

Decentralised Periodic Encounter Community Detection

VLunch
27th May 2010

Matthew Williams
(*M.J.Williams@cs.cf.ac.uk*)

Supervisors: Dr. Roger Whitaker, Dr. Stuart Allen

Overview

- Background and motivation
- Periodic encounter communities
- Decentralised periodic encounter community detection
- Analysis of decentralised detection algorithm
- Conclusions and future work

Background & Motivation

Opportunistic Networks

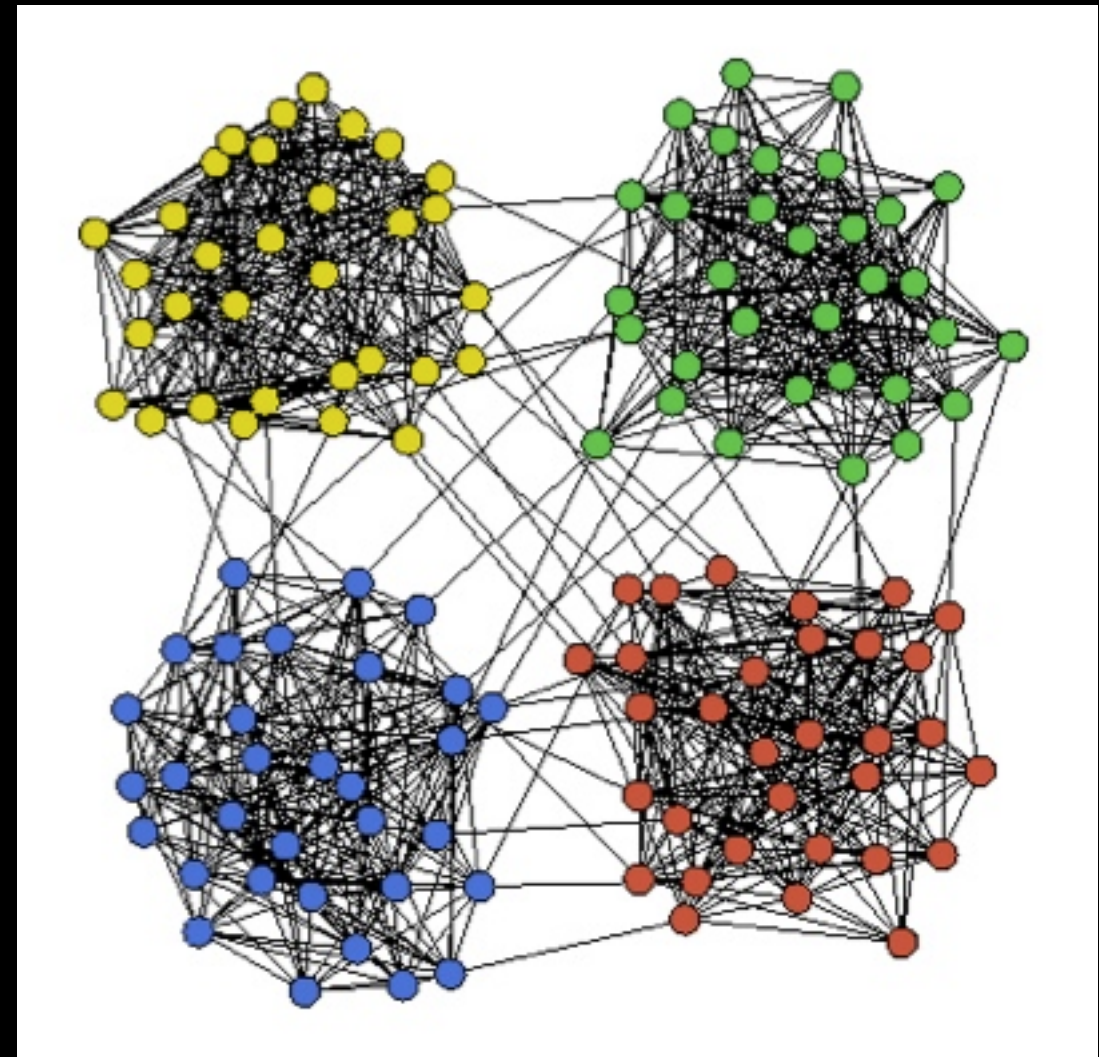
- **Opportunistic networks (oppnets)** are a broad class of networks where messages are spread by the **mobility of individuals** and their occasional **physical encounters**
- **Encounters** are the fundamental unit of communication in these networks
- In many real-world cases, the behaviour of these nodes results in temporary **physical communities** occurring
 - Example networks include: **human** (PSNs), **vehicular** (VANETs), and **wildlife monitoring**

Encounter-Aware Content Sharing in OppNets

- Recently, oppnets content sharing protocols have tried to capture **patterns in encounters**
- Attempt to assess likelihood of seeing node in the future
- Some protocols model **periodicity** of encounters
- All models only consider immediate neighbours -- broader **community structure** is ignored
- All models require **predefined** domain-specific periods to be set

Static Community Detection

- Identify **components** in large graphs
- Global-knowledge, offline algorithms
- Static: single, time-agnostic graph
- Distributed algorithm used in oppnets content sharing

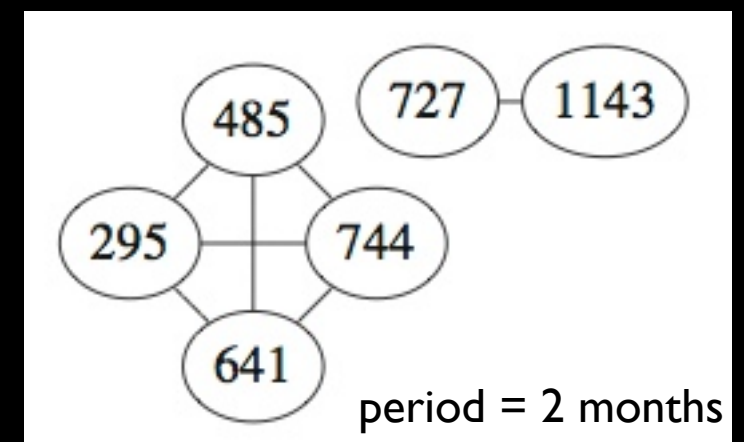
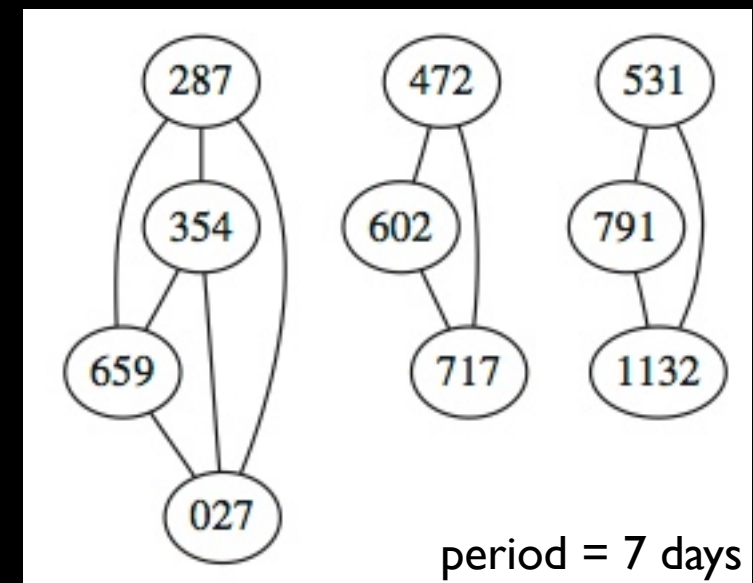


Fortunato 2010

Periodic Communities

- It is intuitive that the underlying behaviour of nodes results in **communities** of nodes **re-appearing regularly** in time
- Also evidenced in empirical datasets by **PSE-Miner** and other analyses
- We seek to join the concepts of **node communities** and **periodicity**
- **Decentralised** approach necessary in oppnets
- With **automatic** detection of **periods**

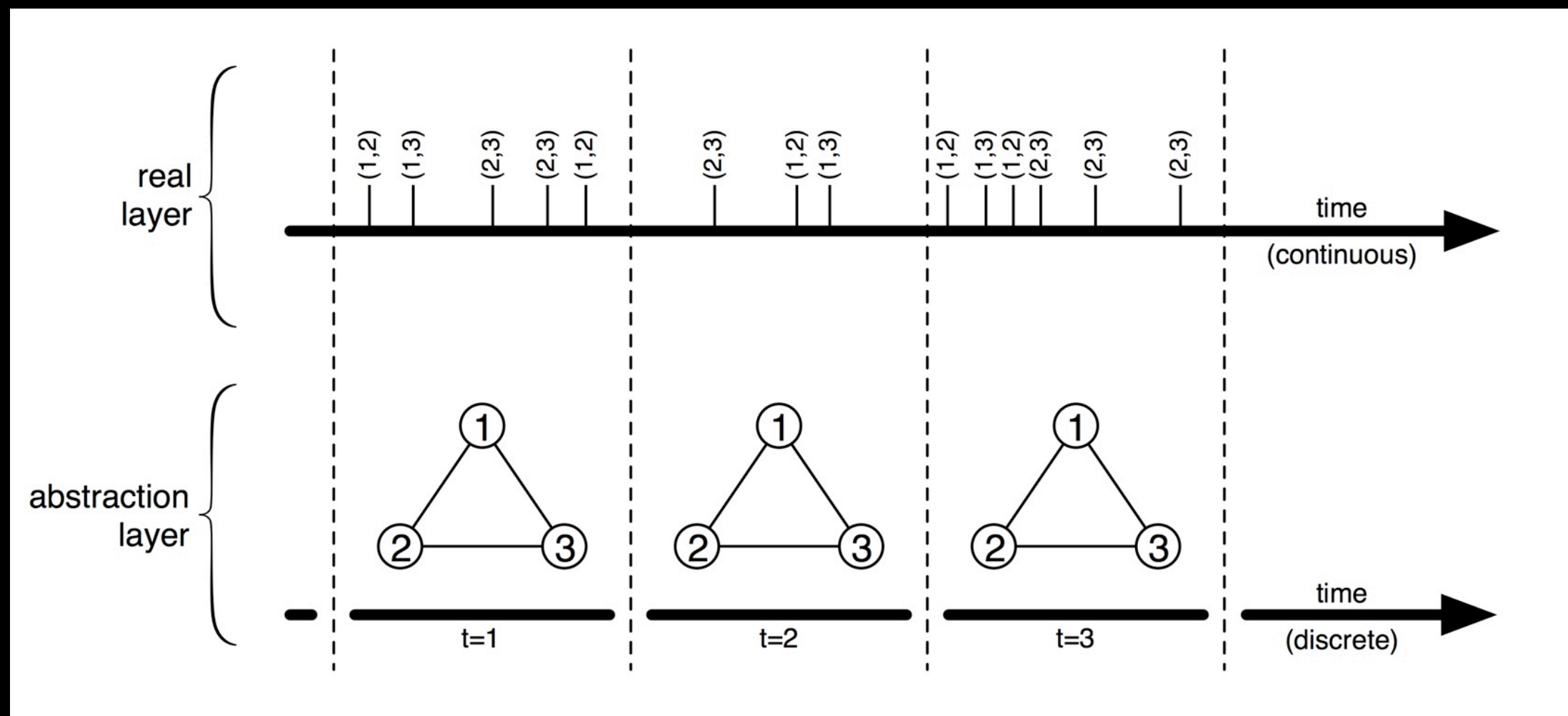
Periodic Zebra Communities



Periodic Encounter Communities

Dynamic Encounter Representation

- A **dynamic encounter network** is a time series of graphs
- Each graph is a **snapshot of encounters** occurring during a time interval



Periodic Encounter Community

- We formalise a Periodic Encounter Community (**PEC**) as

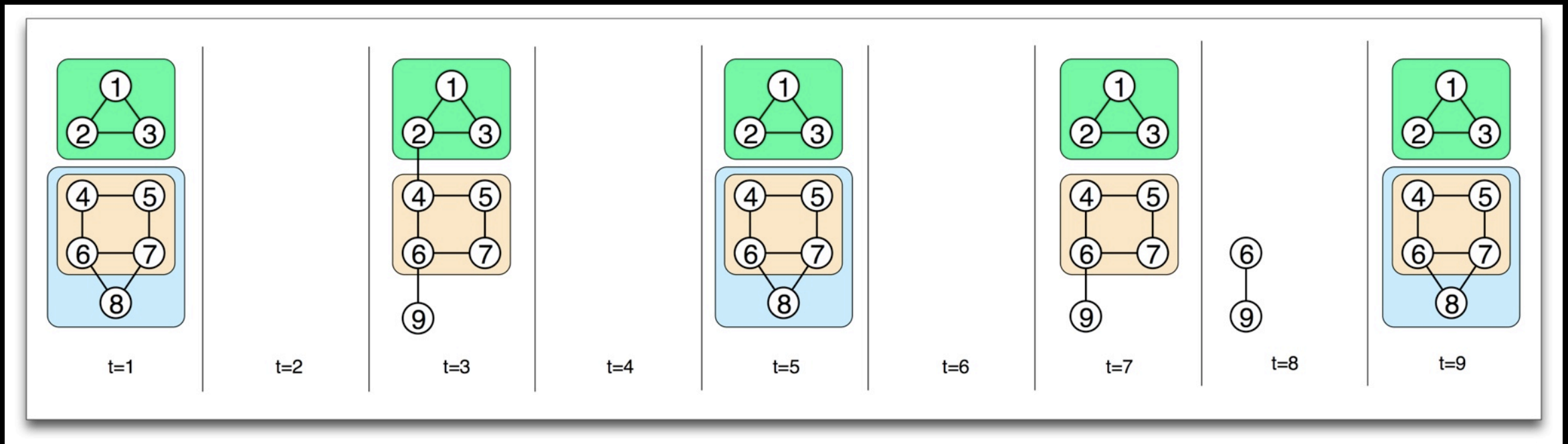
$$\langle C, S \rangle$$

- where
 - C is a **connected** graph (the **community**)
 - S is the **harmonic information**

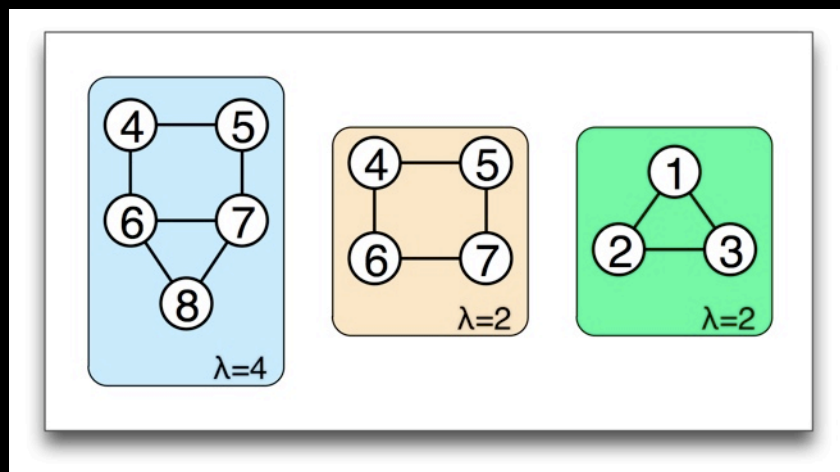
$$S = (t_{start}, t_{end}, \lambda)$$

PEC Example

Example Dynamic Network

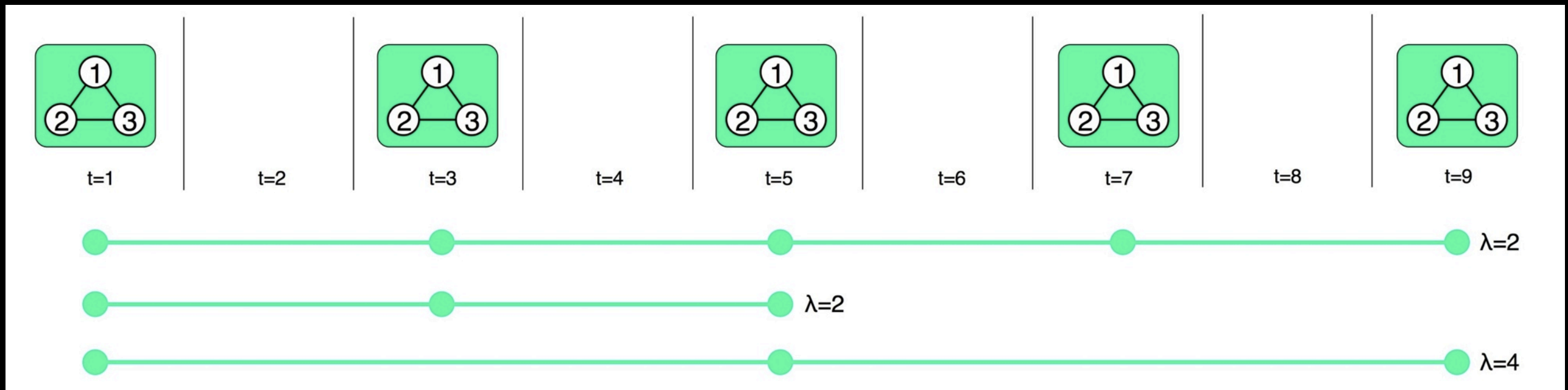


Periodic Encounter Communities



PEC Redundancy

- Some PECs capture more information than others
- One PEC may **subsume** another's information



- **Harmonic maximality:**
 - Multiple ways to fit harmonic information to the same community, but only one is **parsimonious**

Maximality and Parsimony

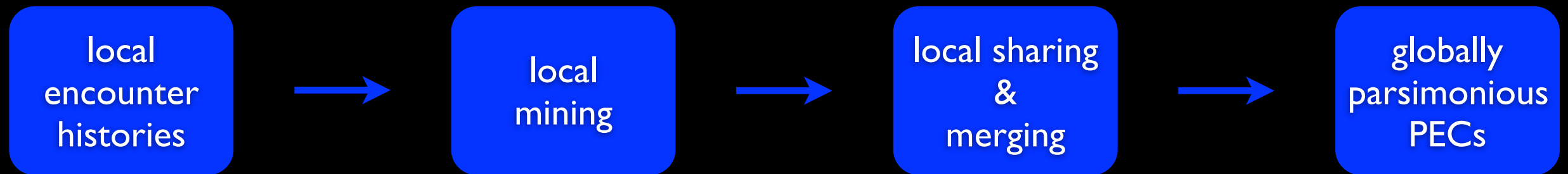
- **Harmonic** maximality:
 - Community does not exist for factors of the period, nor can it be extended in time
- **Structural** maximality:
 - Cannot add edges or nodes to the community and still maintain its existence in the dynamic network
- **Parsimony:**
 - A PEC is **parsimonious** if it is both **harmonically** maximal and **structurally** maximal

Decentralised PEC-D Problem

- Decentralised PEC Detection is the problem of having all nodes detect the **parsimonious** PECs they belong to, without **global knowledge** of the network

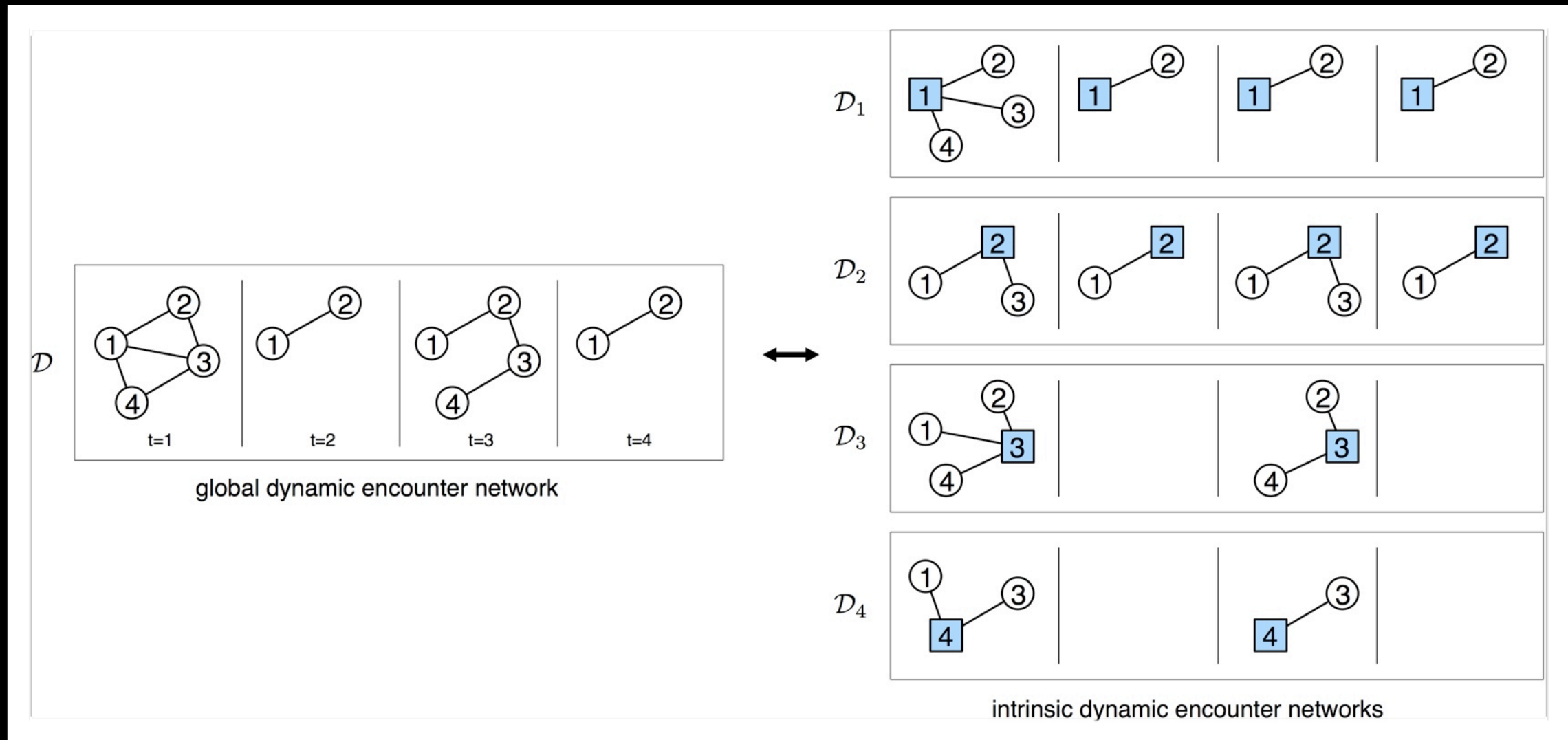
Decentralised PEC Detection Algorithm

Algorithm Overview



- Local Mining:
 - Obtain PECs that are parsimonious in their **local encounter histories**
- Local Sharing:
 - Nodes share and combine their intermediate parsimonious PECs when they meet
- Over time, nodes build towards the PECs that are parsimonious in the **global dynamic graph**

Intrinsic Dynamic Networks



- Global dynamic network can be decomposed into **intrinsic dynamic networks**
- Intrinsic DN corresponds to the encounter information **directly observable** by a node

Local Miner Algorithm

- Invertible map from graphs to sets of integers
 - Edges and nodes given unique integer identifiers
- Becomes a problem of mining **periodic subsets** in a time series of integer sets
- **Periodic pattern mining** in **temporal data mining** field
- Polynomial time complexity
- Local returns **locally-parsimonious** PECs

Joining PECs

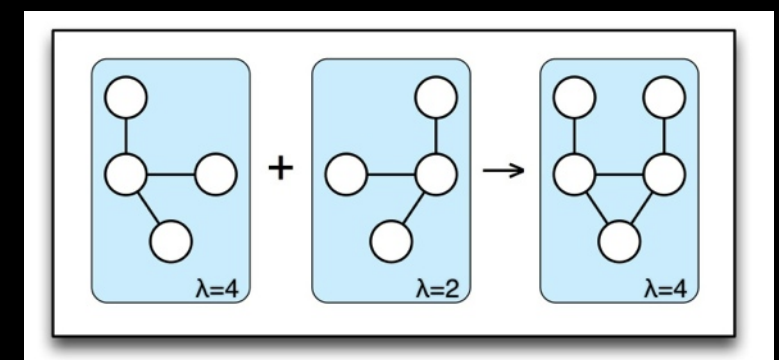
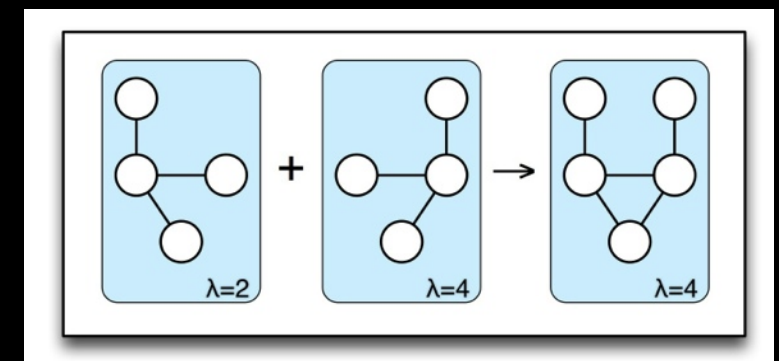
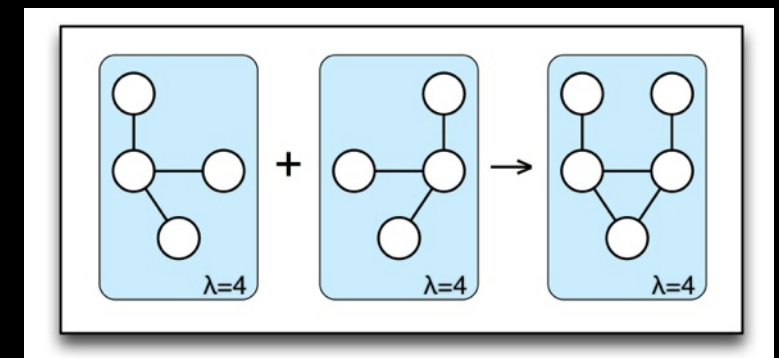
- Two PECs are **compatible** if the following hold:
 - their communities intersect
 - the PECs are harmonically equal, or one harmonically subsumes the other
- If compatible, there are three generation cases:

case	action	
harmonic equality	merge communities	keep harmonic information
P1 harmonically subsumes P2	merge communities	harmonic information from P2
P2 harmonically subsumes P1	merge communities	harmonic information from P1

Opportunistic Construction

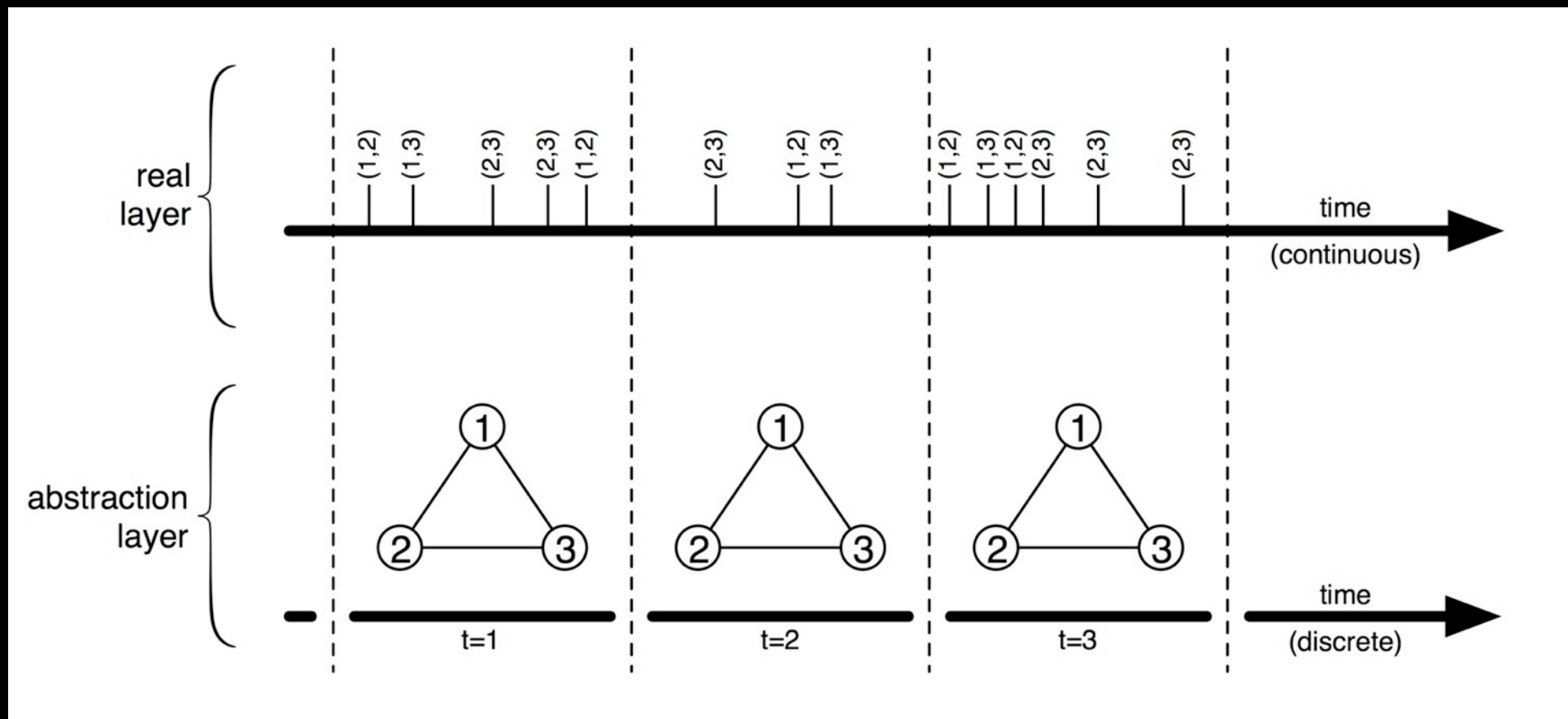
- Each node holds local Knowledge Base (**KB**) of its PECs so far
- Node only holds non-subsumed PECs which node itself belongs to
- On encounter, a pair of nodes:
 - Share KBs
 - Generate candidate PECs
 - Store any more-maximal candidates
 - Remove any redundancies

PEC Generation Cases



Analysis of Decentralised Detection

PEC Construction Lag



- Parsimonious PEC construction relies on encounters between nodes
- The underlying ordering of edges influences how PECs propagate to nodes
- Given the encounters predicted by a PEC, how long does it take the nodes to reach parsimony?

Worst Case Construction Lag

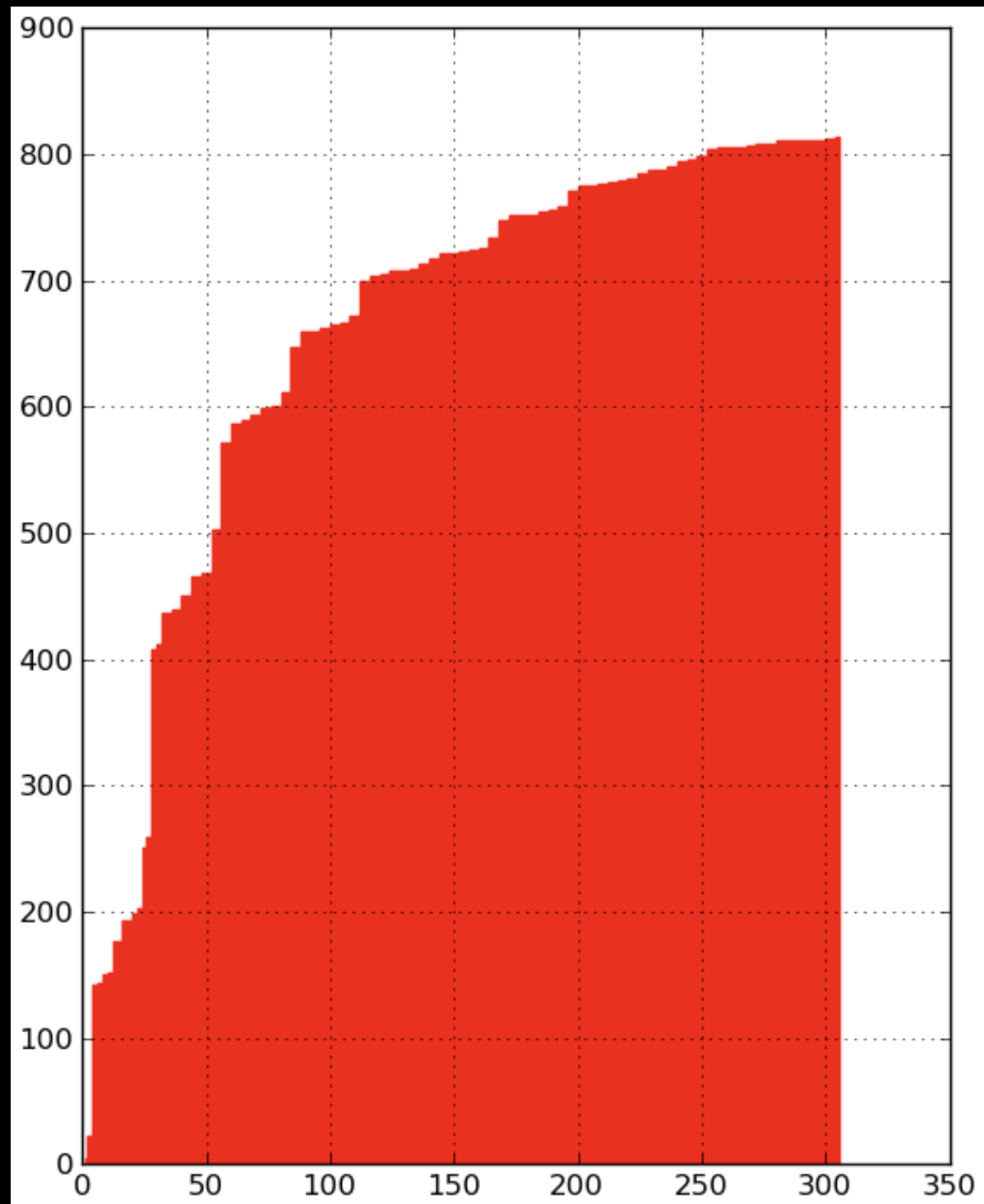
- Pessimistic constraints:
 - No encounters extrinsic to the PEC -- only use edges predicted by the PEC
 - No repeat encounters in same time step
- Under these constraints, construction lag depends on underlying ordering of encounters
- With a worst-case encounter ordering, the construction lag for a PEC is function of the period and diameter:

$$\lambda(d - 1) + 1$$

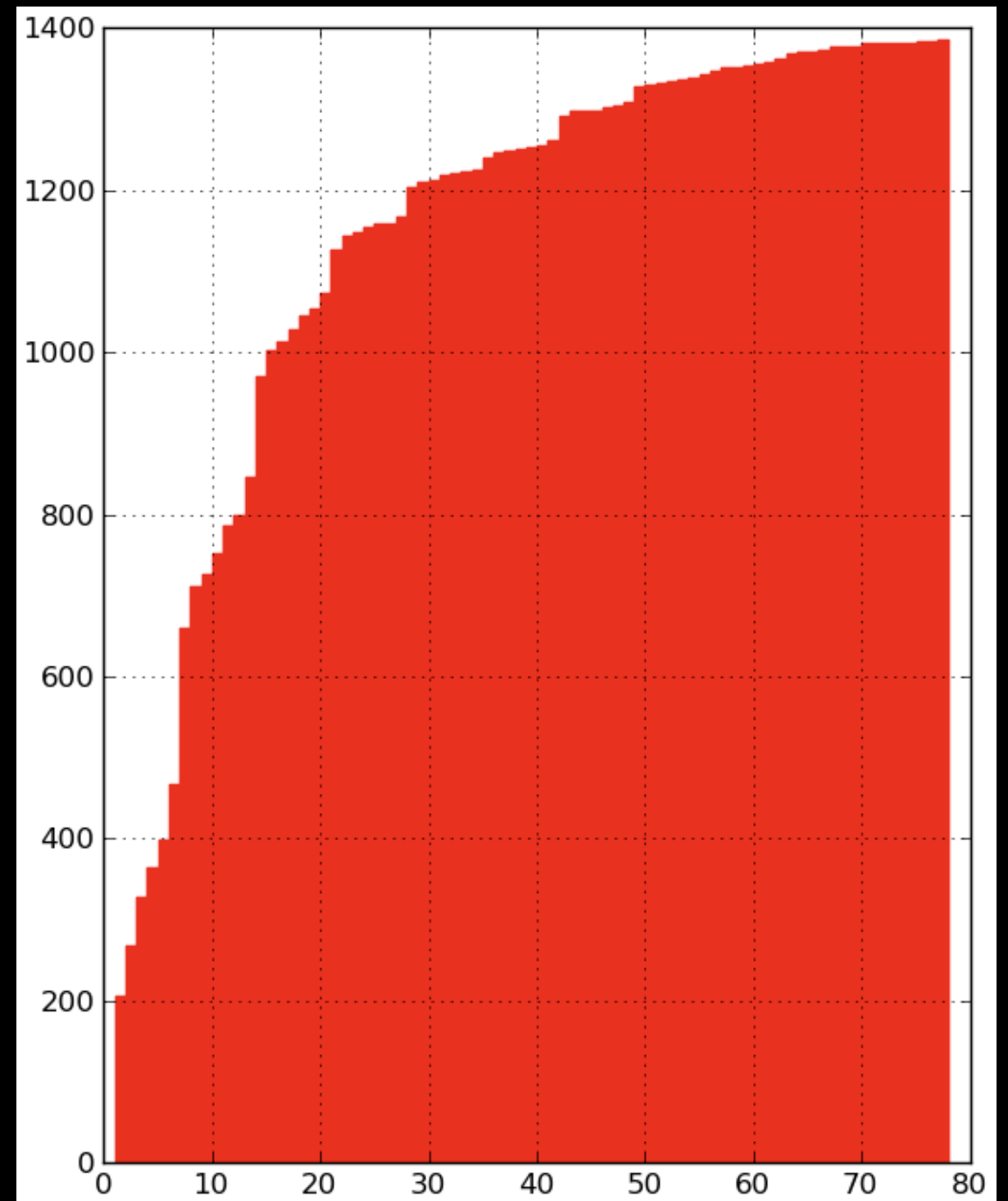
Real Data Experiments

- MIT Reality Mining data
 - Bluetooth encounters between 100 subjects over 9 months
- Construction lag experiments
 - For a given parsimonious PEC, using only the edges predicted by that PEC, how long would it take the whole community to detect the PEC?
 - With reference to the worst-case?

PEC Periodicities (cumulative dist.)



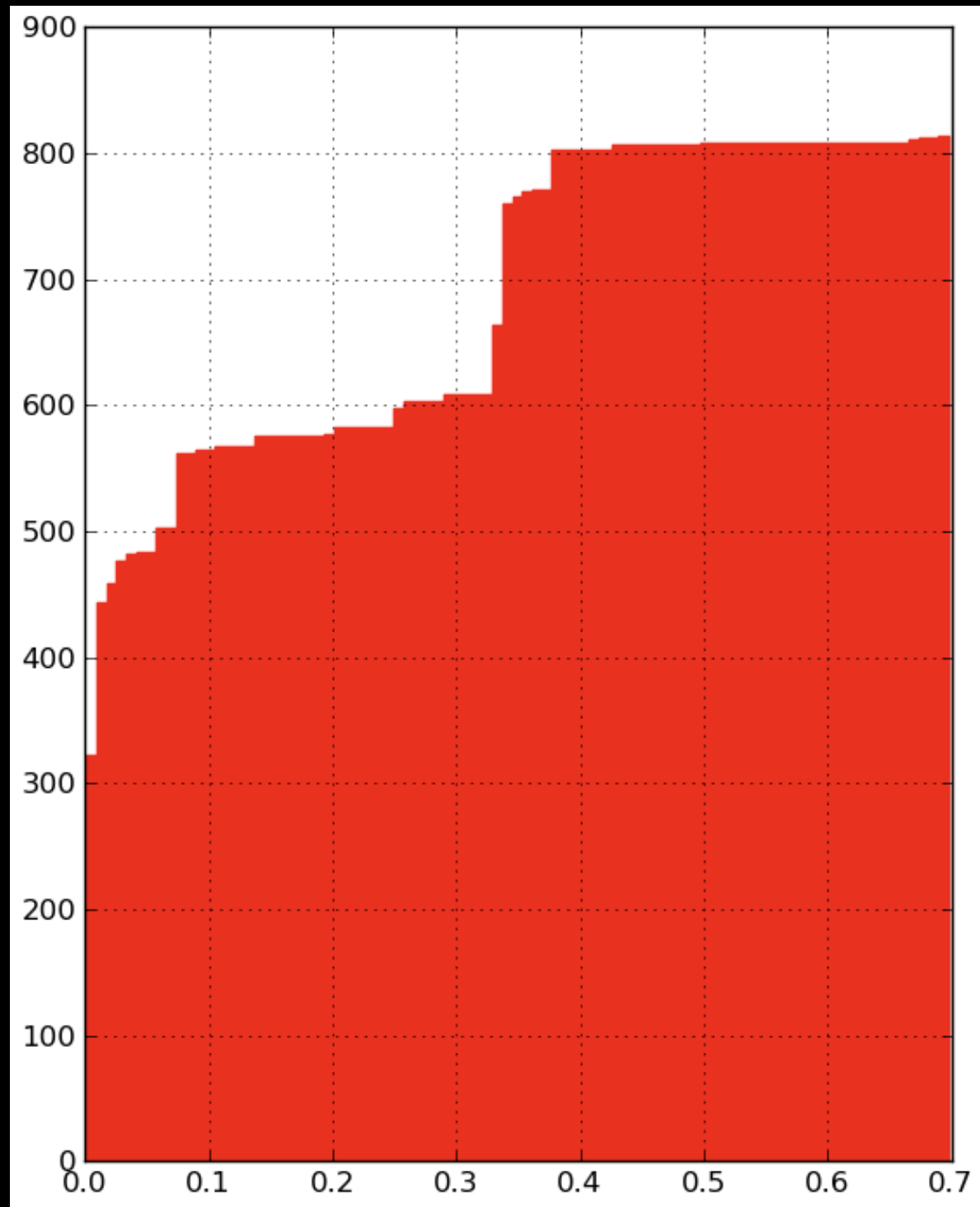
granularity = 6 hours



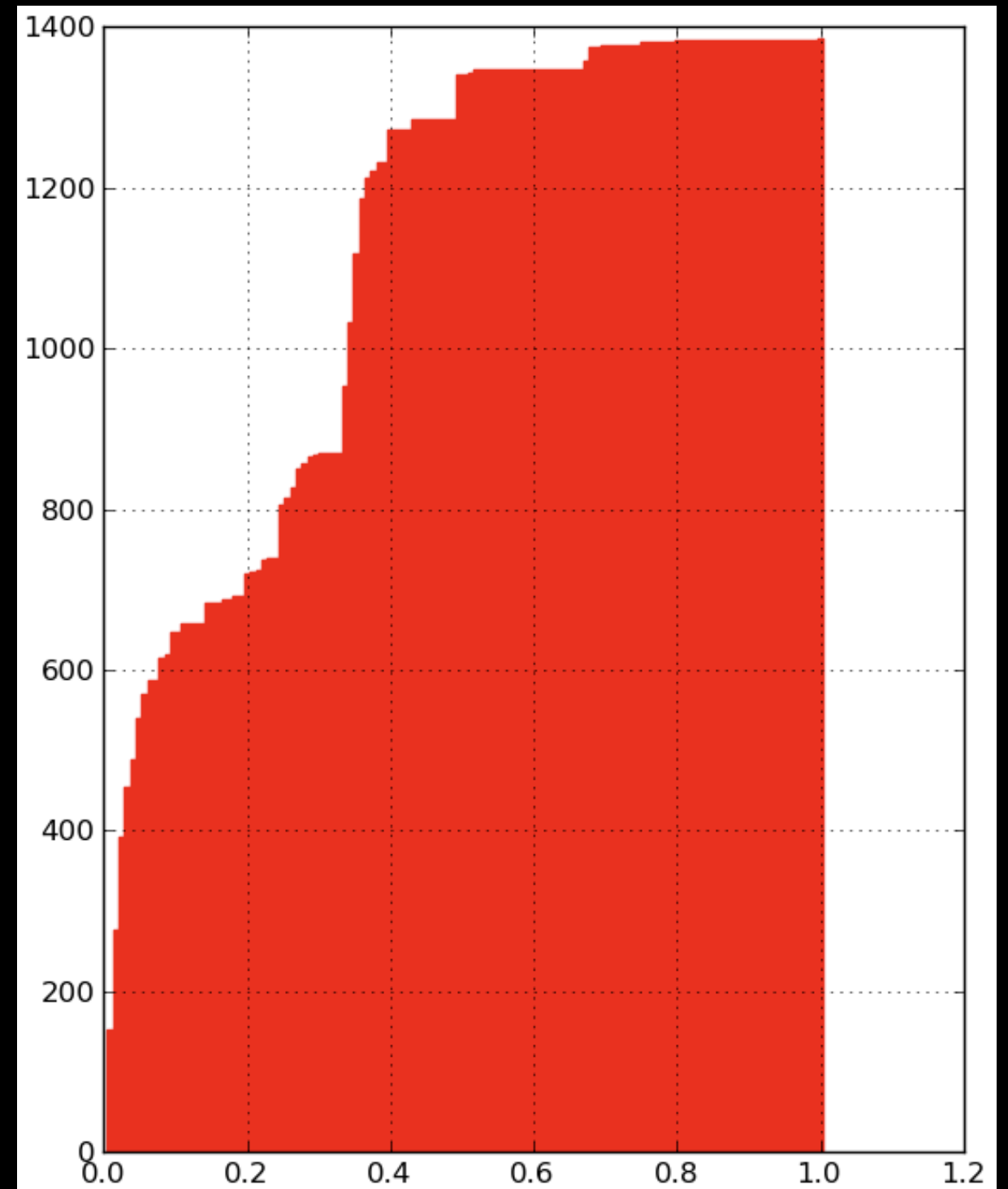
granularity = 24 hours

Normalised Construction Lags

(cumulative dist.)



granularity = 6 hours



granularity = 24 hours

Conclusions

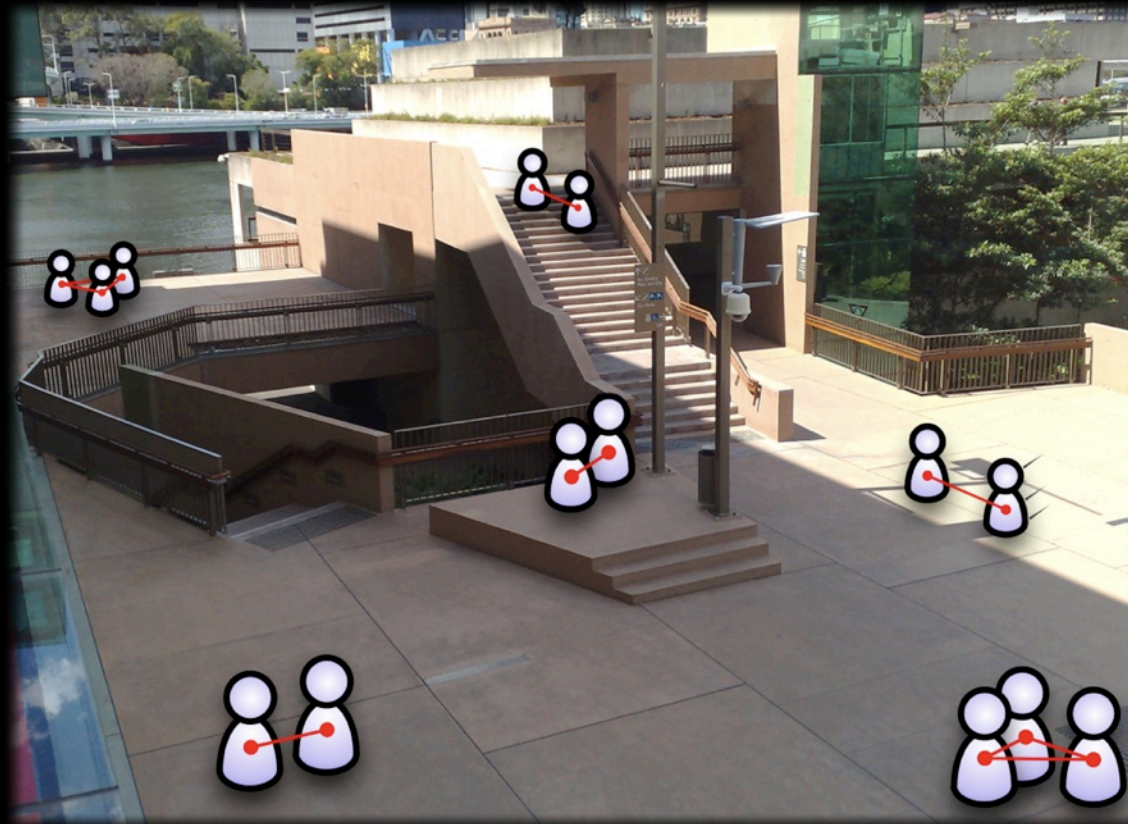
Conclusions

- Globally parsimonious PECs can be mined decentrally, and with automatic periodicity identification
- Time for globally parsimonious PEC construction is bounded by PEC period and diameter
- On real data, construction time is much better than the analytic worst-case

Future Work

- Other datasets
- Synthetic models for generating encounter orderings
- Fuzzy PEC-D
- PEC-D applied to oppnets content sharing

Thanks!



Any questions?

socialnets

<http://www.social-nets.eu>

Matthew Williams

<http://users.cs.cf.ac.uk/M.J.Williams>

M.J.Williams@cs.cf.ac.uk

References

M. Lahiri and T.Y. Berger-Wolf. *Periodic subgraph mining in dynamic networks*. Knowledge and Information Systems, Volume 24, Issue 3 (2010), p. 467.

S. Fortunato. *Community detection in graphs*. Physics Reports, 486 (3-5) (2010), pp. 75–174

Attribution

Library Courtyard. nevolution. <http://www.flickr.com/photos/nevolution/2906377551/>