# Large Scale Autonomous Systems

## From Anarchy to Self-Structuring

Kermarrec, Mostefaoui, Raynal, Tredan and Viana

presented by Michel RAYNAL

IRISA, Université de Rennes, France

raynal@irisa.fr

# Summary

- Motivation
- Virtual coordinate system:
  - ⋆ Definition, Properties, Construction
- Geometric structuring
  - ⋆ From a mathematical function to a structure
- What is a virtual coordinate system?
- Conclusion

# References

“ *Large-scale networked systems:*
*from anarchy to geometric self-structuring* ”

Kermarrec, Mostefaoui, Raynal, Tredan and Viana

- Tech Report #1875, IRISA, Université de Rennes (F)
  19 pages, December 2007
- Submitted to publication

# INITIAL MOTIVATION

# Autonomous systems

- An autonomous system is characterized by the ability to automatically adapt its behavior according to modifications of its environment without requiring external intervention
- This falls in the well-known self-* properties
  - ⋆ Self-healing, self-stabilization, etc.
- Numerous examples:
  - ⋆ Peer-to-Peer systems, Wireless networks
  - ⋆ Sensor-based systems deployed on a large area
  - ⋆ Etc.

# Self-structuring

- Self-structuring represents the abilty of a system to let emerge a specific structure from scratch without requiring external intervention
- A key feature of autonomy
- In sensor networks: self-structuring represents an important requirement for operations such as forwarding, load balancing, leader election energy consumption management, etc.
- Example: Partitioning into several zones for monitoring purposes, or selection of sensors to ensure specific functions (and save energy)

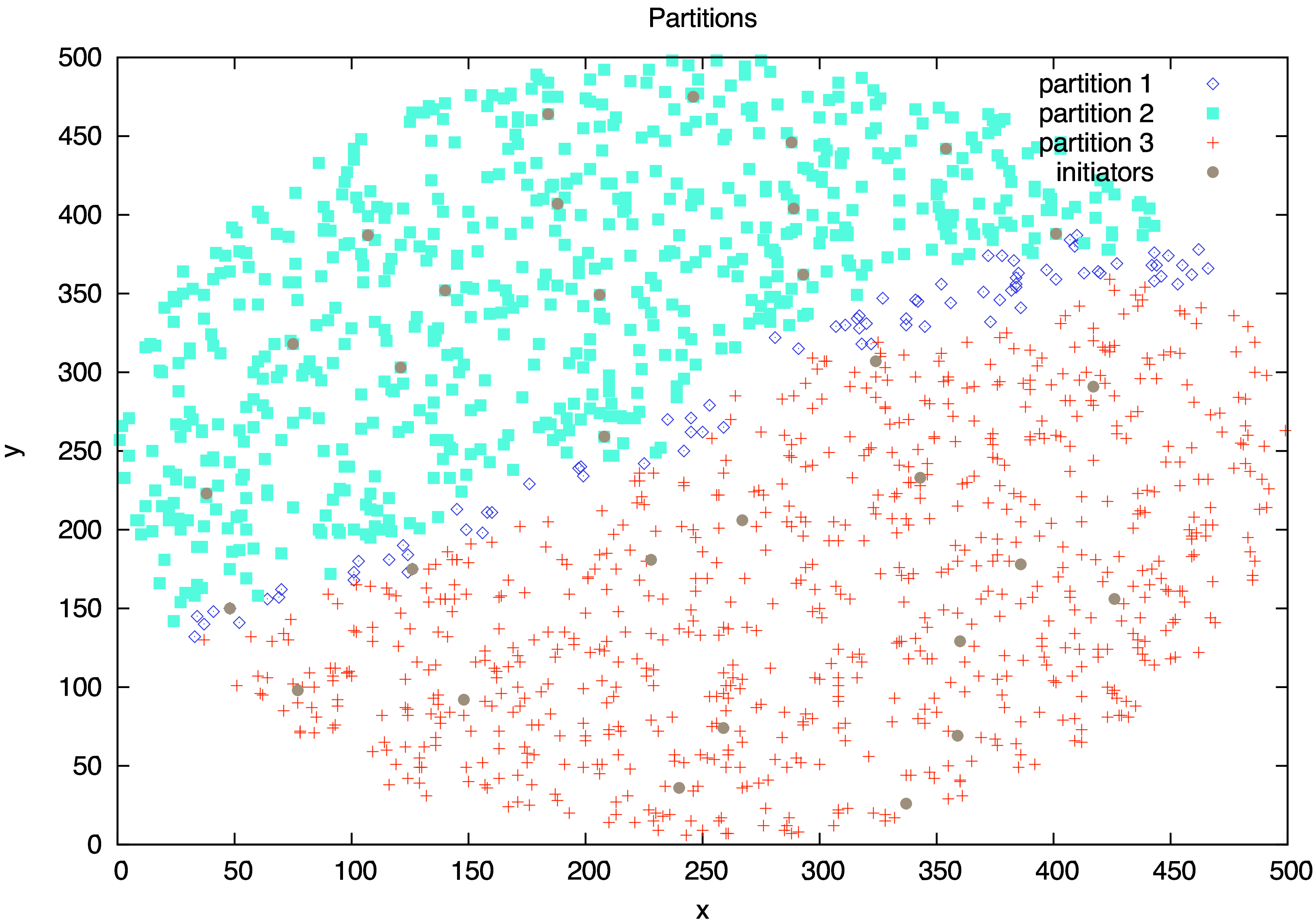
# On the initial knowledge

- Difficulty of self-structuring depends strongly on the amount of knowledge of the network initially known by each entity
- Types of knowledge:
  - ⋆ External knowledge: provided by external devices (e.g., GPS), or global assumptions known by every entity (e.g., size of the network, topology)
  - ⋆ Intrinsic knowledge: gained by computation executed by each entity
  - ⋆ Similar to external clock synchronization vs internal clock synchronization
  - ⋆ The more external knowledge is required, the less robust (autonomous) a system is

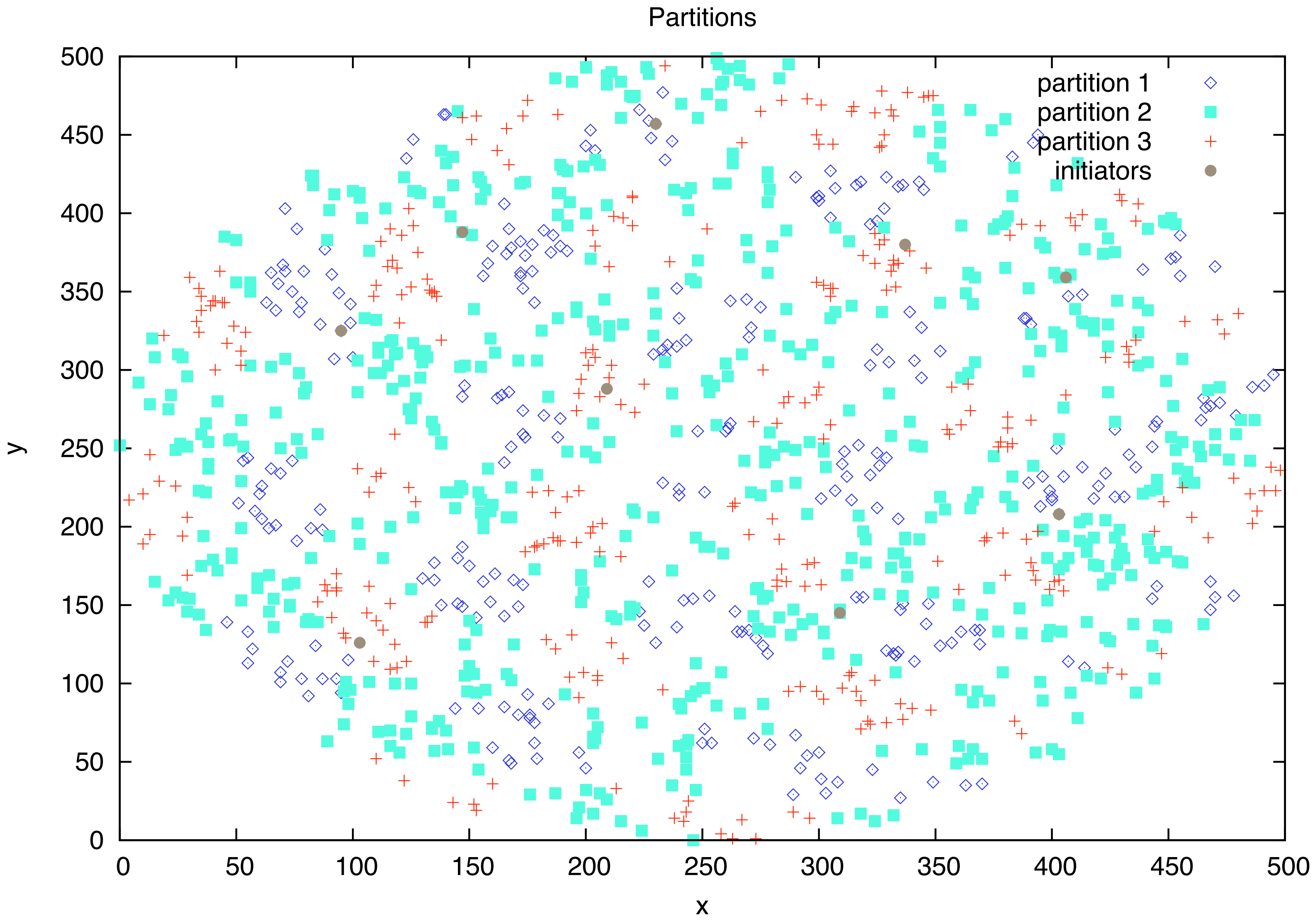
# Network organization: Geometric structuring

- A network organization is based on an underlying structure that is geographical or functional
- Geographical example: sector-shaped clustering
- Functional example: awake and sleep entities

# Example 1: North, South, and Equator zones



# Example 2: Wake up/Sleep entities

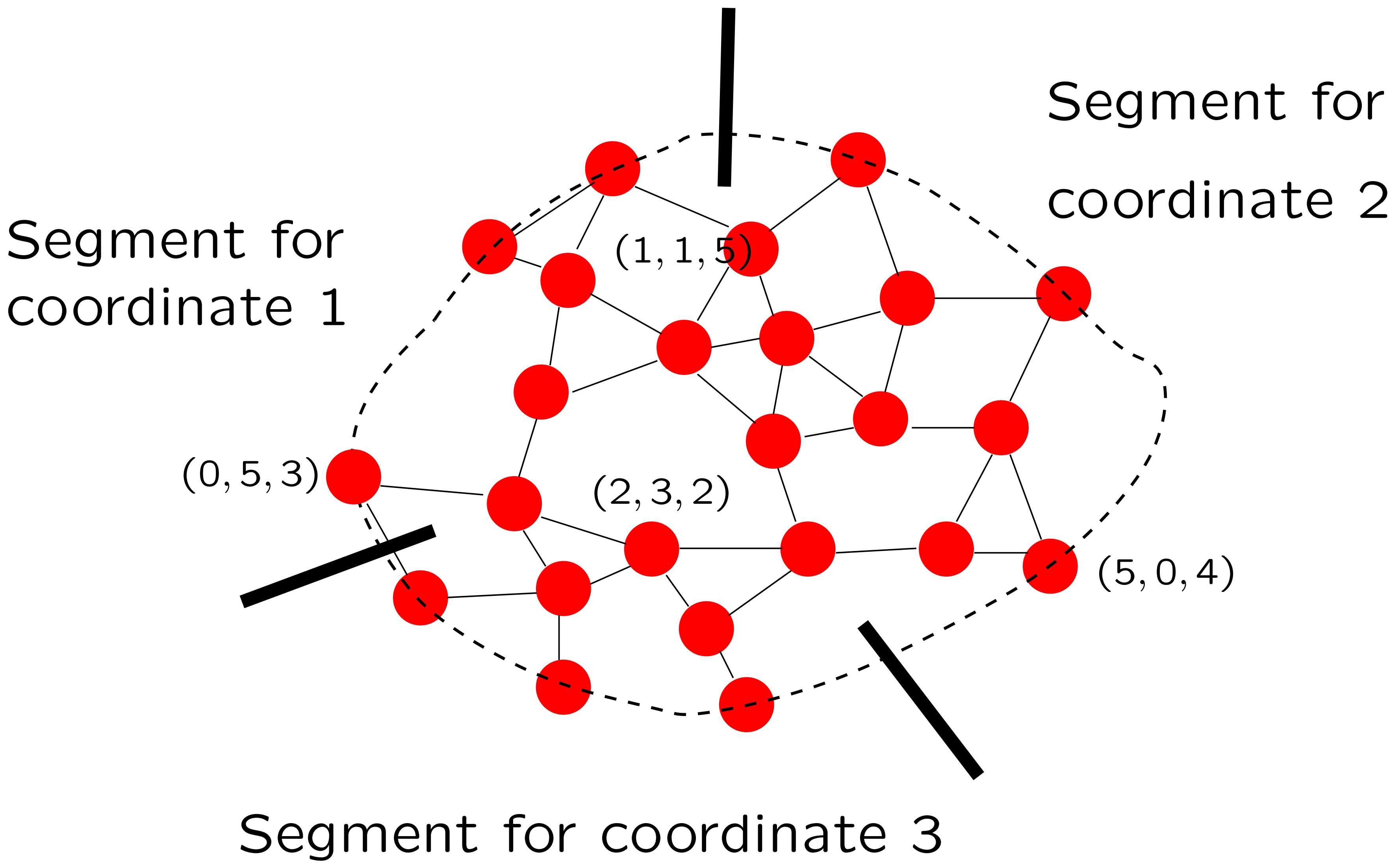


# A VIRTUAL COORDINATE SYSTEM

# Virtual $d$-dimensional space

- The aim of a coordinate system is to provide locations with names (naming system) satisfying some properties
- Let $d$ be the dimension of the coordinate system
- The border of the area is decomposed into $d$ segments
- The virtual coordinates of an entity $x$ is a $d$-uple of integers $\boxed{(x_1, x_2, \ldots, x_d)}$ such that $x_i$ is the projection of $X$ on the $i$th dimension
- More specifically, $x_i$ is the smallest number of hops (projection) from the node $x$ to the border segment $i$

# Virtual coordinates: illustration

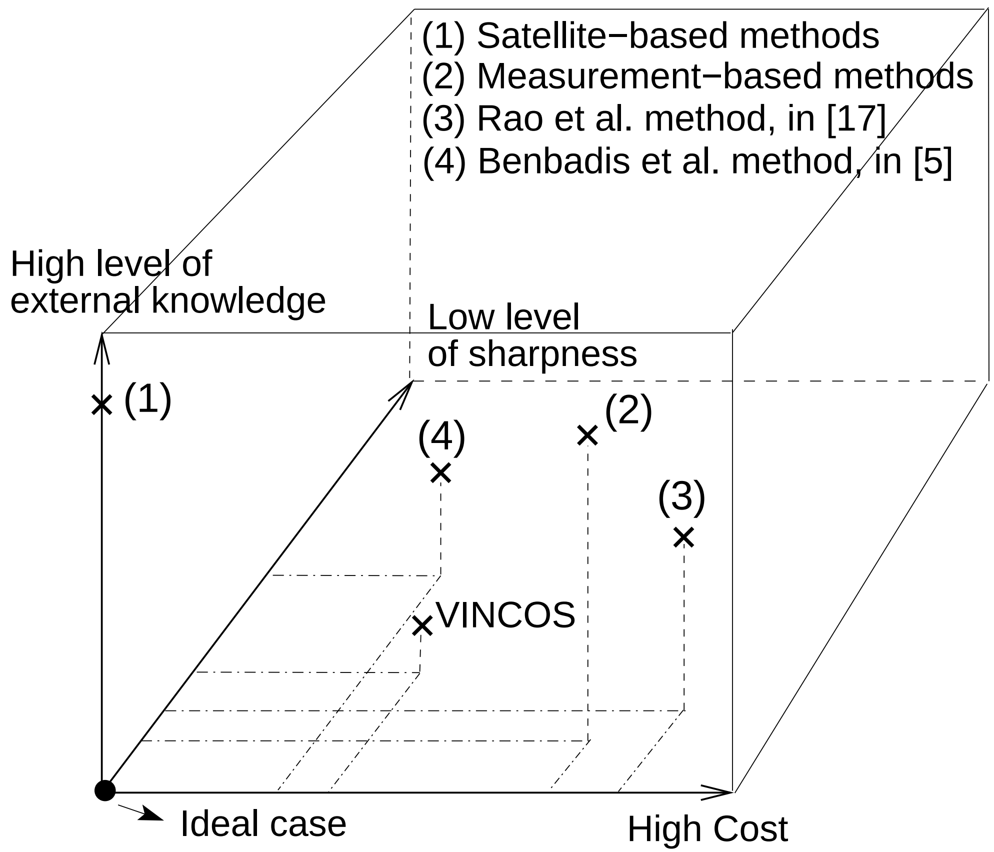


# Underlying rationale

- Not directly related to "real geographic coordinates"
- Connectivity-based approach: the coordinates reflect only the underlying connectivity
- Can adapt to obstacles (mountains, underground, etc.)

# Related work

- GPS-based
- Landmark-based
  Anchor-based
- Hybrid
- Connectivity-based



# Network model: Entities (sensors)

- Each entity has a unique id
- Initial knowledge:
  - ⋆ Each entity knows that all ids are different
  - ⋆ No entity knows the actual nb of entities, the structure of the network, the density distribution, etc.
- Except for their ids, the entities are clone of each other: they are all "equal"
- Each entity has a local clock whose drift is bounded
- No GPS, landmarks, "initially specific" entities, etc.

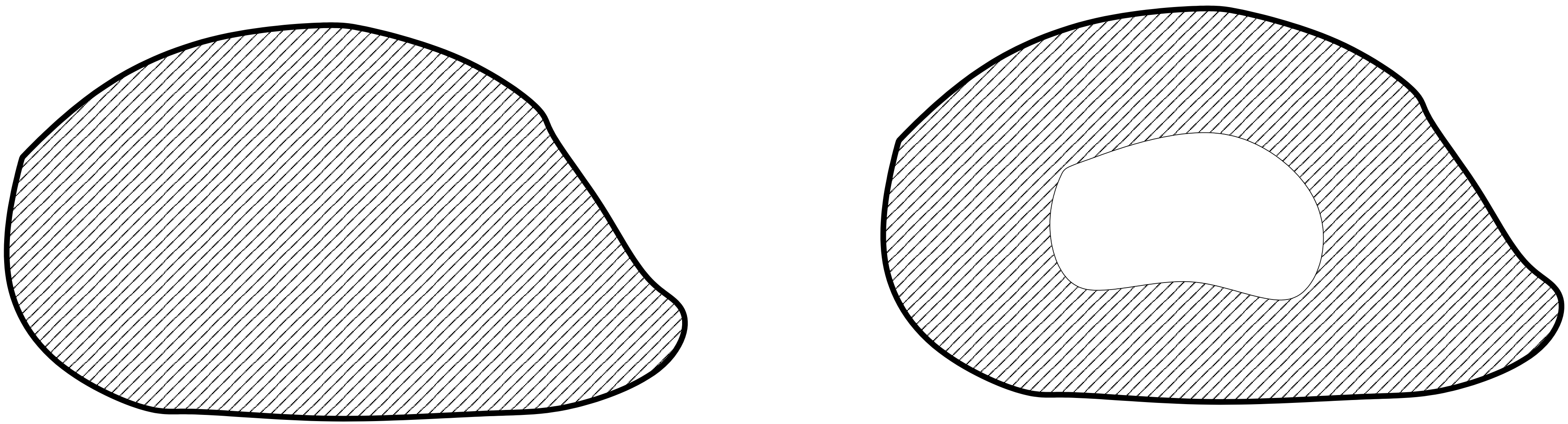
## Network model: Communication

- No node has geographical topology information
- each node has a communication range $R$
- Reachability depends not only on geographical distances, but also on natural obstacles (e.g., valleys separated by mountains)

Unit Disk Graph with obstacles

- The entities within the range of entity $i$ that are directly reachable define the set $com\_neighbors_i$
- Global assumption: the density of entities is such that the network is connected
- There is an upper bound on message transfer delay

# Network model: Network shape



# From anarchy to virtual coordinates

The VC algorithm works in four phases

- Phase #1: Detection of initiators
- Phase #2: Border score definition
- Phase #3: Border-belt construction
- Phase #4: Coordinate computation

Anonymity property: The code executed by entity $x$ with id $ID$ is the same as the code entity $y$ with id $ID$

# Phase #1: Detection of initiators

- An initiator is a "locally maximum" entity from a density point of view
- $i$ is an initiator:

$$\forall\, j \in com\_neigbhors_i : \quad |com\_neigbhors_i| > |com\_neigbhors_j|$$

- The initiators are used to detect border

# Phase #2: Border score definition

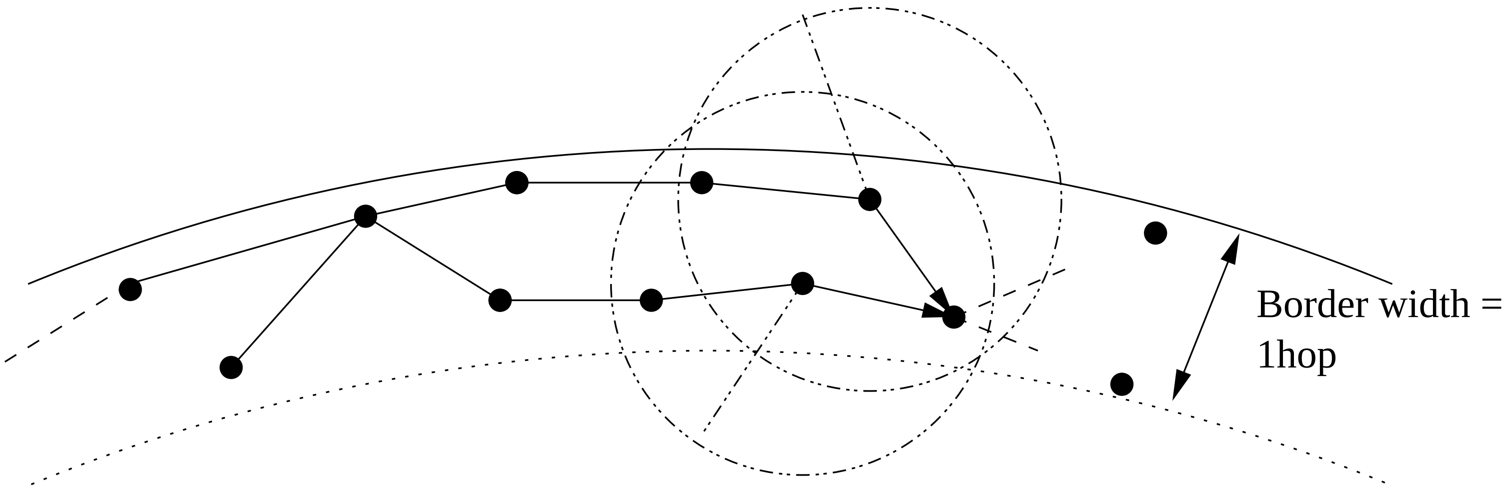
- Every $i$ computes its distance wrt each initiator $j$

$$dist(i,j) = \min\{dist(i,\ell) \mid \ell \in com\_neigbhors_i\} + 1$$

- The border score of an entity $i$ "measures" its average distance to an initiator: $score_i = \sum \; dist(i,j)$
- An entity can compute its score and the scores of its com_neigbhors
- As initiators define "centers" of the system, entities on the border have higher score than the entities that are not on the border
- This allows discovering entities that are (for sure) on the border

# Phase #3: Border-belt construction

- Use of a probe forward-or-discard mechanism (this is the only place where entity ids are used)



# Phase #4: virtual coordinate determination

- At the end of Phase #3, each node on the border knows the number of the segment it belongs to
- On each segment $h$, $1 \leq h \leq d$, each border entity broadcasts a message that is forwarded from entity to entity, thereby allowing each entity to determine the value of its $h$ coordinate

# GEOMETRIC STRUCTURING

# Using VC for network structuring

- Geometric/Functional structuring
- Aim: associate a partition number $p$ with each entity
- Use a mathematical function to define a structure
  - ⋆ Let an entity $i$, with coordinate $c_i$
  - ⋆ The function $f()$

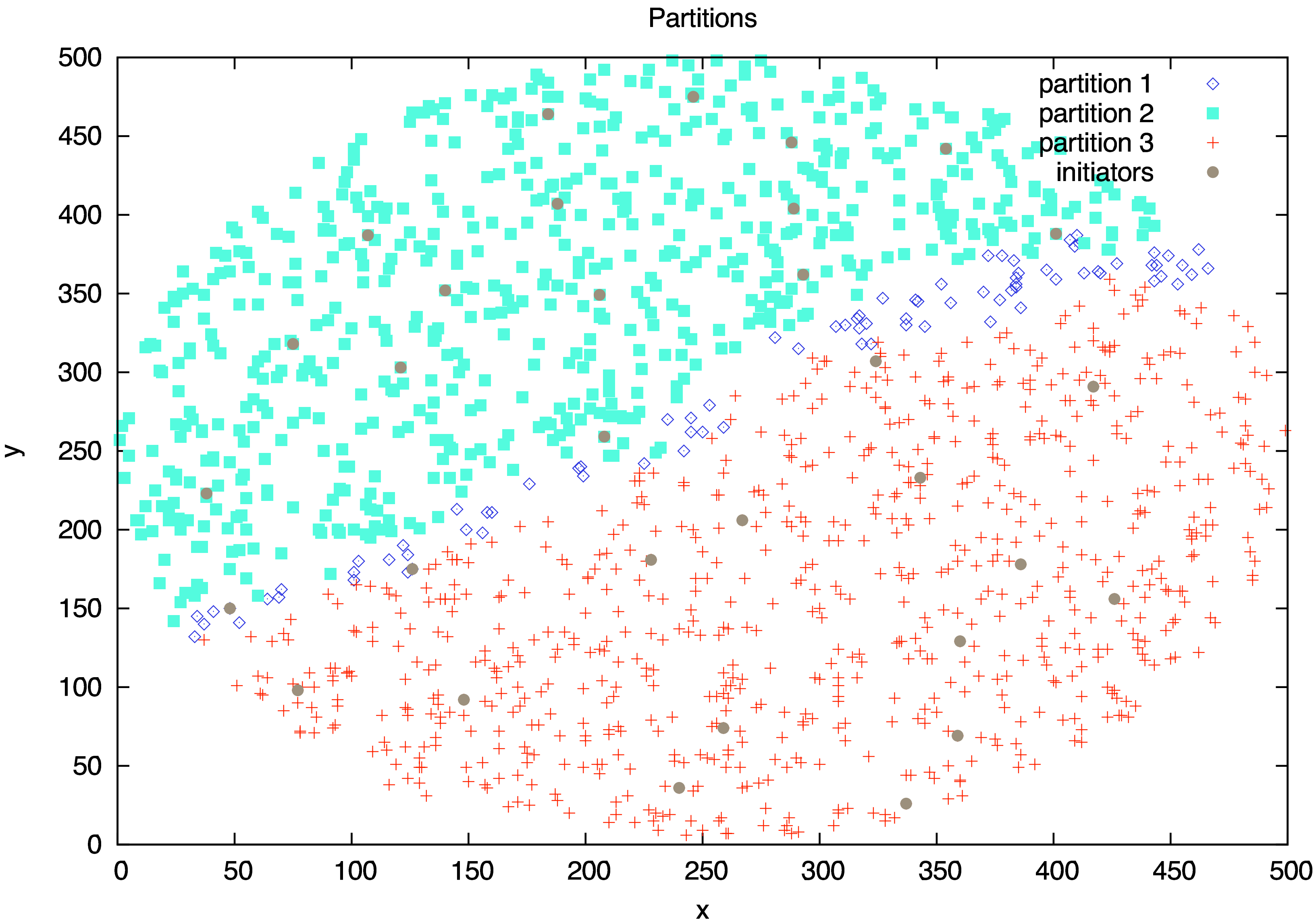
$$
\begin{aligned}
f : \mathcal{K} &\rightarrow \{0, \ldots, p\}, \\
f(c_i) &\rightarrow p_i.
\end{aligned}
$$

# Geometric structuring (1.1)

- North, South and Equator partitioning
- $d = 2$, $p = 3$
- The function $f()$

$$
\begin{array}{rcl}
f : \mathbb{N} * \mathbb{N} & \rightarrow & \{1, 2, 3\} \\
 & & 1 \;\; when \;\; x_1 > x_2 \\
f(x_1, x_2) & \rightarrow & 2 \;\; when \;\; x_1 = x_2 \\
 & & 3 \;\; when \;\; x_1 < x_2
\end{array}
$$

# Geometric structuring (1.2)

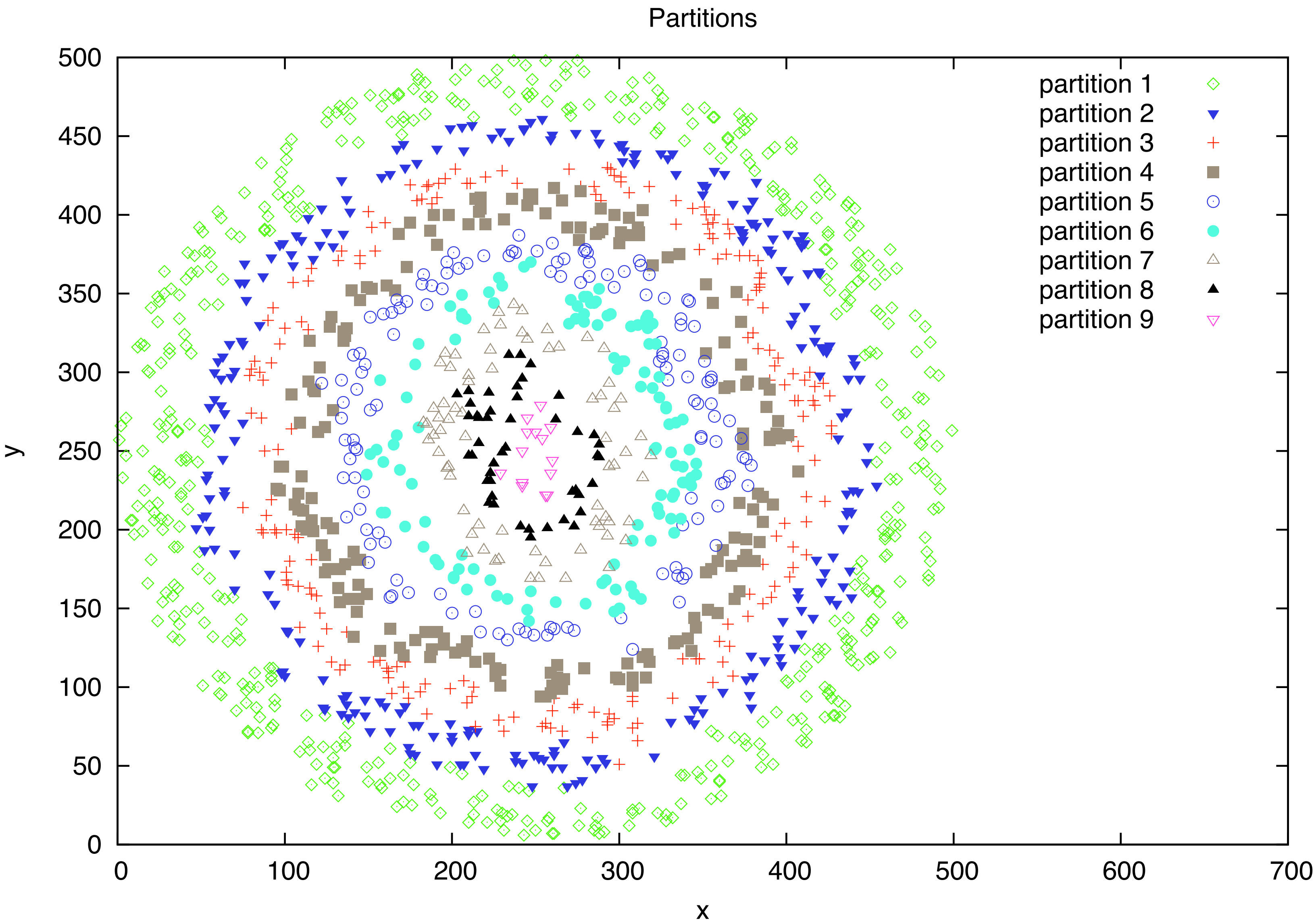


# Geometric structuring (2.1)

- Target partitioning
- $d = 1$, $p > 0$
- “Wave” waking up
- The function $f()$

$$\begin{aligned} f : \mathbb{N} &\longrightarrow \mathbb{N} \\ f(x_1) &\longrightarrow x_1 \end{aligned}$$

# Geometric structuring (2.2)



# Functional structuring (3.1)

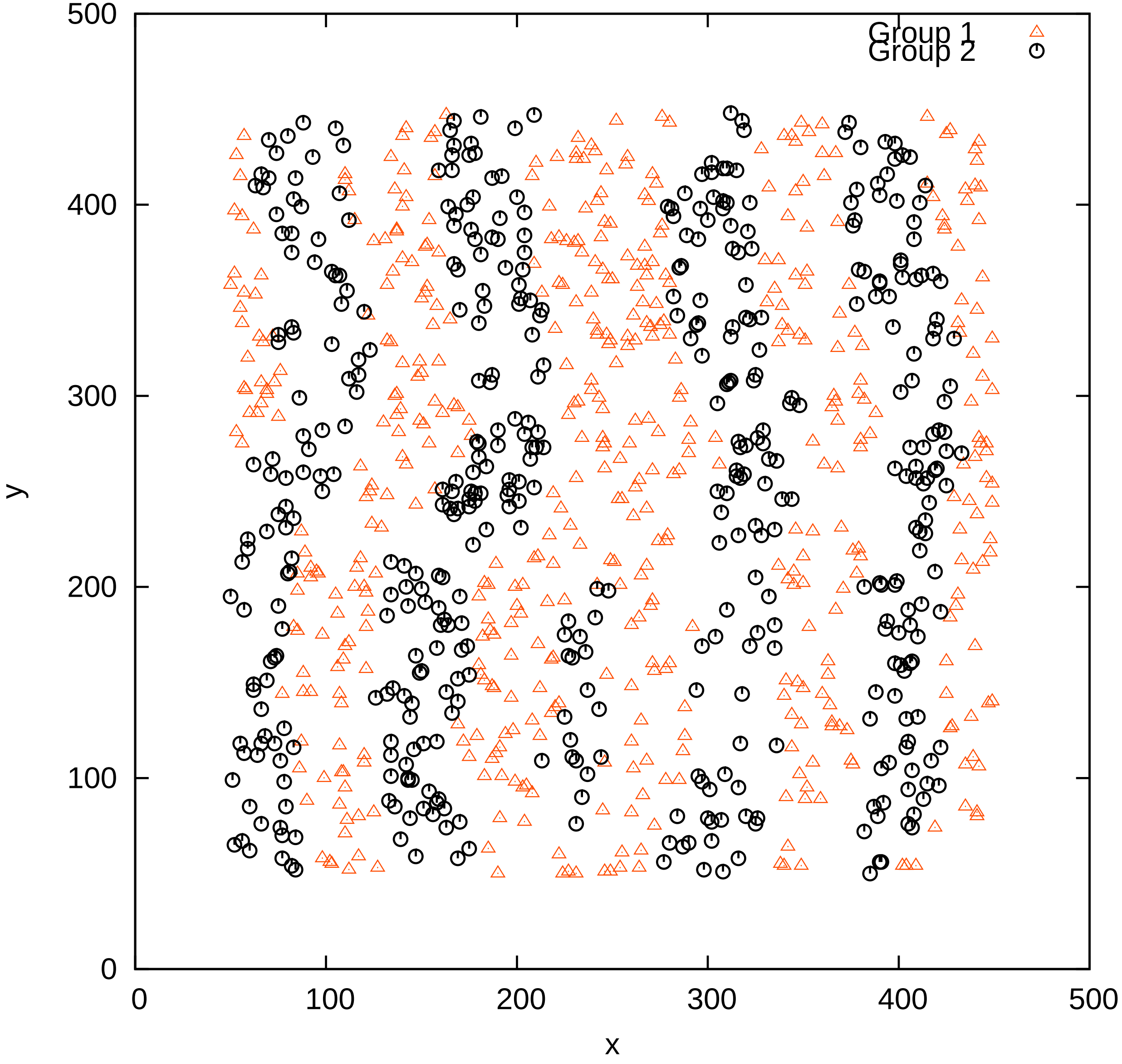
- Vertical lines partitioning
- $d = 4$, $p = 2$ (value of the modulo)
- The function $f()$

$$\begin{aligned} f : \mathbb{N}^4 &\longrightarrow \{0, 1\} \\ f(x_1, x_2, x_3, x_4) &\longrightarrow \text{max}(x_2, x_4) \text{ mod } 2. \end{aligned}$$

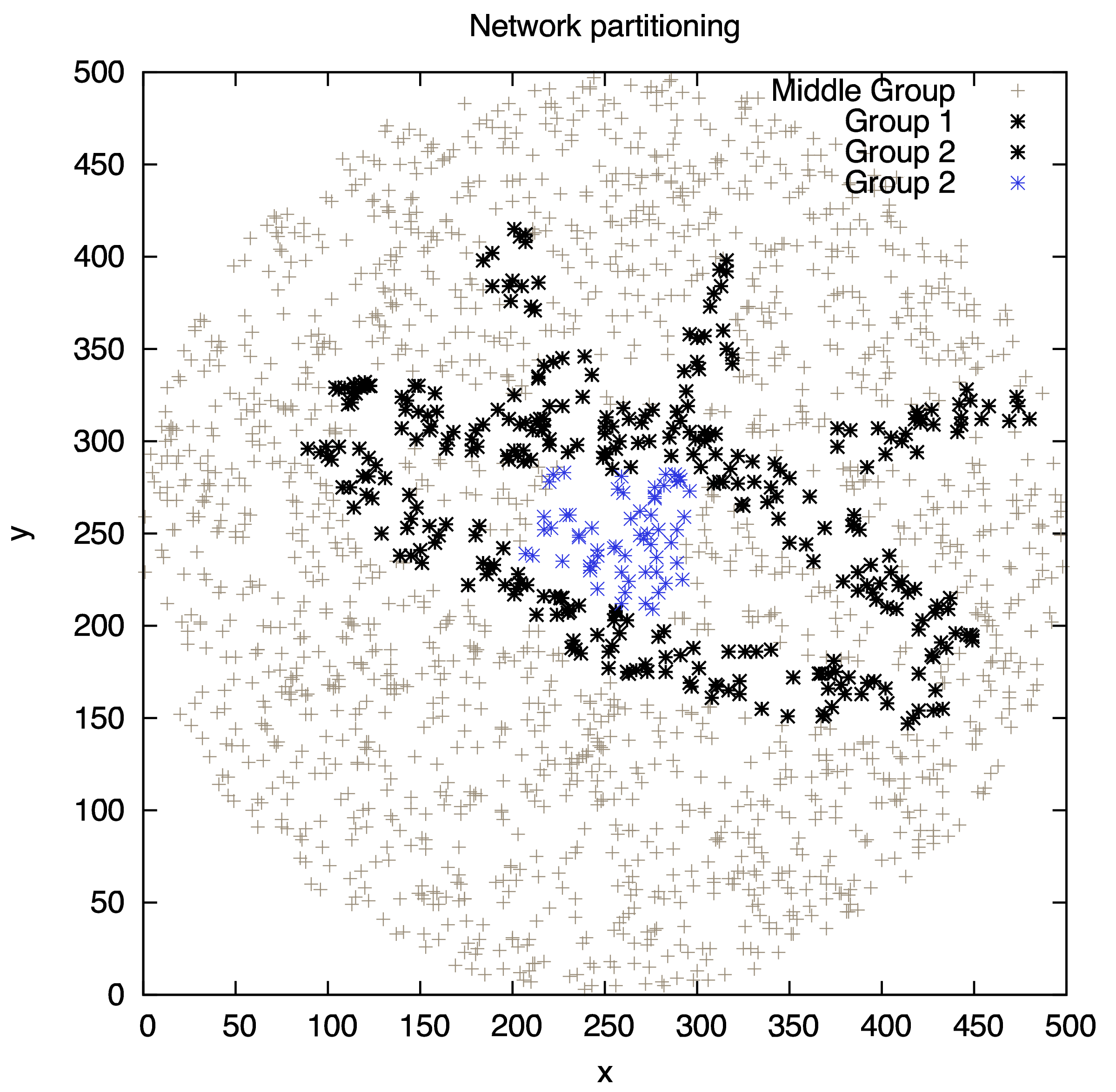
- Horizontal lines partitioning

$$\begin{aligned} f : \mathbb{N}^4 &\longrightarrow \{0, 1\} \\ f(x_1, x_2, x_3, x_4) &\longrightarrow \text{max}(x_1, x_3) \text{ mod } 2. \end{aligned}$$

# Functional structuring (3.2)



# Functional structuring (4.1)



# Functional structuring (4.2.1)

- Eye-like partitioning
- $d = 4$, $p = 5$ (value of the modulo)
- The function $f()$

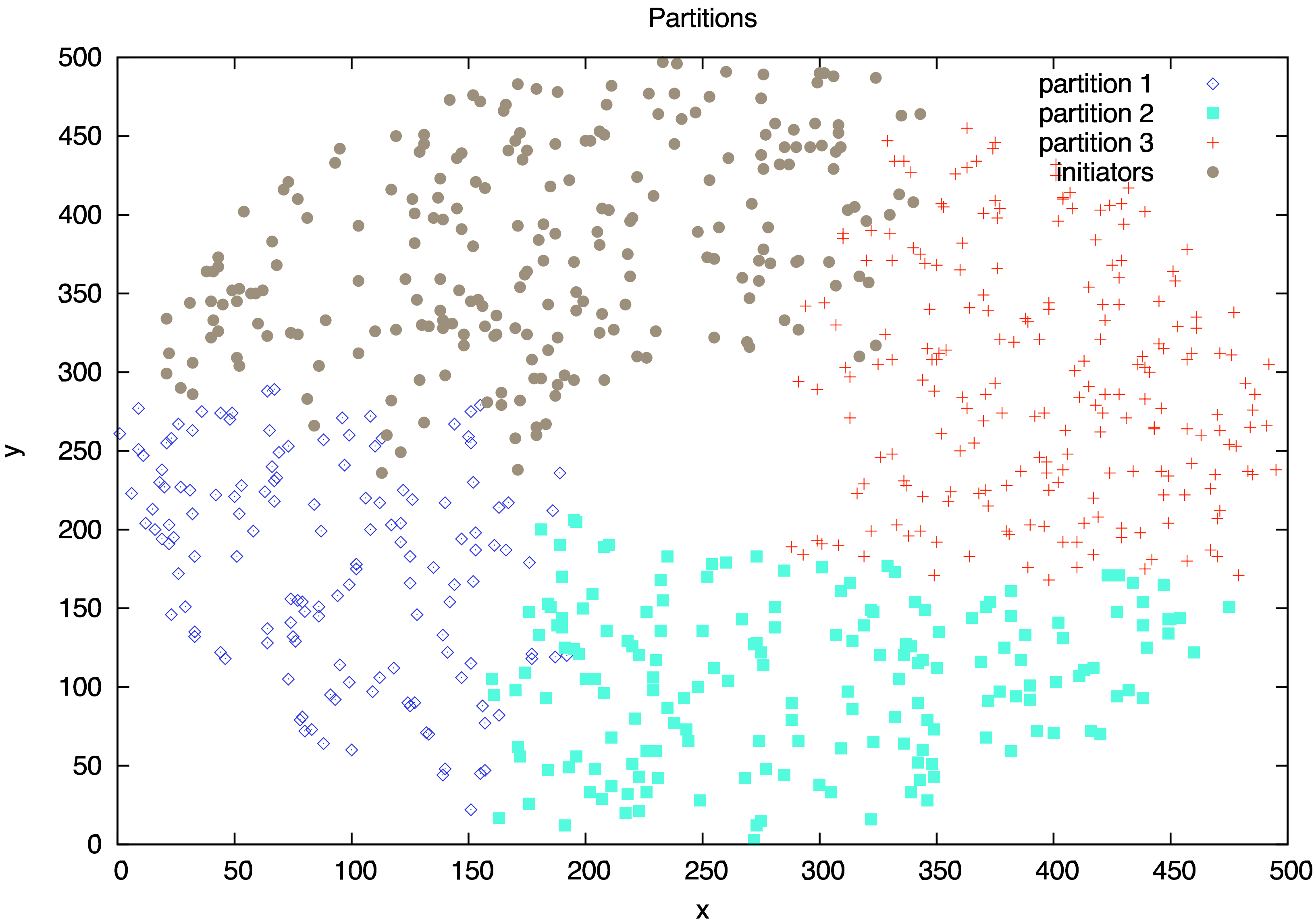
$$
\begin{aligned}
f : \mathbb{N}^4 &\rightarrow \{0,1,2,3,4\} \\
f(x_1, x_2, x_3, x_4) &\rightarrow \begin{cases} 0 & \textit{if} \quad \textit{eyelid} \\ 1 & \textit{if} \quad \textit{pupil} \\ 2 & \textit{if} \quad \textit{iris} \\ 3 & \textit{if} \quad \textit{eyelashes} \\ 4 & \textit{otherwise} \end{cases}
\end{aligned}
$$

# Functional structuring (4.2.2)

Where the predicates are

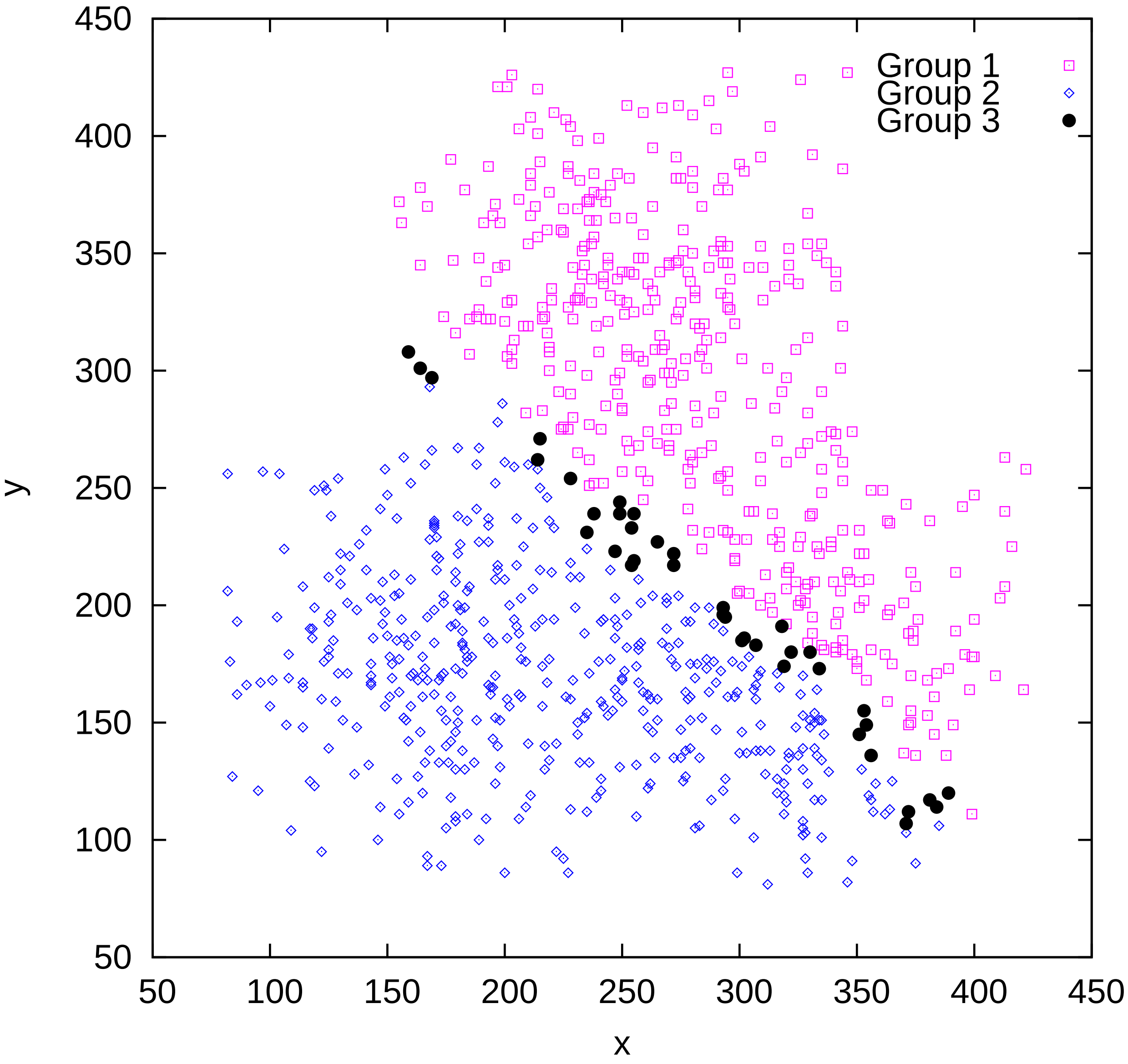
| Condition | Description |
|---|---|
| *eyelid* | $(x_0 = 9 \wedge x_2 < x_0) \vee (x_2 = 9 \wedge x_0 < x_2)$ |
| *pupil* | $x_1 = x_2 = x_3 = x_4$ |
| *iris* | $(\lvert x_0 - x_2 \rvert < 2) \vee (\lvert x_1 - x_3 \rvert < 2)$ |
| *eyelashes* | $(x_0 < 12) \wedge (x_1 = x_2 \vee x_2 = x_3 \vee x_1 = x_3)$ |

# Functional structuring (5): with a hole at the center



# Functional structuring (6)

The “line” predicate with three hot spots of different node densities

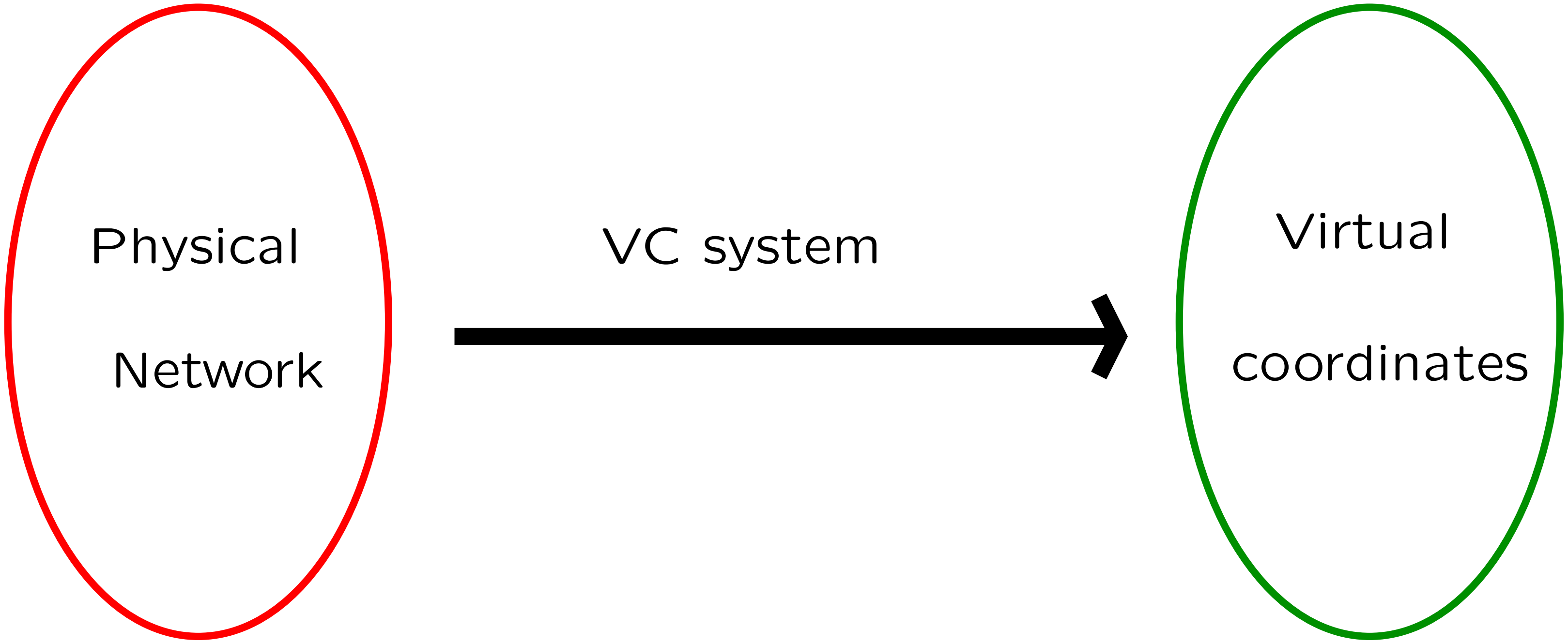


# Part IV

## WHAT IS A VIRTUAL COORDINATE SYSTEM?

# Physical vs logical neighborhood

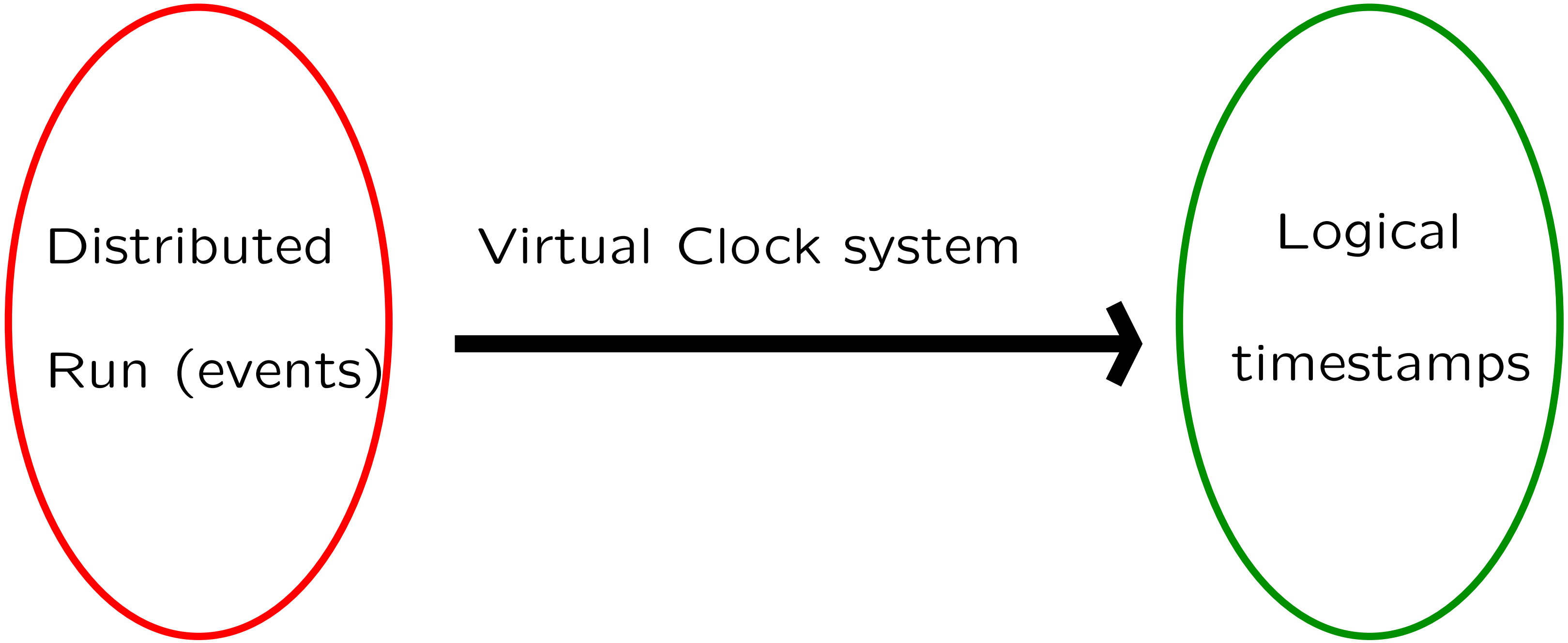
- $j \in com\_neighbors_i \Leftrightarrow i$ can directly communicate with $j$
- Cartesian coordinates (e.g., GPS) do not define a correct com_neigbhorhood relation (as they do not take into consideration natural obstacles)
- A virtual coordinate system associates a point in a $d$-dimensional (integer) space with each entity
- So, a coordinate of an entity $x$ is a $d$-uple $(x_1, \ldots, x_d)$
- How to characterize the "usefulness" of a given VC system?




Notion of communcation neighbors

Every $x$ has a "name" $(x_1, \ldots, x_d)$

# Reminder: how causality is captured




Notion of "physical passage of time"

Every event $x$ has a "name" $(x_1, \ldots, x_d)$

# Consistency of a virtual clock system

- Let $\rightarrow$ be the causality relation
- Observation: It is not because two events are close in time that they are causally related
- Lamport (scalar) clock (dimension $d = 1$)
  - ⋆ Let $a$ and $b$ timestamped $h_a$ and $h_b$
  - ⋆ Consistency: $a \rightarrow b \Rightarrow h_a < h_b$
- Vector clock (dimension $d = n$)
  - ⋆ Let $a$ and $b$ timestamped $vc_a$ and $vc_b$
  - ⋆ Consistency: $a \rightarrow b \Leftrightarrow vc_a < vc_b$
- Plausible vector clocks, approximate vector clocks, etc.

# Properties required from a VC system (1)

Observation: It is not because two entities are close in the physical space that they are communication-close (i.e., they are close from a communication point of view)

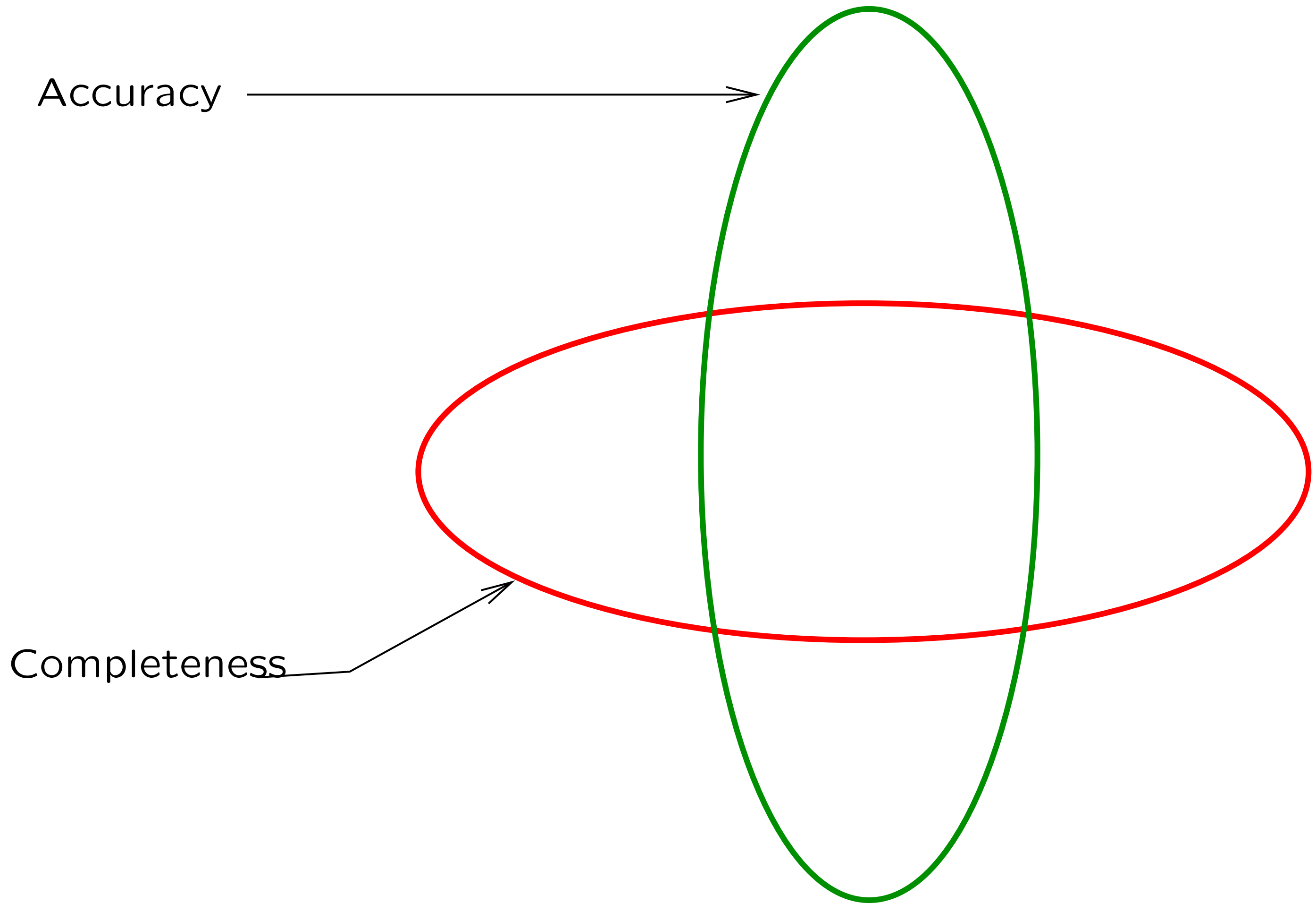
- Let VC be a virtual coordinate system
- Completeness: (no false negative)

  If $x$ and $y$ are com-neighbors, they are VC-neighbors

- Accuracy: (no false positive)

  If $x$ and $y$ are VC-neighbors, they are com-neighbors

# A more global view



# A solution?

- Let the virtual coordinate of an entity $x$ be the pair

$$(ID_x, \{ID_y \mid y \in com\_neigbhors_x\})$$

- The size of the coordinates depends on the connection degree (i.e., it is network dependent)
- This VC system is complete and accurate, but...
- It does not give "direct" information on the network, it is too much "local" (no notion of distance)
- In general we are interested in a VC system:
  - ⋆ Whose size $d$ is fixed a priori (i.e., network-independent)
  - ⋆ That gives (directly) "global" information

# Properties required from a VC system (2)

- Completeness and Accuracy are not sufficient
- The coordinates of the entities have to provide information that is globally consistent wrt their respective logical position
- This is captured by the following validity property: if two entities are communication-close (physical system), they are "close" in the VC system

# Properties required from a VC system (3)

- Completeness: (no false negative)

  If $x$ and $y$ are com-neighbors, they are VC-neighbors

- Accuracy: (no false positive)

  If $x$ and $y$ are VC-neighbors, they are com-neighbors

- Validity: To be valid, the set of coordinates has to be recognized by a distance function $d()$, i.e., a function $d()$ such that

  - $\star$ $d(VC_x, VC_x) = 0$
  - $\star$ $d(VC_x, VC_y) = d(VC_y, VC_x)$
  - $\star$ $d(VC_x, VC_y) \leq d(VC_x, VC_z) + d(VC_z, VC_y)$

# Returning to the proposed VC system

- $x$ and $y$ are VC-neighbors if $\bigwedge_{1 \le i \le d} (|x_i - y_i| \le 1)$
- The VC system satisfies the completeness property
- $d(VC_x, VC_y) = \max(\{|x_i - y_i|\}_{1 \le i \le d})$ is an appropriate distance function
- $VC_x$ and $VC_y$: VC-neighbors $\stackrel{\text{def}}{=} d(VC_x, VC_y) \le 1$
- Remark: There is no $f()$ based only on ids that allows to define a distance function (which means that ids cannot be used to define a VC system)
- Other interesting properties can be revealed by the existence of other distance functions that recognize VC

# Results for Homogeneous Unit Disk Graph (UDG)

- Homogeneous sensor network: modeled by a UDG
- Computing VC for sensors becomes then finding a representation of a given UDG
- It is NP-hard to determine a set of virtual coordinates that satisfy all the UDG constraints for any given unit disk graph [Breu and Kirpatrick, 1998]
- Even approximating the constraints to within a factor of $\sqrt{3/2}$ is NP-hard [Khun, Moscibroda and Wattenhofer, 2004]

# Weakening the properties

- Accuracy: (no false positive)

  If $x$ and $y$ are VC-neighbors, they are physical neighbors

- Idea: quantify the confidence on physical neighborhood obtained from virtual coordinates

- $\epsilon$-Accuracy: (the aim is to reduce false positive)

  If $x$ and $y$ are VC-neighbors, they are com-neighbors with probability $1 - \epsilon$

# Other properties

- Let $x$ and $y$ with VC $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$
- Granularity: (weaken precision)

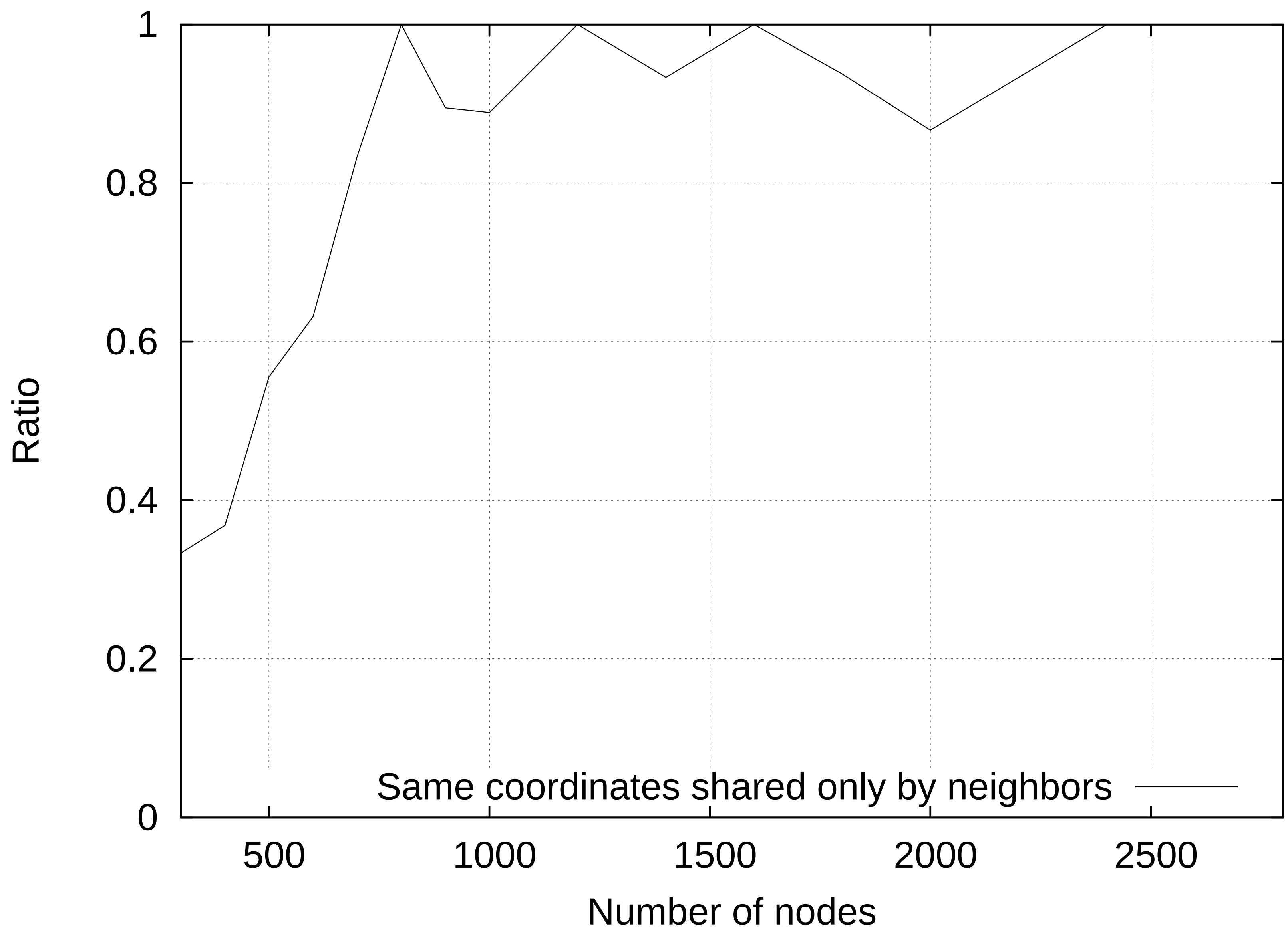
  $x$ and $y$: not com-neighbors $\Rightarrow \exists i : \; x_i \neq y_i$

  Two entities that are not com-neighbors cannot be confused: they have different virtual coordinates

- Sharpness: $x \neq y \Rightarrow \exists i : \; x_i \neq y_i$

  Sharpness is stronger than granularity (no two entities have the same VCs)

# Ratio of neighbors with same VC



# CONCLUSION

# What we have seen

- Context: autonomous self-structuring systems
- Design of an algorithm that assigns virtual coordinates to entities
- Very weak assumptions, very distributed and localized
- Geometric structuring
- Bypassing the routing pb: what is a VC system?