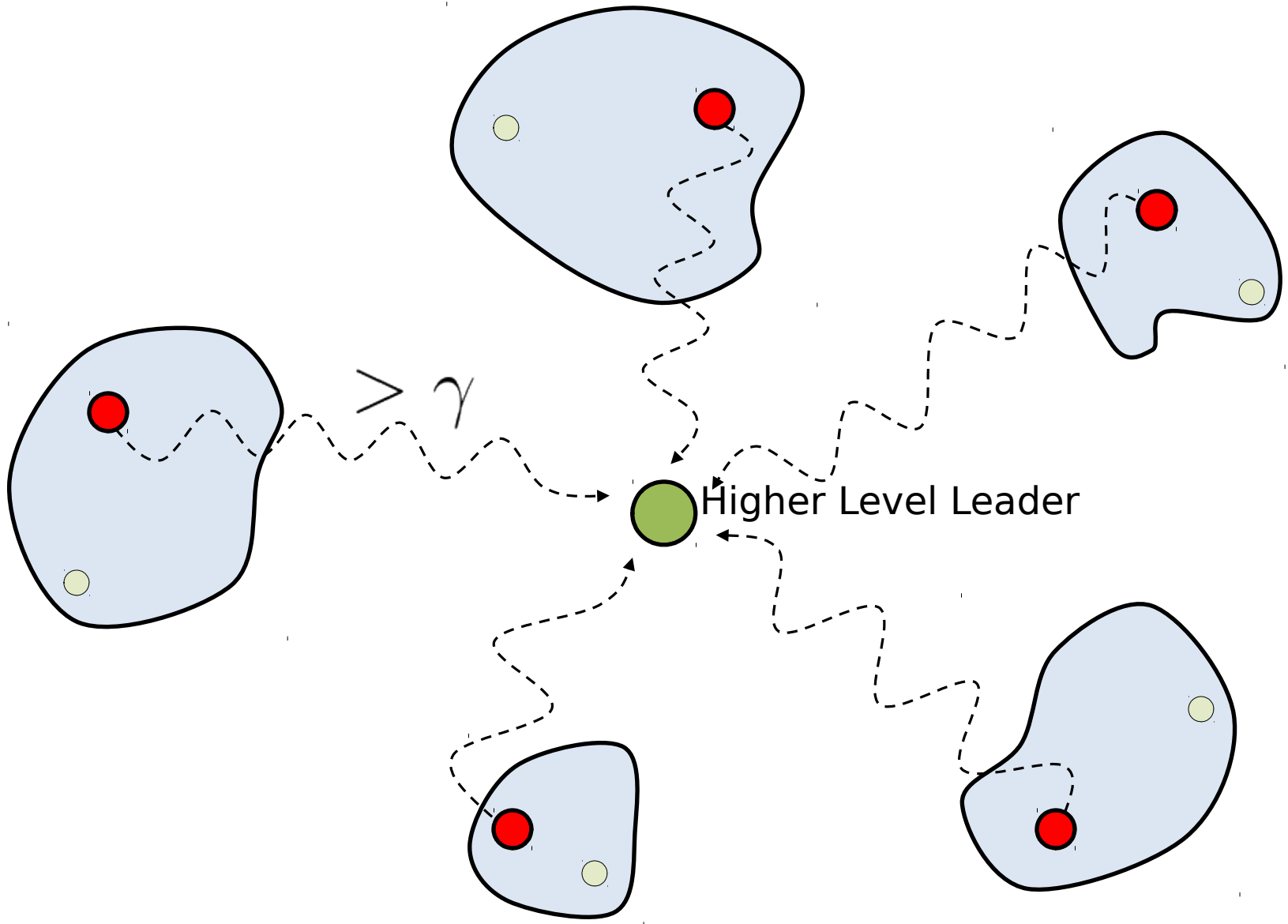
Benefit of Low-Color Partitions



Benefit of Low-Color Partitions



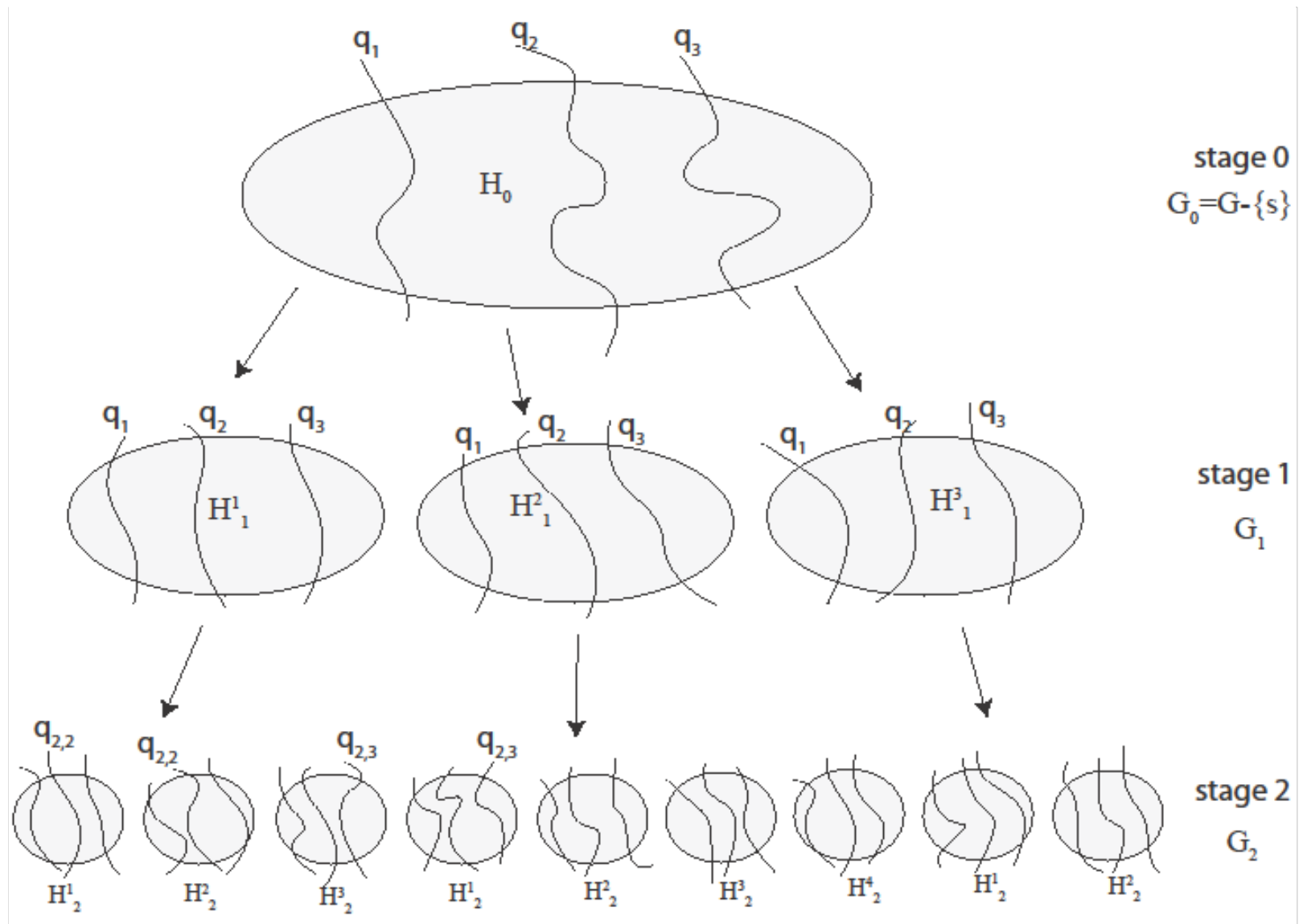
Benefit of Low-Color Partitions



Path Separators

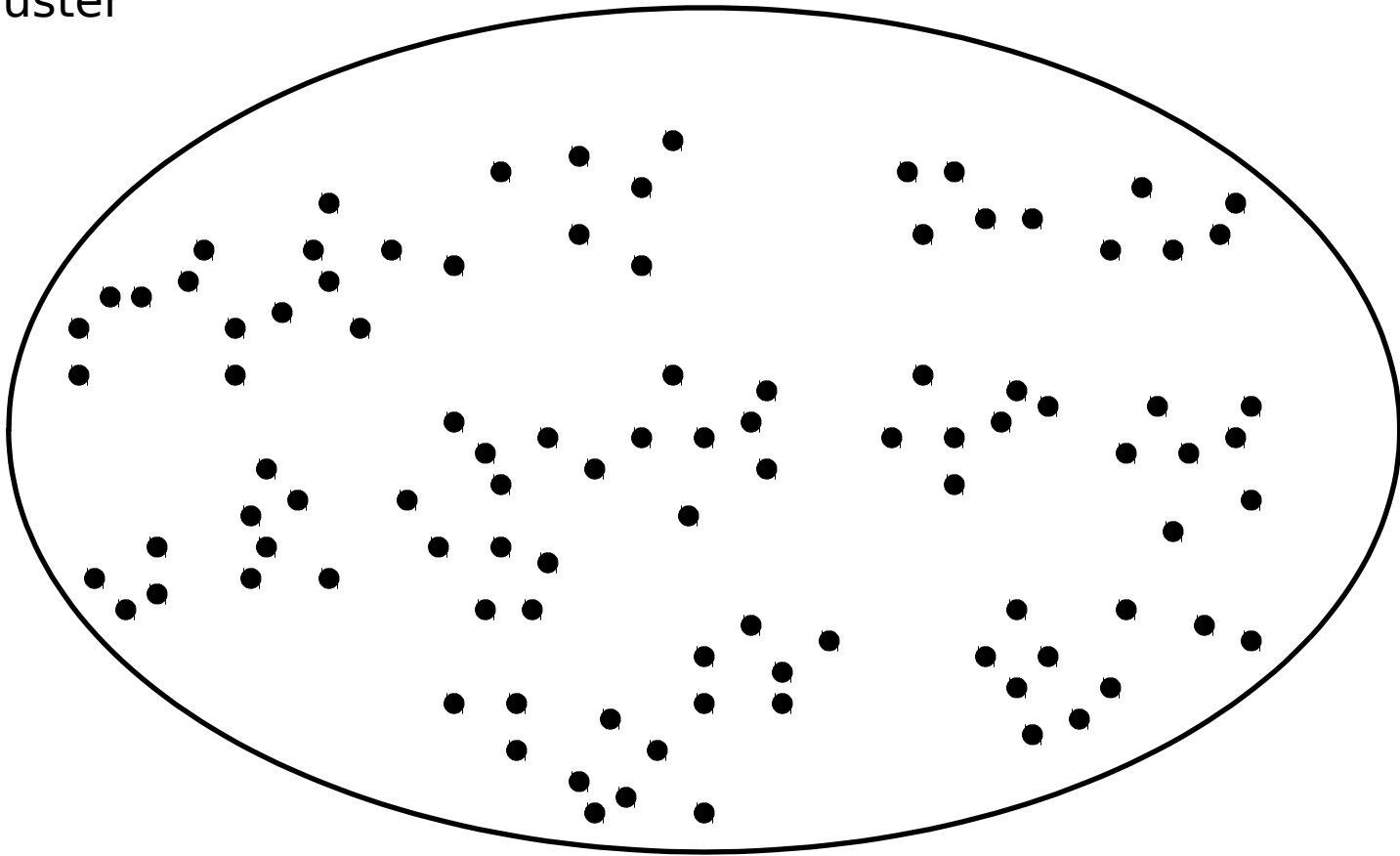
- A set of shortest paths that partition a graph into two or more components of size at most $n/2$ (n is total size of the graph).
- Path Separators can be computed in polynomial time
 - Planar Graphs are 3-path separable
 - H-Minor Free Graphs are k -path separable

Graph Decomposition (Planar Graph)



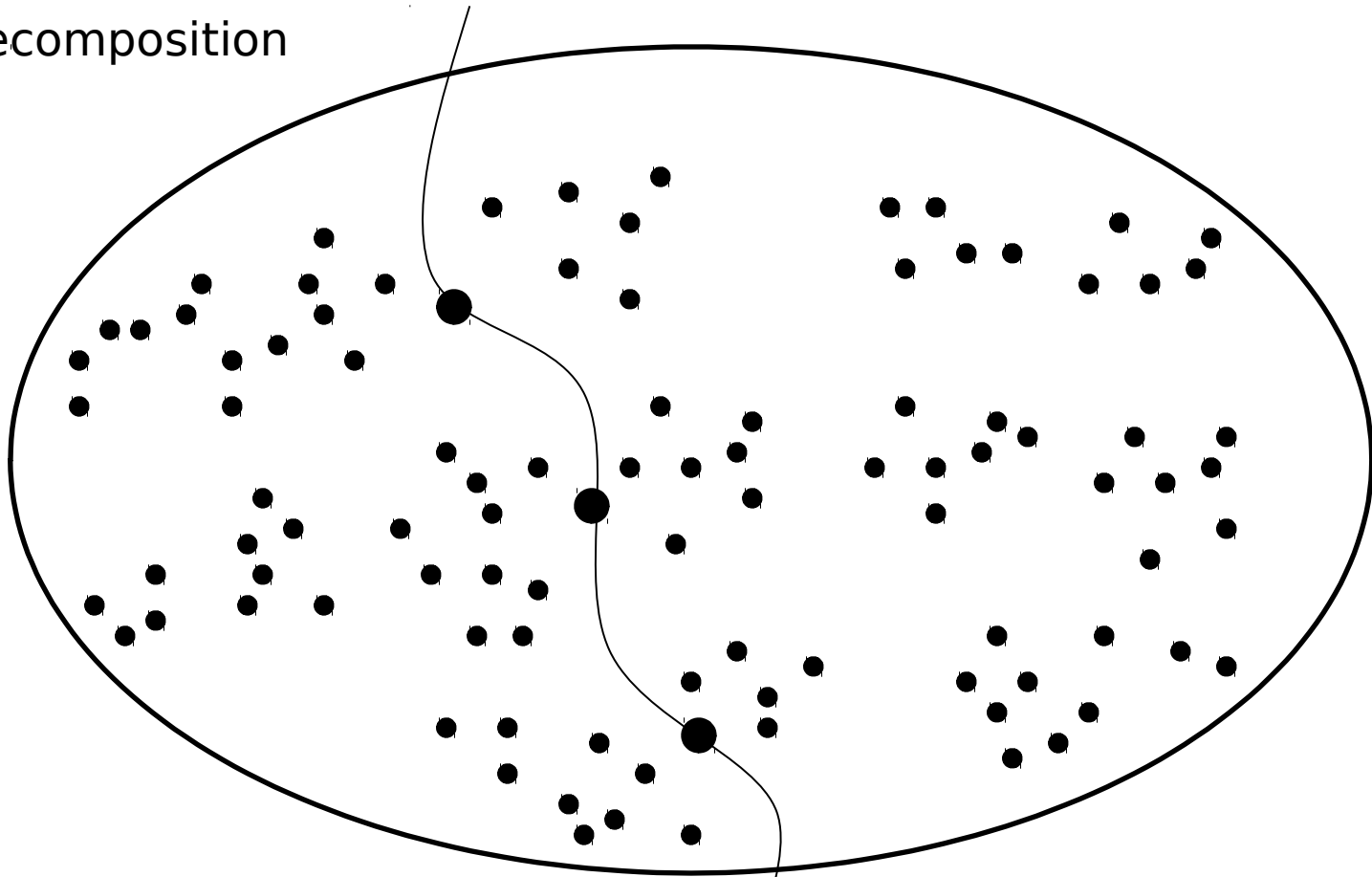
Graph Decomposition

Level 1 Cluster



Graph Decomposition

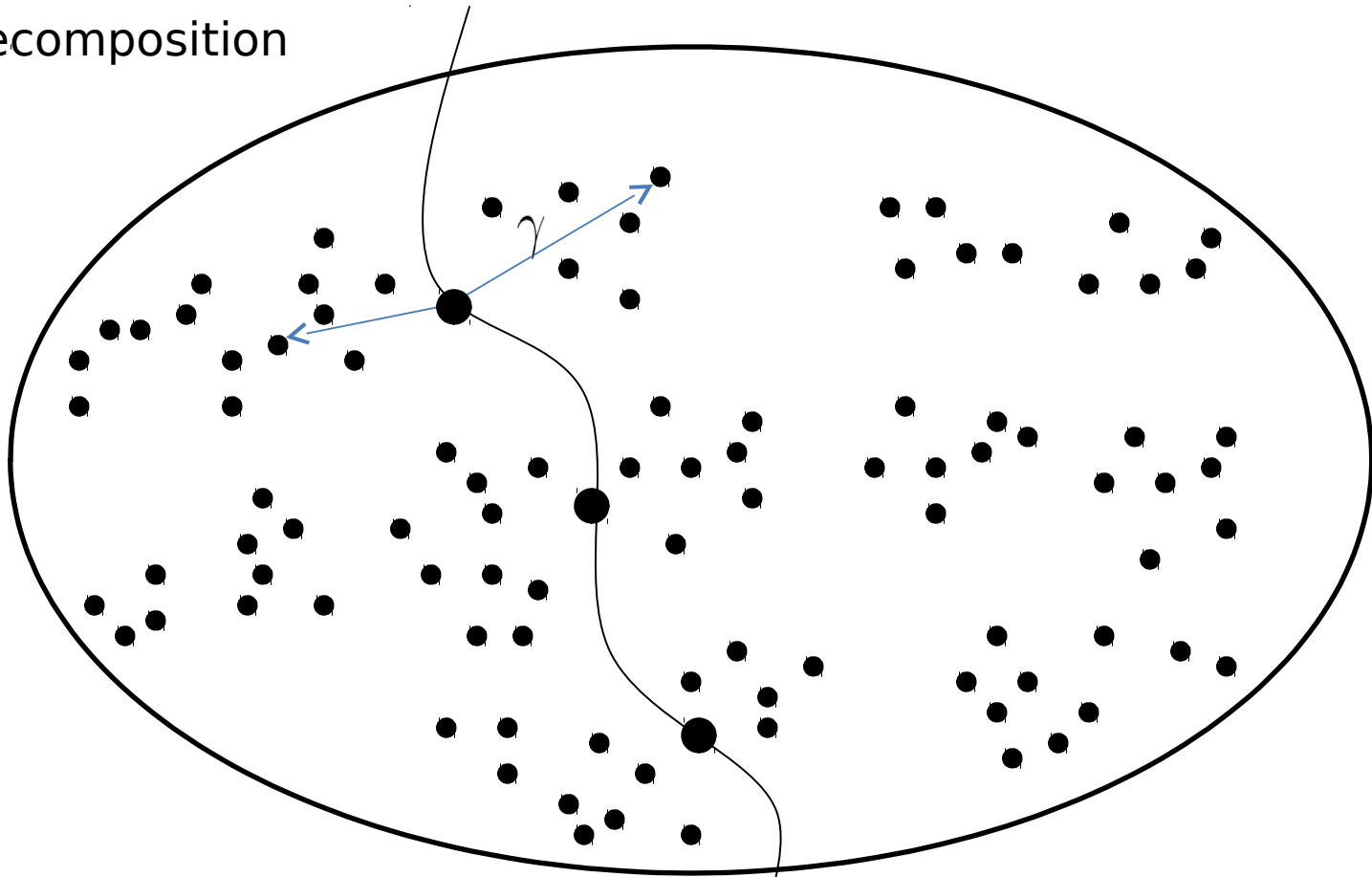
Level 1 Decomposition



Length (Pi) γ
= C.

Graph Decomposition

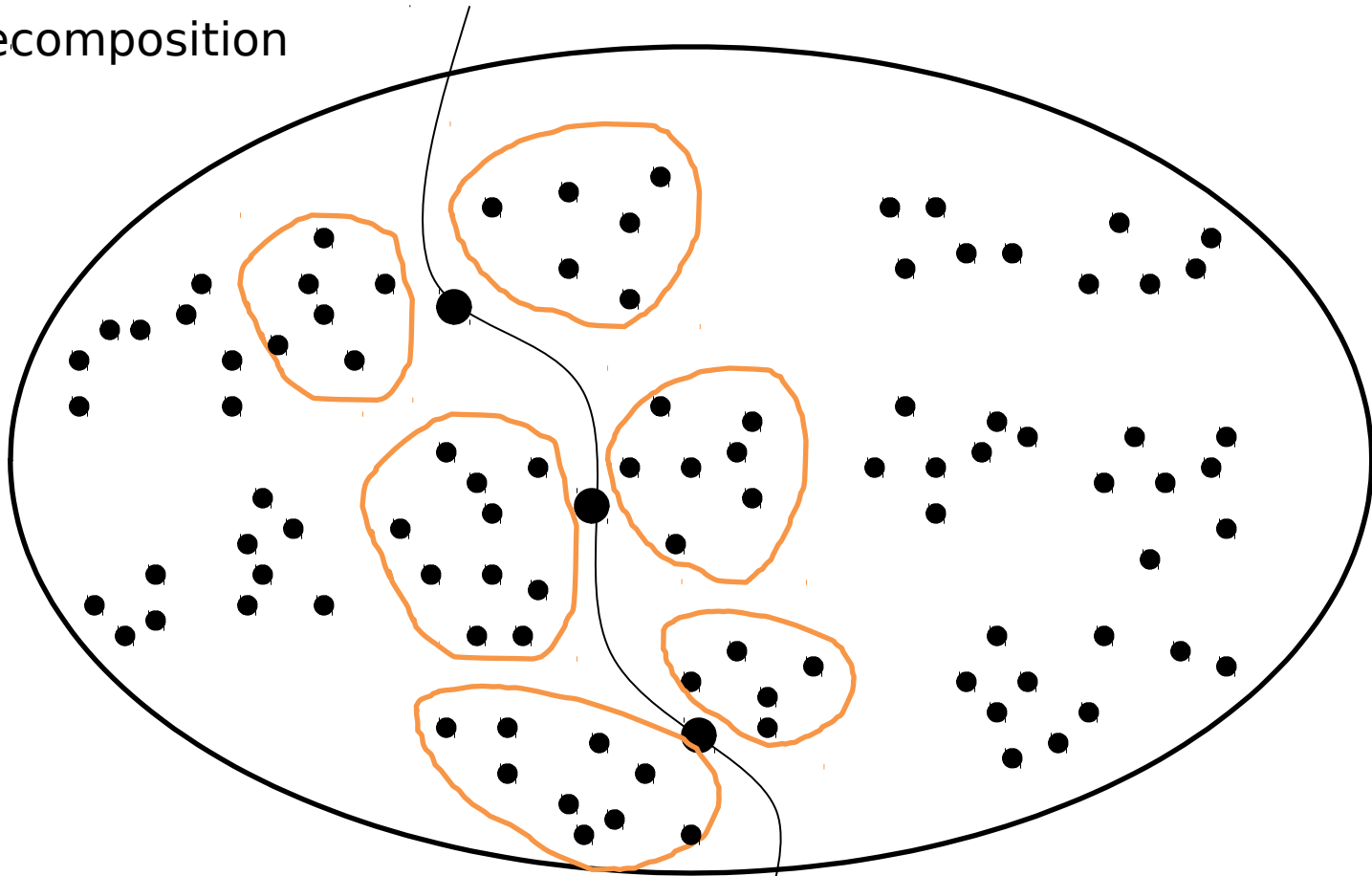
Level 1 Decomposition



Length (Pi) γ
= C.

Graph Decomposition

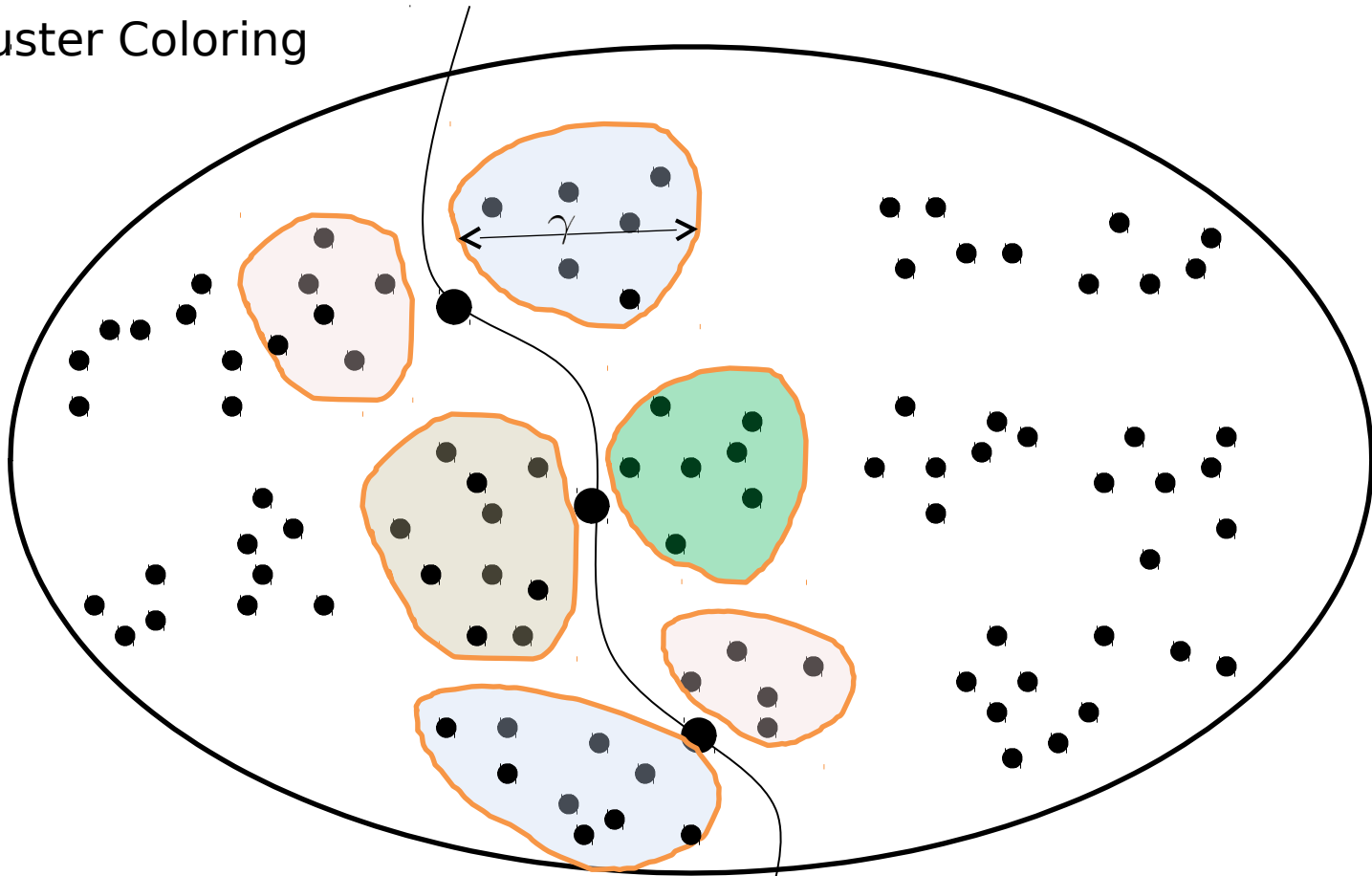
Level 1 Decomposition



Length (Pi) γ
= C.

Graph Decomposition

Level 1 Cluster Coloring

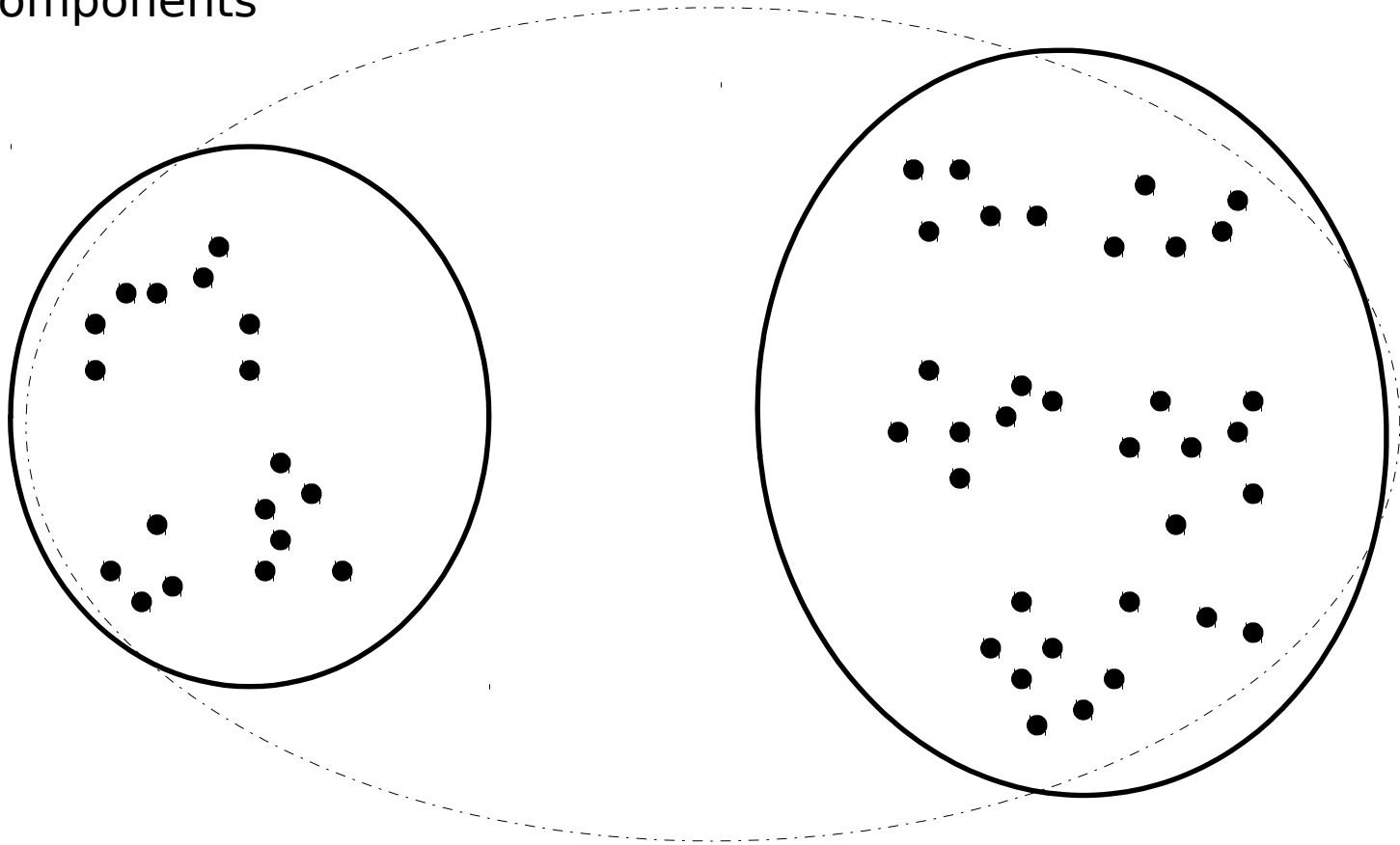


NOTE: Number of such clusters is small

Length $(P_i) \gamma$
= C .

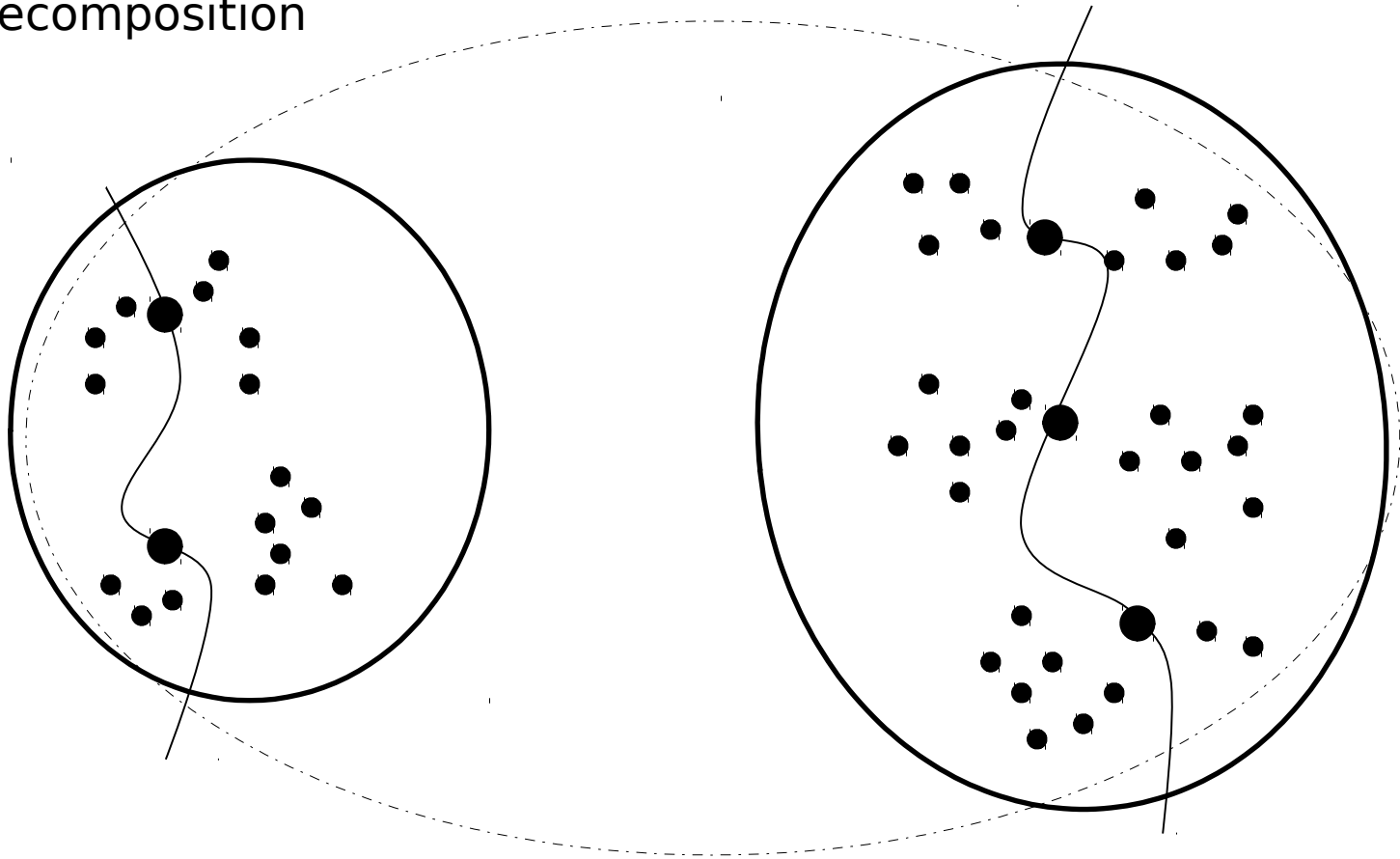
Graph Decomposition

Level 2 Components



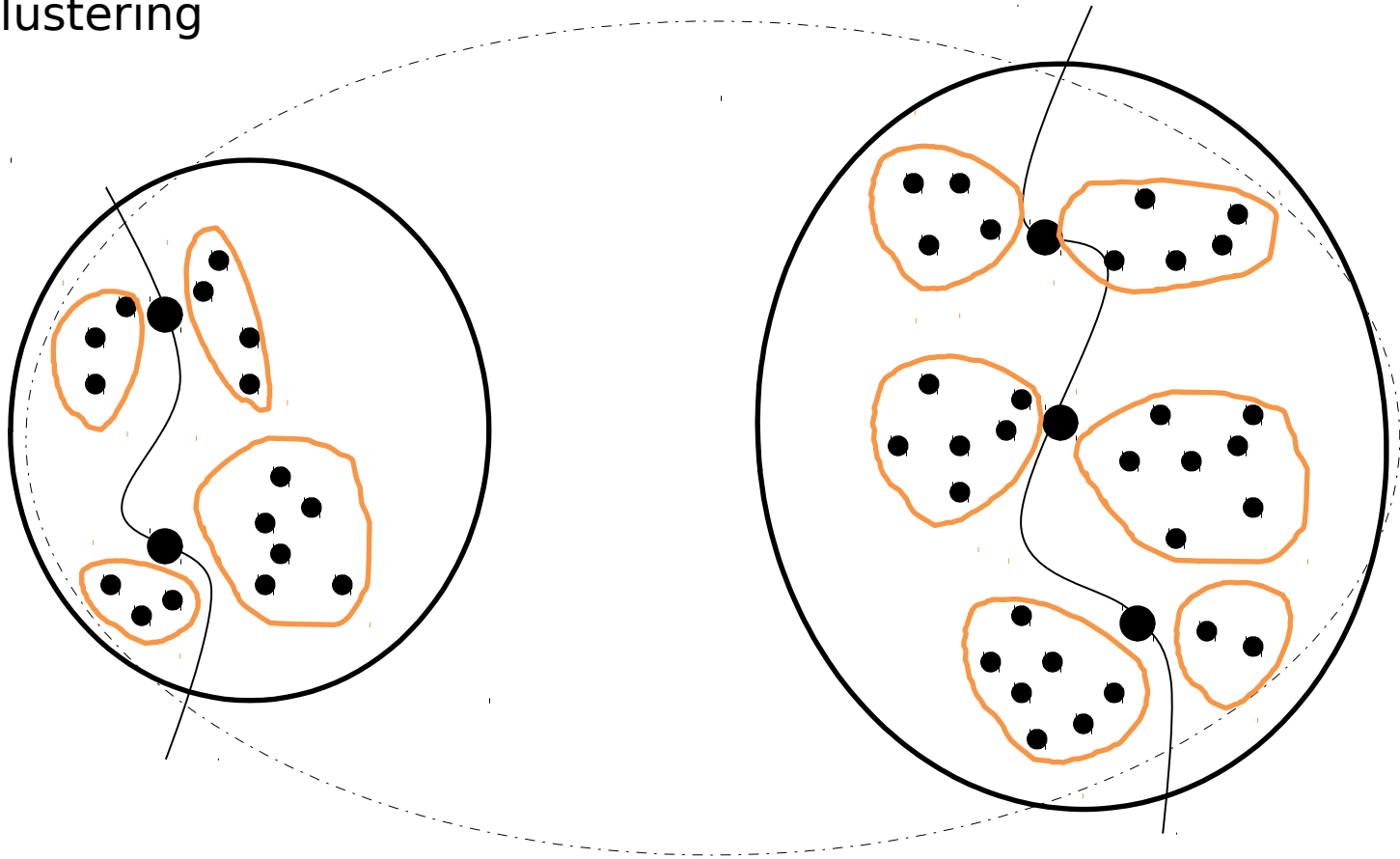
Graph Decomposition

Level 2 Decomposition



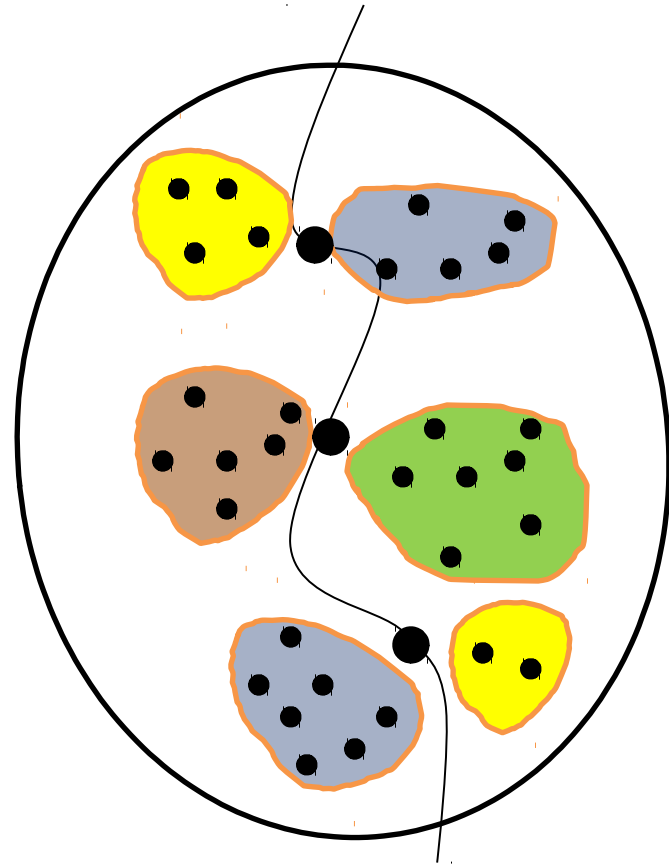
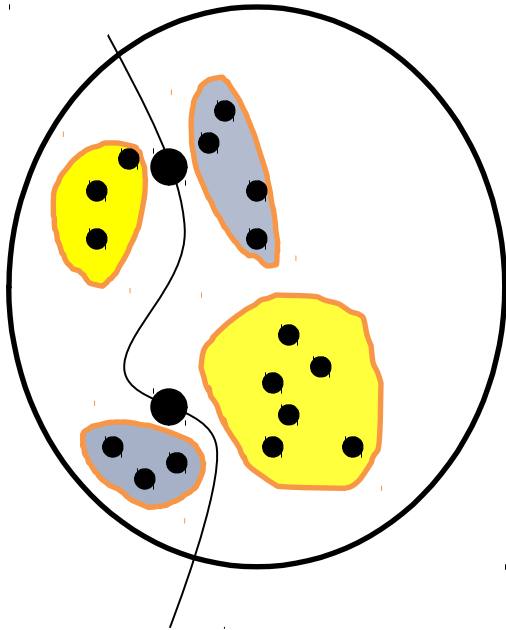
Graph Decomposition

Level 2 Clustering



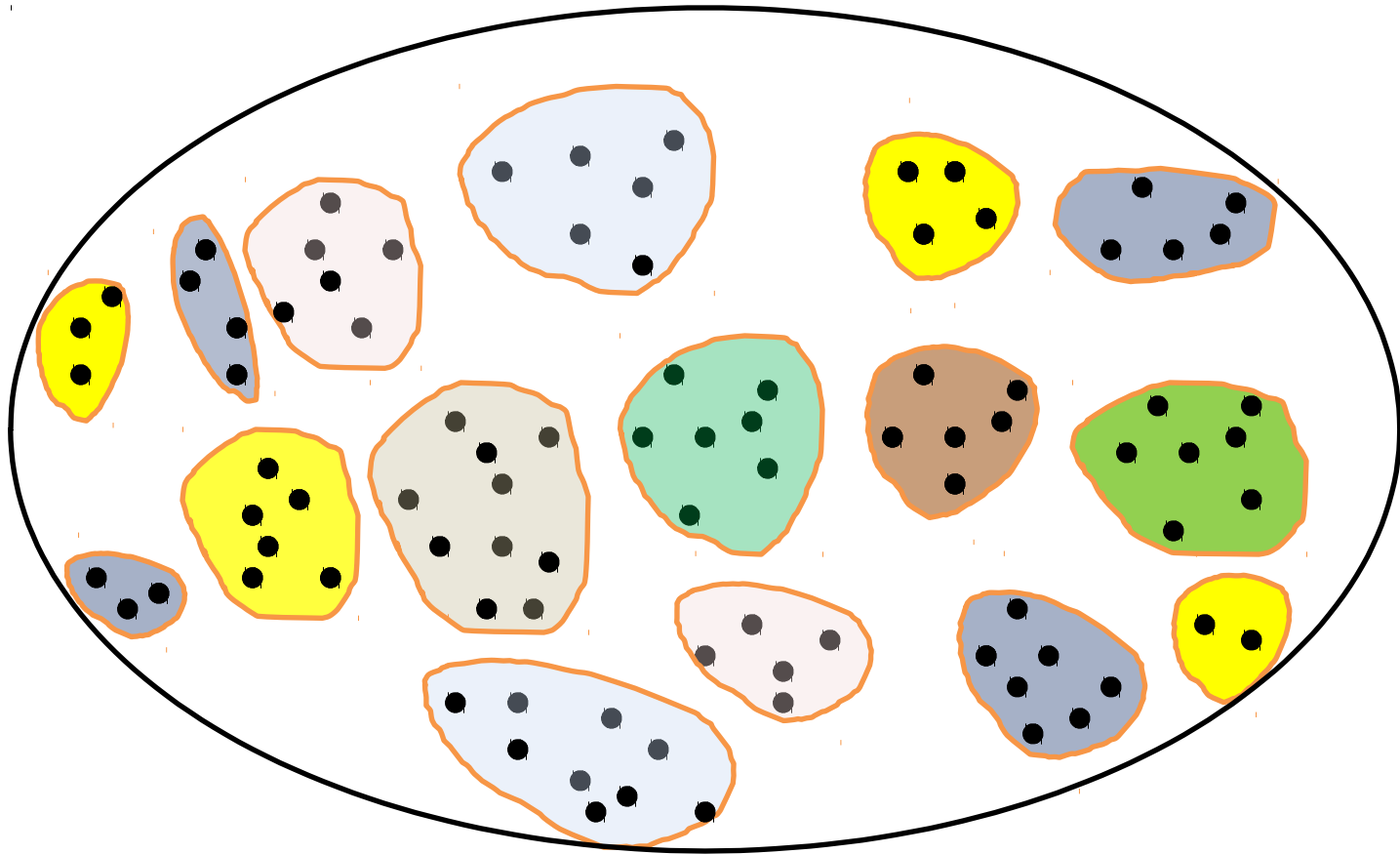
Graph Decomposition

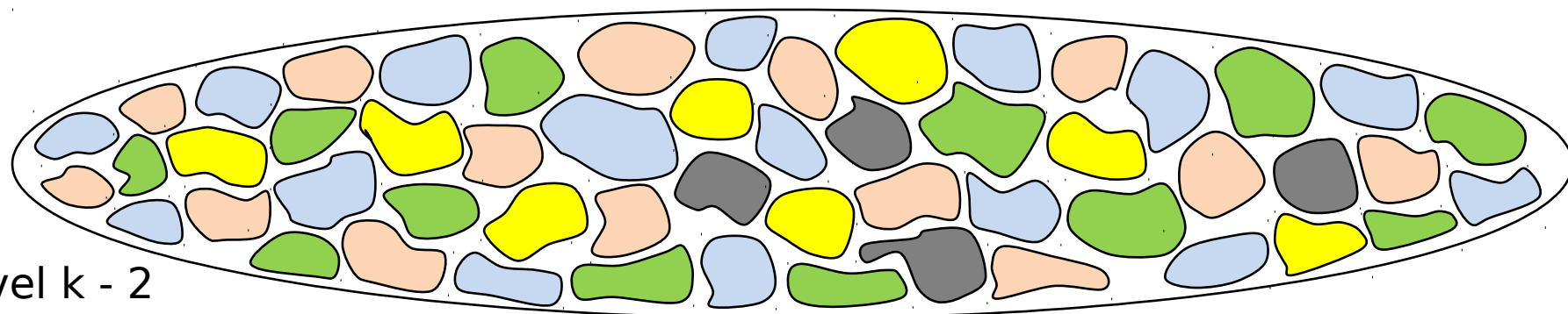
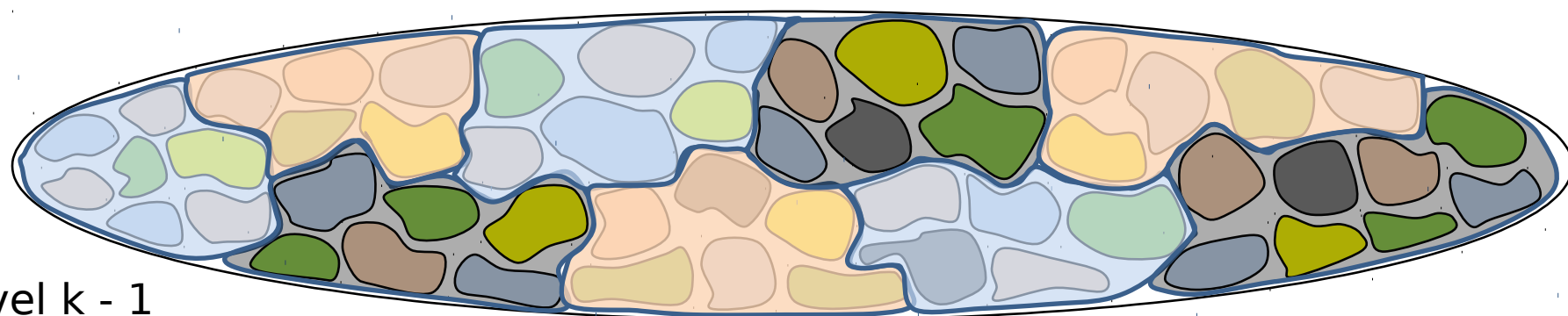
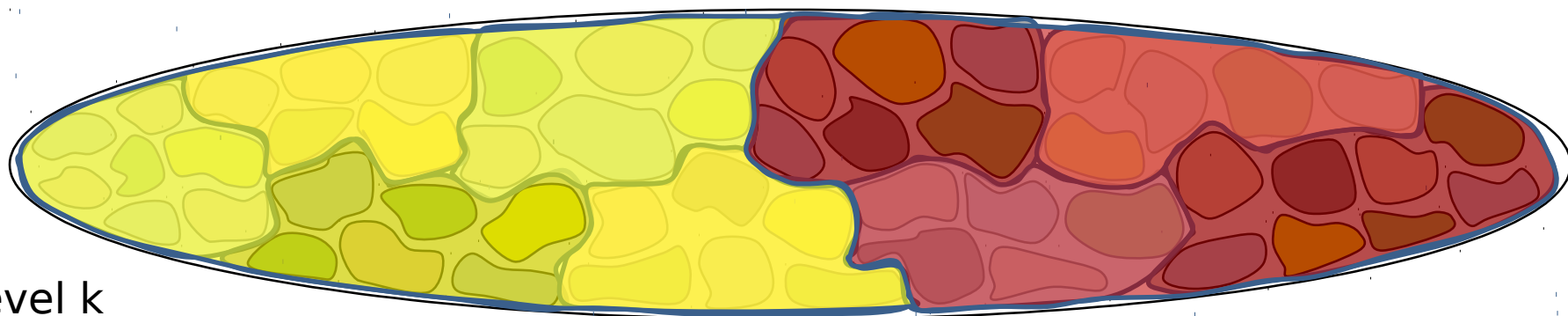
Level 2 -Cluster Coloring



Graph Decomposition

Over Coloring of Clusters (upto level 2)





Contributions

[Arbitrary Graph]

3) Oblivious Network Design that is independent of the sources and aggregation cost function. We provide a deterministic algorithm for constructing an overlay tree.

- On arbitrary/general graphs (no notion of distance/triangle-inequality).
- Provides a $O(\log^3 n)$ -approximation to optimal aggregation cost (measured as total involved aggregation cost).
- **Currently investigating a spanning tree construction.**

Arbitrary Graph

- Use of Sparse Partitions
 - Low Diameter Separators
 - Baruch Awerbuch and David Peleg, FOCS 1990

$$O(\log^3 n)$$

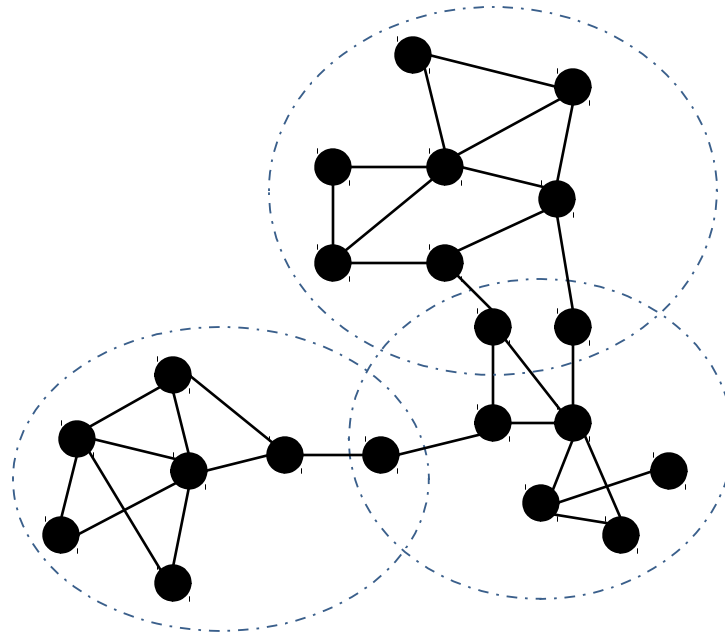
- Competitive Ratio of

Sparse Neighborhood Covers

- Set of sparse neighborhood covers
 - 2^i -neighborhood cover $0 \leq i \leq \log n$
- For each node:
 - For any r , the r -zone is contained within a cluster of diameter $O(r \log n)$
 - The node is in $O(\log^2 n)$ clusters
- Applications:
 - Tracking mobile users
 - Distributed directories for replicated objects

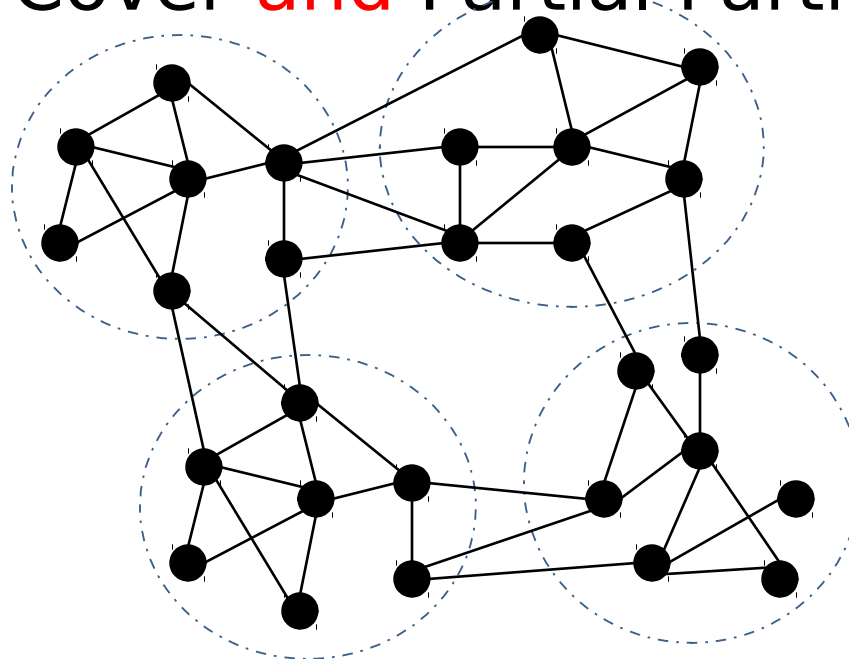
Clusters, Covers and Partitions

- Cluster = Connected subset of vertices $S \subseteq V$
- Cover of $G = (V, E, w)$ = Collection of clusters containing all vertices of G
 $S = \{S_1, S_2, \dots, S_m\}$
 (i.e. $\bigcup_1^m S = V$).



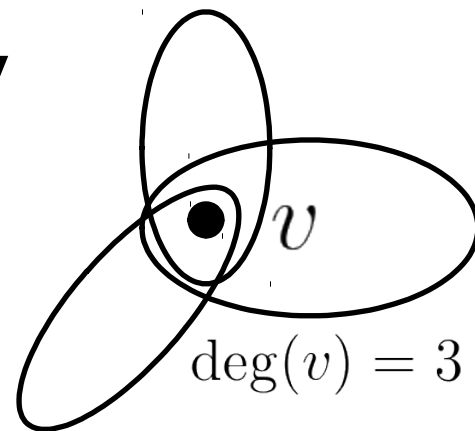
Partitions

- Partial Partition of G = Collection of
 $S = \{S_1, S_2, \dots, S_m\}$ disjoint clusters, $S \cap S' = \phi$
 where
- Partition = Cover **and** Partial Partition.



Measure of Sparsity

The measure is amount of overlap.



$\deg(v, \mathcal{S})$ = # of occurrences of v in clusters $S \in \mathcal{S}$
 $(V, \mathcal{S})$

i.e., the in hypergraph

$\Delta^c(\mathcal{S})$ $\mathcal{S}$

= Maximum degree of cover

$Avg \Delta^c(\mathcal{S})$ $\mathcal{S}$ $\frac{\sum_{v \in V} \deg(v, \mathcal{S})}{n}$

= $\frac{\sum_{S \in \mathcal{S}} |S|}{n}$ Average Degree of $\mathcal{S}$ = n

=

n

Wrap Up – a comparison chart

Related Work	Algorithm Type	Graph Type	IAF	ISS	Approx Factor	Tree Type
Lujun Jia <i>et al.</i> [14]	Deterministic	Random Deployment	×	✓	$O(\log n)$	One Overlay
Ashish Goel <i>et al.</i> [7]	Randomized	General Graph Δ -inequality	✓	×	$O(\log k)$	One Overlay
Ashish Goel <i>et al.</i> [8]	Probabilistic	General Graph	✓	×	$O(1)$	Multiple Overlay
Gupta <i>et al.</i> [11]	Randomized	General Graph	✓	✓	$O(\log^2 n)$	Multiple Overlay
Srivathsan <i>et al.</i> [17]	Deterministic	Low Doubling Dimension	✓	✓	$O(\log^3 D \cdot \log n)$	One Spanning
Srivathsan <i>et al.</i>	Deterministic	Planar Graph	✓	✓	$O(\log D \cdot \log^2 n)$	One Overlay
Srivathsan <i>et al.</i>	Deterministic	Arbitrary Graph	✓	✓	$O(\log^3 n)$	One Overlay

Future Work

- Real Spanning Trees for
 - Planar Graphs
 - Arbitrary Topologies
- Efficient Trees for
 - Smart Grids
 - VLSI Power Circuitry
 - Data Center Networks
- Dynamic Data Structures
 - When more data-sources emerge
 - Consume outdated data along the path
 - Multiple sinks
 - Fault-Tolerance / Survivable Networks.

Other Interesting Problems

- Multiphase Steiner Trees:** Given an edge-weighted graph B with vertex set X and subsets $X_1, Y_1, \dots, X_m, Y_m$ of X with $X_i \cap Y_i = \emptyset$, the problem is to find a minimum weighed subgraph G such that for every $i = 1, \dots, m$, $G|_{Y_i}$ contains a Steiner tree for X_i without using vertices not in Y_i .
- Multiphase Spanning Trees:** Given an edge-weighted complete graph with vertex set X and subsets $X_1, \dots, X_m$ of vertices, the problem is to find a minimum weighed subgraph G such that for every $i = 1, \dots, m$, $G|_{X_i}$ contains a spanning tree for X_i .

Other Interesting Problems

- **(Rectilinear) Steiner Arborescence:** Given a weighted directed graph G , a vertex r and a subset P of n vertices, a **Steiner arborescence** is a directed tree with root r such that for each $x \in P$ there exists a path from r to x . The shortest Steiner arborescence is also called a **minimum Steiner arborescence**.
 - NP-Hard
 - Best approximation is $O(\log n)$
- Bottleneck Steiner Problems (applications in Optical/DWDM Networks)

Bottleneck Steiner Problems

Wavelength Assignment, Edge Length Control etc.

Input

- A network $G=(V,E)$
- A varying set of demands
 - demand $d_i = (s_i, t_i, R_i)$

Minimize the max edge Length

- WDM fibers
 - U : fiber capacity, number of wavelengths per fiber

Min # of amplifiers

Min # of WDM Colors

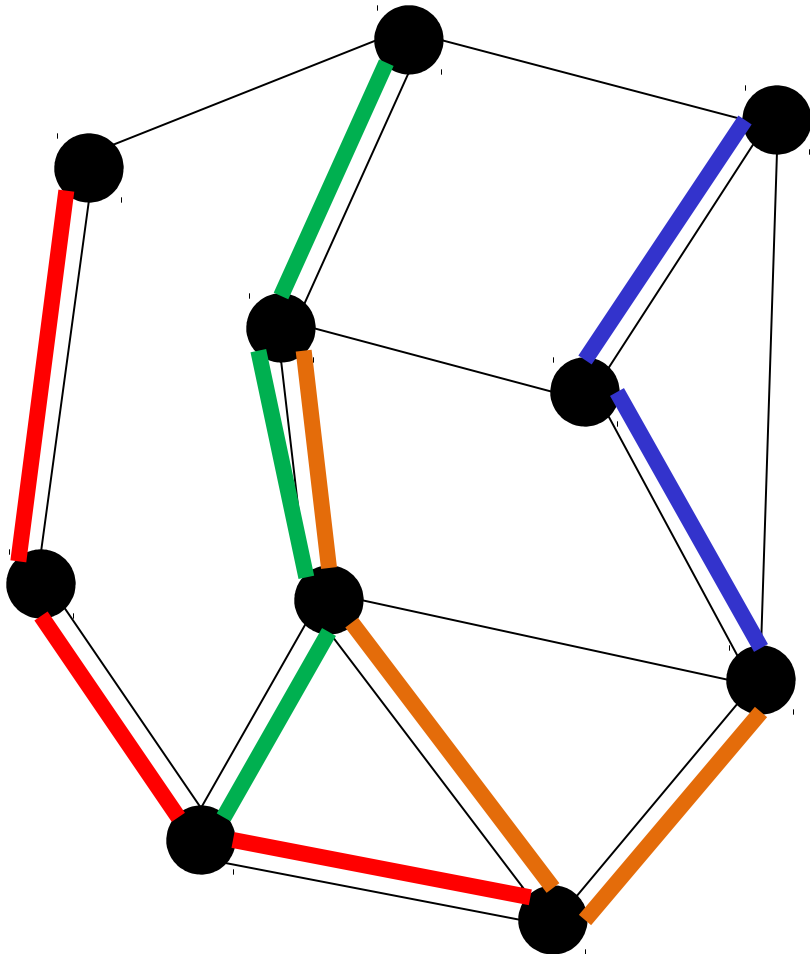
Output

- Assign a wavelength for each demand route
- Demand paths sharing same fiber have distinct wavelengths

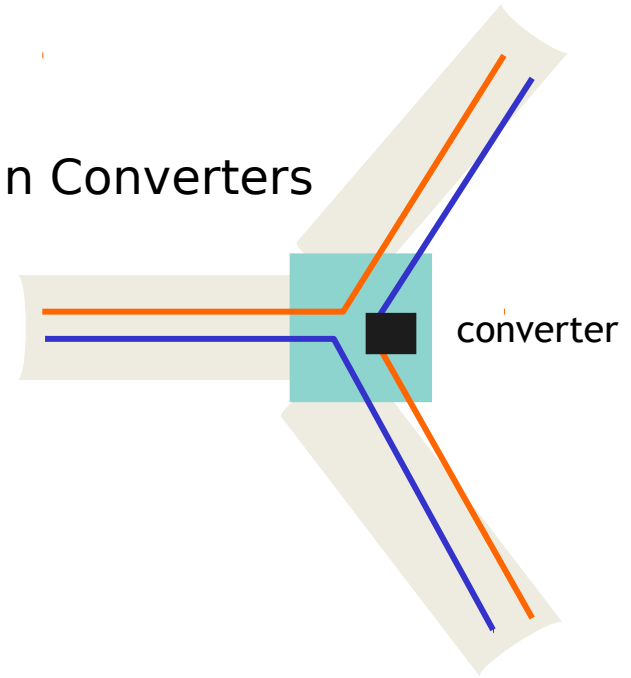
limited transmission power, signals can travel a limited distance for guaranteed transmission.

amplifiers or transceivers at some locations.

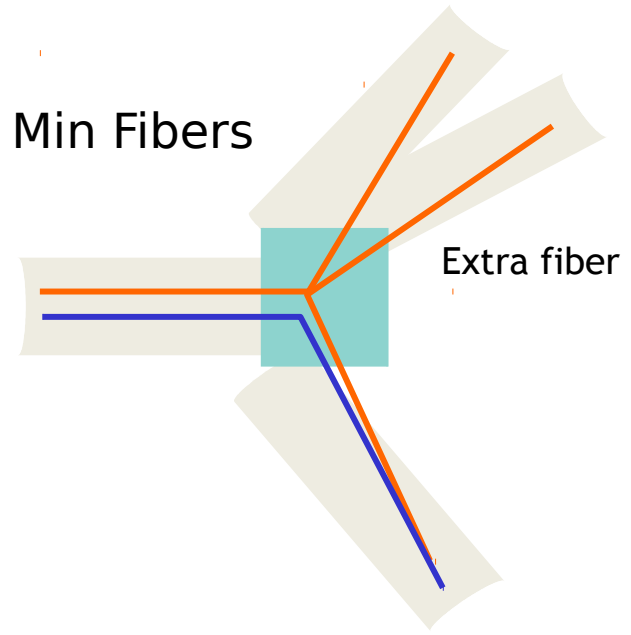
DWDM Networks



Min Converters



Min Fibers



List of Publications

Journal Publications

J2 Srivathsan Srinivasagopalan, Costas Busch, S.S. Iyengar, "Oblivious Network Design for Doubling Metrics", Manuscript in preparation.

Refereed Conference Publications

C7 Srivathsan Srinivasagopalan, Costas Busch, S.S. Iyengar, "An Oblivious Spanning Tree for Buy-at-Bulk Network Design Problems", submitted to 36th International Workshop on Graph Theoretic Concepts in Computer Science, WG 2010, Zaros, Crete, Greece, June 28-30, 2010.

C6 Srivathsan Srinivasagopalan, Costas Busch, S.S. Iyengar, "Universal Data Aggregation Trees for Sensor Networks in Low Doubling Metrics", brief announcement at International Workshop on Algorithmic Aspects of Wireless Sensor Networks, ALGOSENSORS' 09, Rhodes, Greece, July 10-11, 2009. To appear in Lecture Notes in Computer Science (LNCS), Springer-Verlag 2009.

List of Publications

C4 Nageswara Rao, Mallikarjun Shankar, Jren-Chit Chin, David Yau, S. Srivathsan, S.S. Iyengar, Yong Yang, Jennifer Hou, "Identification of Low-Level Point Radiation Sources Using a Sensor Network", ACM/IEEE Conference on Information Processing in Sensor Networks, (IPSN'08), April 22-24, 2008, St. Louis, Missouri.

C3 Suman Kumar, S. Srivathsan, Seung-Jong Park, S.S. Iyengar, "Estimating Data Redundancy in Sensor Networks", Third International Innovations and Real-Time Applications of Distributed Sensor Networks Symposium, November 26-27, 2007, Shreveport, Louisiana.

C2 S. Srivathsan, S.S. Iyengar, "Minimizing Latency in Wireless Sensor Networks - A Survey", Third IASTED Conference on Advances in Computer Science and

Questions?

RSMT Problem

- Rectilinear Steiner minimal tree (RSMT) problem:
 - Given pin positions, find a rectilinear Steiner tree with minimum WL
 - NP-complete
- Optimal algorithms:
 - Hwang, Richards, Winter [ADM 92]
 - Warme, Winter, Zachariasen [AST 00] *GeoSteiner package*
- Near-optimal algorithms:
 - Griffith et al. [TCAD 94] *Batched 1-Steiner heuristic (B1S)*
 - Mandoiu, Vazirani, Ganley [ICCAD-99]
- Low-complexity algorithms:
 - Borah, Owens, Irwin [TCAD 94] *Edge-based heuristic, $O(n \log n)$*
 - Zhou [ISPD 03] *Spanning graph based, $O(n \log n)$*
- Algorithms targeting low-degree nets (VLSI applications):
 - Soukup [Proc. IEEE 81] *Single Trunk Steiner Tree (STST)*
 - Chen et al. [SLIP 02] *Refined Single Trunk Tree (RST-T)*

Minimum Spanning Trees

- The basic algorithm [[Gallagher-Humblet-Spira 83](#)]
 – $O(m + n \log n)$ messages and $O(n \log m)$
- Improved time and/or message complexity [[Chin-Ting 85](#), [Gafni 86](#), [Awerbuch 87](#)]
- First sub-linear time algorithm [[Garay-Kutten-Peleg 93](#)]:
 $O(D + n^{0.91} \log^* n)$
- Improved to $O(D + \sqrt{n} \log^* n)$
- Taxonomy and experimental analysis [[Faloutsos-Molle 96](#)]
- $\Omega(D + \sqrt{n} / \log n)$ lower bound [[Rabinovich-Peleg 00](#)]

Steiner Tree Approximations

- Gabriel Robins and Alexander Zelikovsky: [J. Discrete Mathematics, 2005]
 - 1.55 approximation polynomial-time heuristic.
 - 1.28 approximation for quasi-bipartite graphs.
- Hougardy and Prommel : [SODA 1999] – 1.59 approximation
- Unless $P = NP$, the Steiner Tree Problem for general graphs cannot be approximated within a factor of $1 + \varepsilon$ for sufficiently small $\varepsilon > 0$.
- Rajagopalan and Vazirani [SODA 1999] : Approximation > 1.5
 - Primal-Dual Algorithm
- Zelikovsky [Algorithmica1993]): $11/6$ approximation