Finding Hidden Patterns: Temporal Scale for Dynamic Graphs

Hemanshu Kaul

Illinois Institute of Technology

www.math.iit.edu/∼kaul

kaul@iit.edu

Joint work with

Rajmonda Caceres (MIT-Lincoln Lab) and Michael Pelsmajer (IIT)

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

These "objects" can be:

- People
- Animals
- Proteins
- Countries
- Cities
- Computers
- Websites
- Other Mathematical objects
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

The relationships can be based on:

- Social Interaction
- Proximity
- Chemical Reaction
- Communication connection
- Overlap/ Intersection
- any sort of Dependence
- and so much more

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of “objects”.

“Objects” are called vertices.

Relationships between pairs of vertices are called edges.

What is a Graph?

Graph (or Network) is a mathematical structure that represents relationships within a collection of "objects".

Formally, a graph $G = (V(G), E(G))$, the objects under study are represented by vertices included in V(G).

If two objects are "related" then their corresponding vertices, say u and v in $V(G)$, are joined by an edge that is represented as uv in $E(G)$.

Examples of Graphs

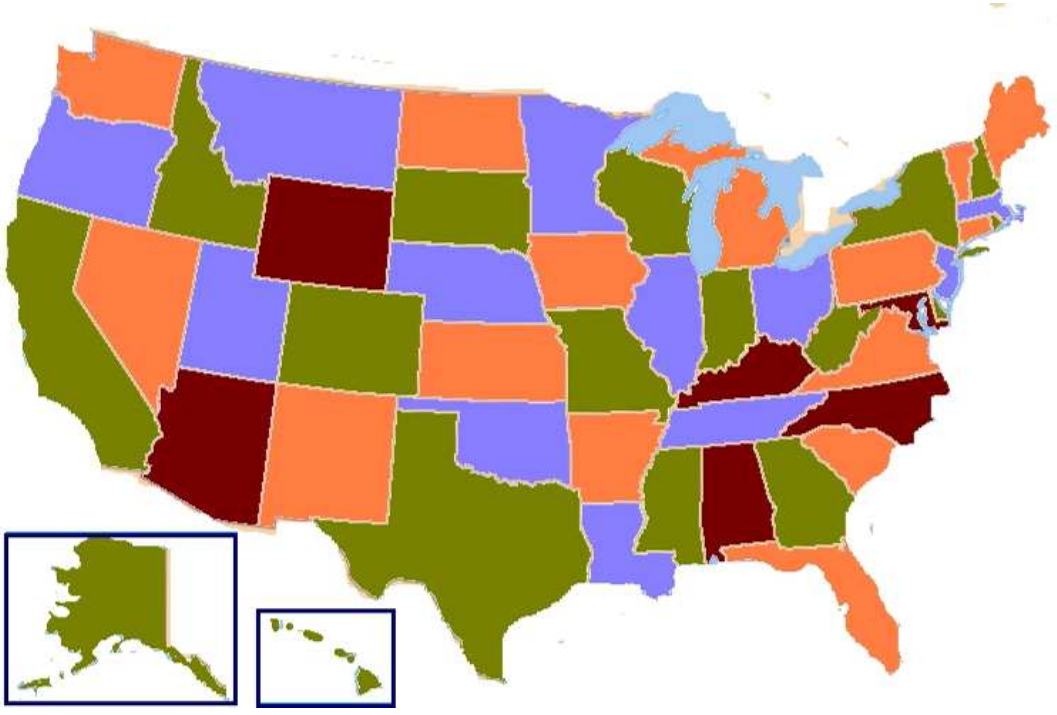
For a university semester, we could define a 'conflict' graph on courses: where each course is a vertex, and edges occur between pairs of vertices corresponding to courses with overlapping timeslots.

More generally, Conflict Graph and Scheduling Problems.

The most famous example of this is the map coloring problem.

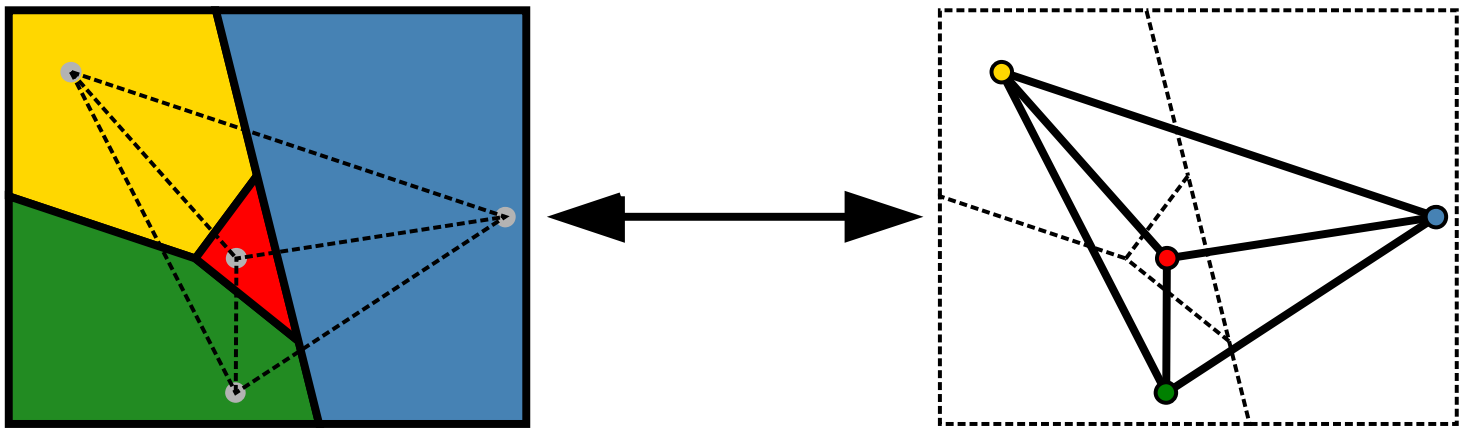
Examples of Graphs

What is the least number of colors needed to color a map so that adjacent regions in the map get different colors?



Examples of Graphs

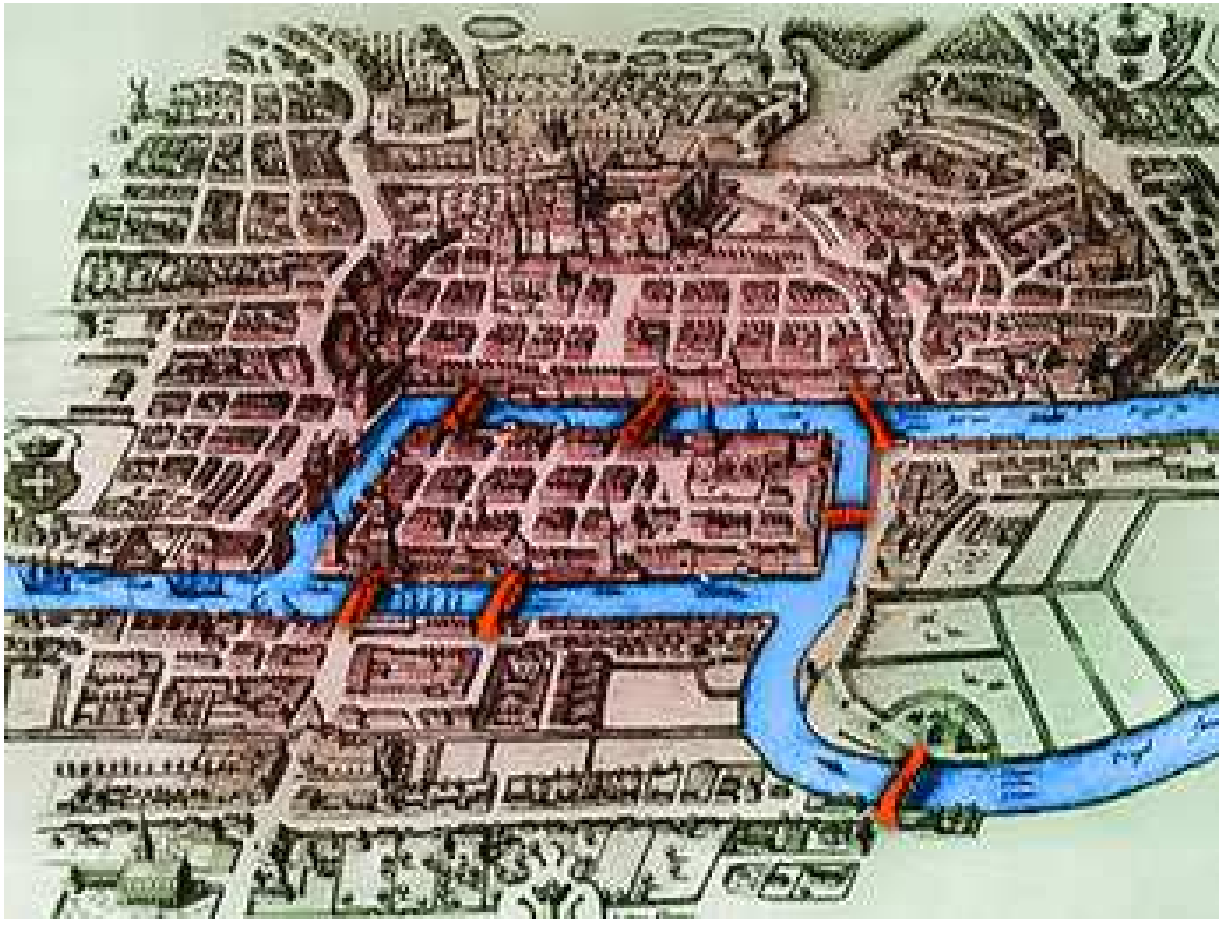
What is the least number of colors needed to color a map so that adjacent regions in the map get different colors?



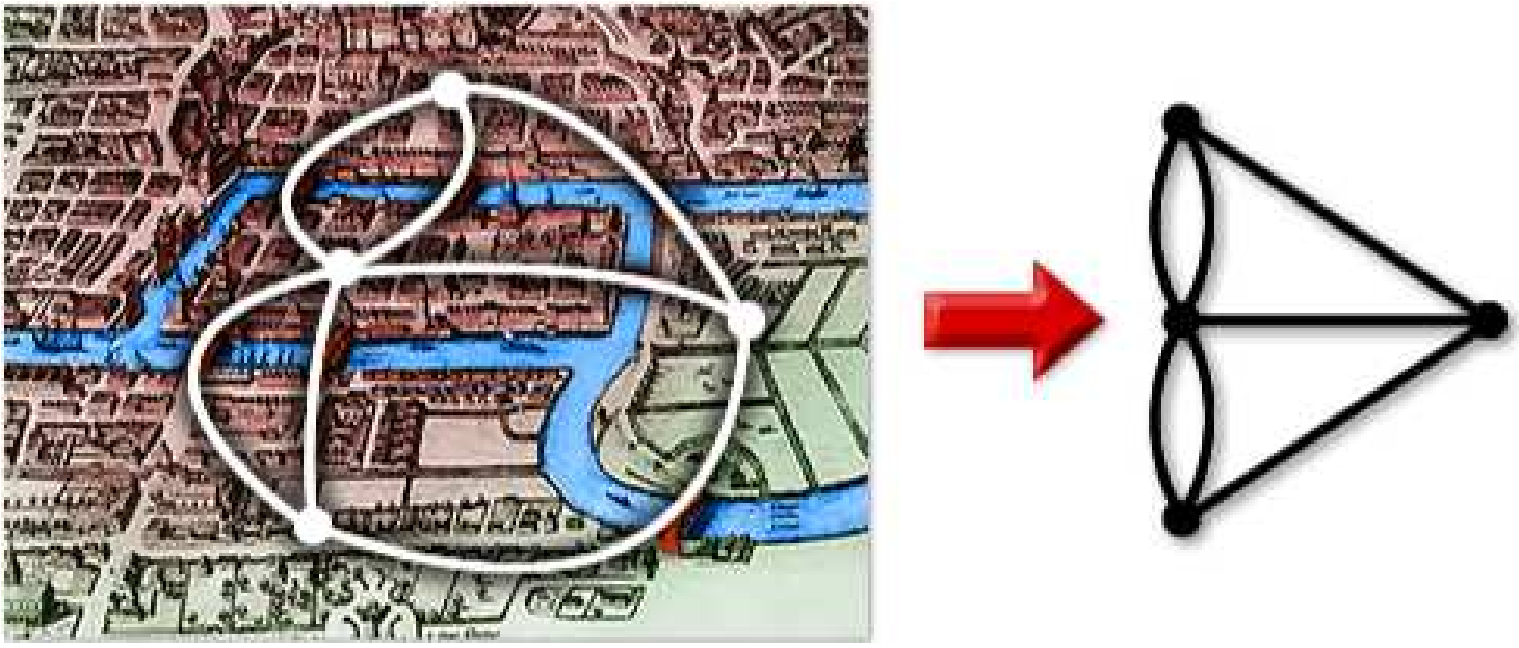
(All figures courtesy: Wikimedia.org)

Examples of Graphs

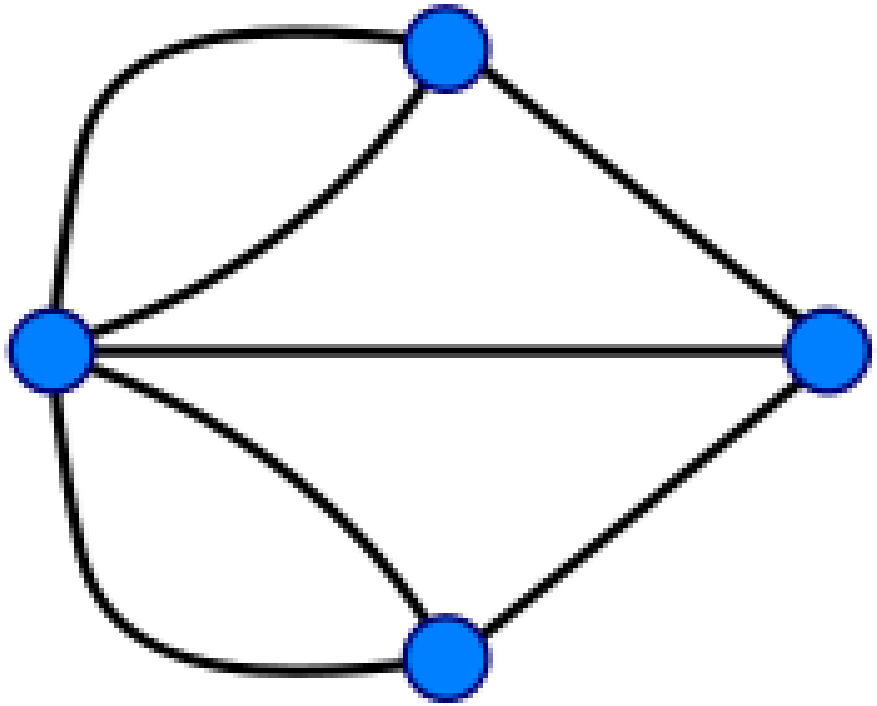
Historically, the first explicit use of Graphs was by Euler for solving Konigsberg Bridges Problem:



Examples of Graphs



Examples of Graphs

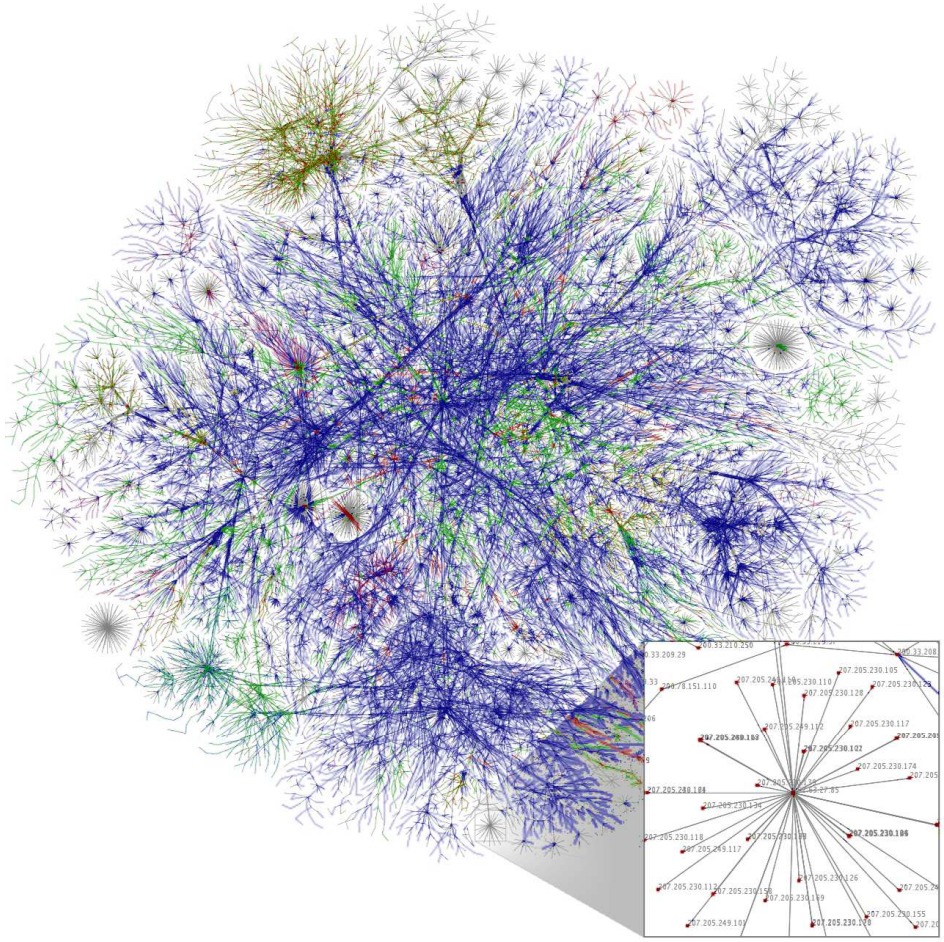


(All figures courtesy: Wikimedia.org)

More generally, finding a fixed subgraph of particular kind in a given graph,
or a routing problem in a graph.

Examples of Graphs

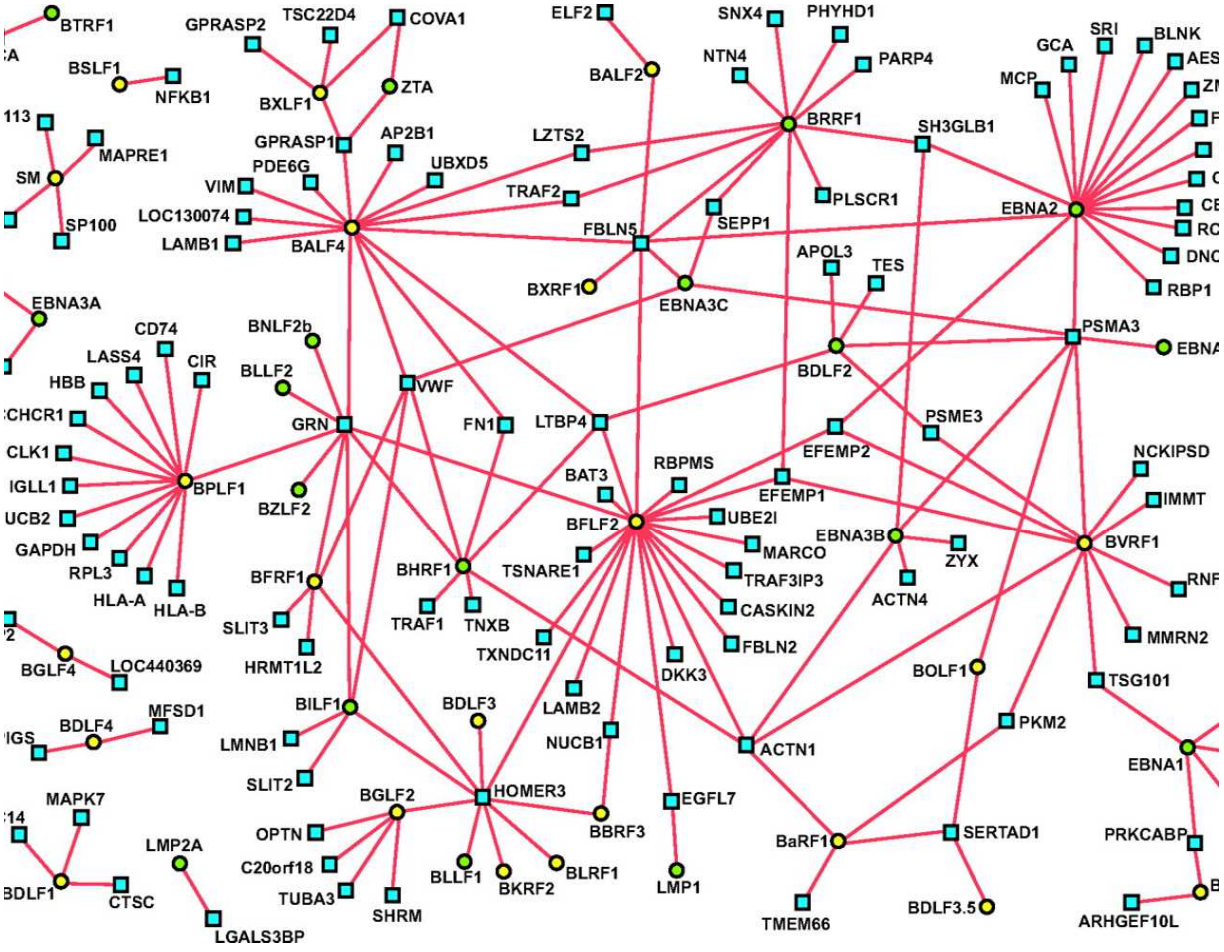
Vertices are internet routers, and edges are links between them.



(Courtesy: Wikimedia.org)

Examples of Graphs

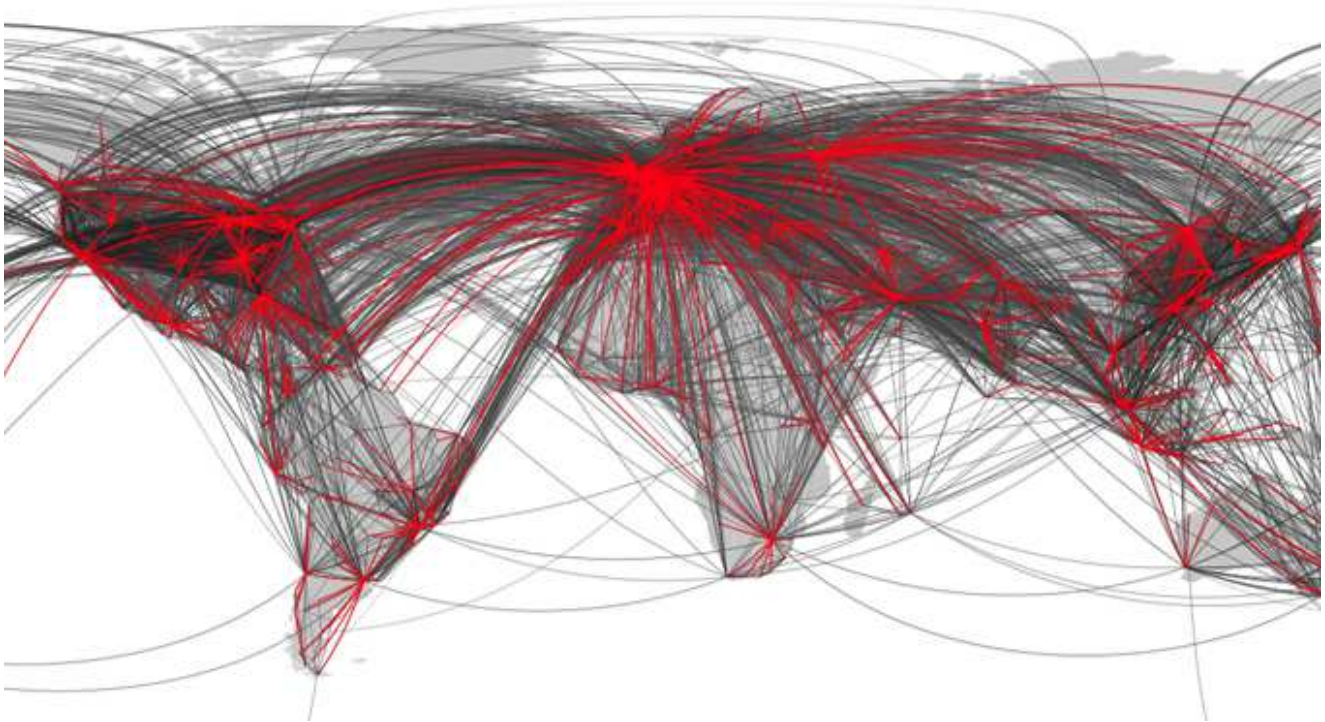
Vertices are various kinds of Herpes virus protein (circles) and human proteins (squares), and edges correspond to interactions between these proteins.



(Courtesy: PNAS.org)

Examples of Graphs

Vertices are cities around the world, and edges are airline connections between any two cities.



(Courtesy: Phys.org)

Examples of Graphs

Often the relationships are not fixed but change with time:

- Vertices are webpages and edges correspond to weblinks
- Vertices are highschool students and edges correspond to friendships
- Vertices are zebras and edges correspond to proximity/interaction
- Vertices are office workers and edges correspond to project teams
- and so on.

The General Problem

Analysis of longitudinal data of "social interactions"
to identify persistent patterns or substructures/ communities.

The General Problem

Analysis of longitudinal data of "social interactions"
to identify persistent patterns or substructures/ communities.

"Social Interactions" are represented as edges over a set of (fixed) vertices, the population under consideration.

Longitudinal data means that these edges are time dependent, interactions change as time goes by.

The General Problem

Traditionally, dynamic is made static:

1. Focus on one particular point in time.

Which time? How to incorporate the evolution of interactions?

The General Problem

Traditionally, dynamic is made static:

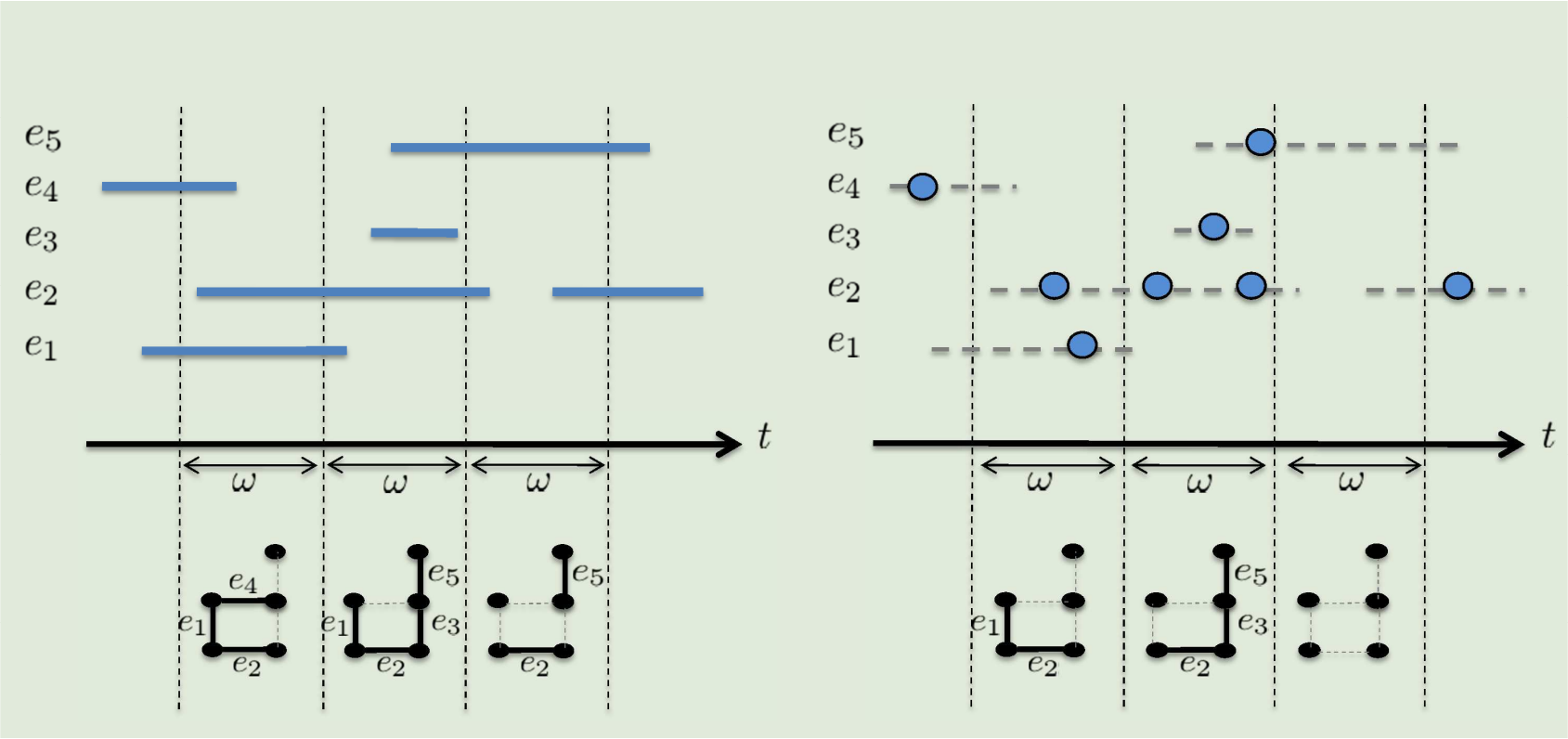
1. Focus on one particular point in time.

Which time? How to incorporate the evolution of interactions?

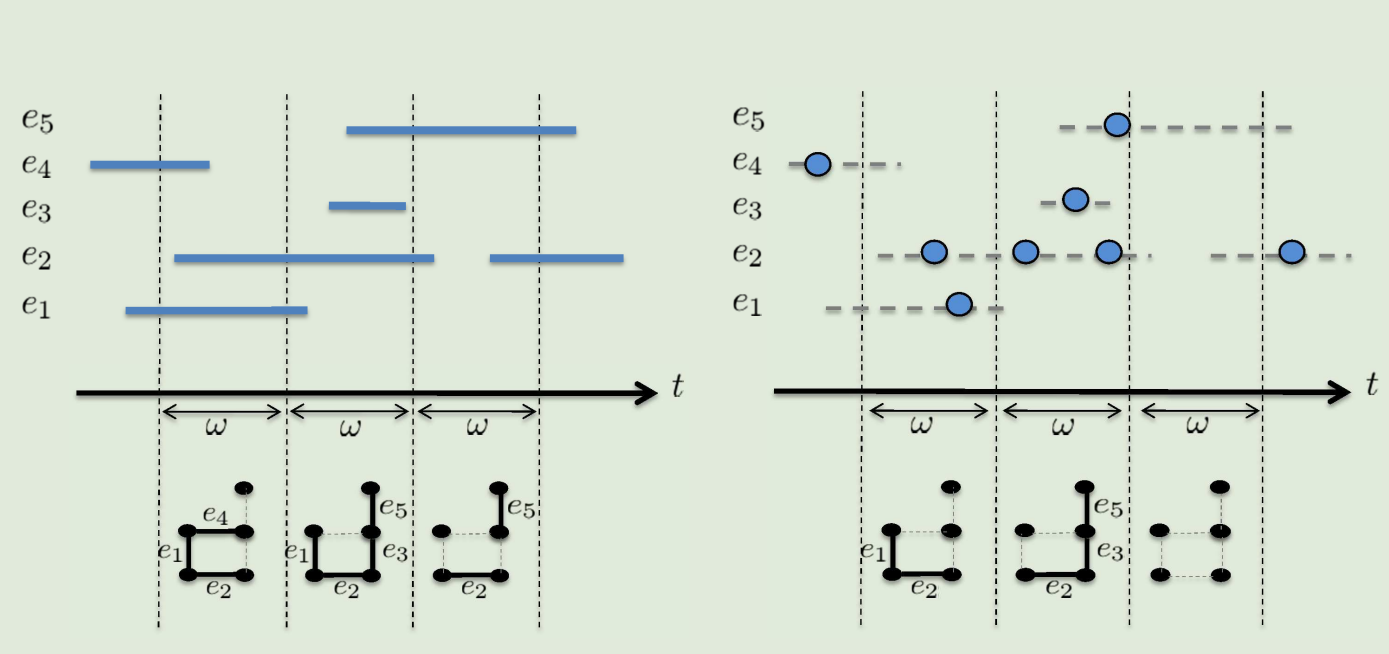
2. Aggregate the data into a single weighted graph.

One such weighted graph can arise from many sequences of such data.

Dynamic Graph Data



Dynamic Graph Data



Interval based interaction stream, for example friendships in a social network.

Instantaneous interaction, for example email communications.

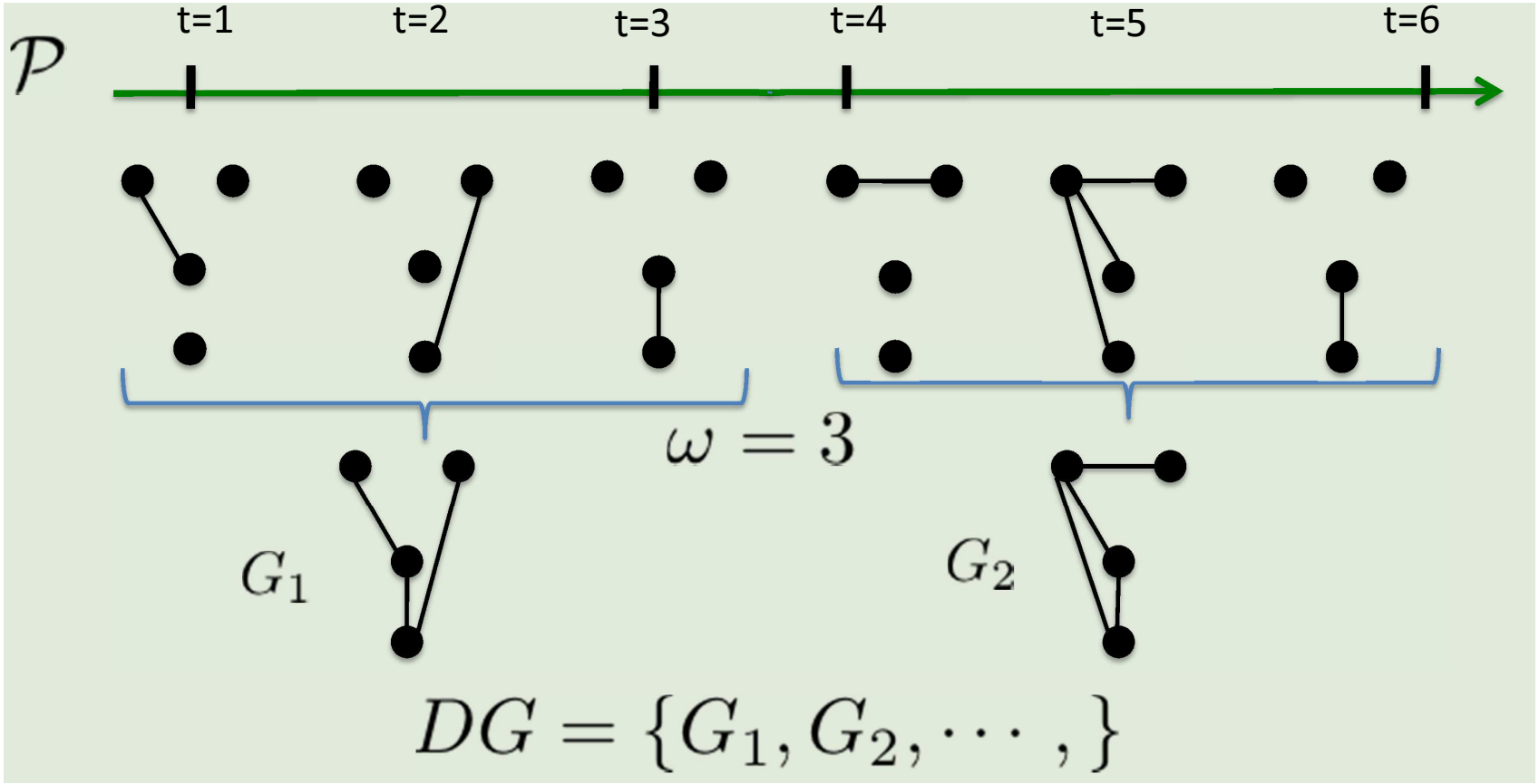
Dynamic Graph Data

Collected data comes from GPS sensors, digital recording of emails, or human observation of animals grooming: the instantaneous times at which the interactions were observed to be present.

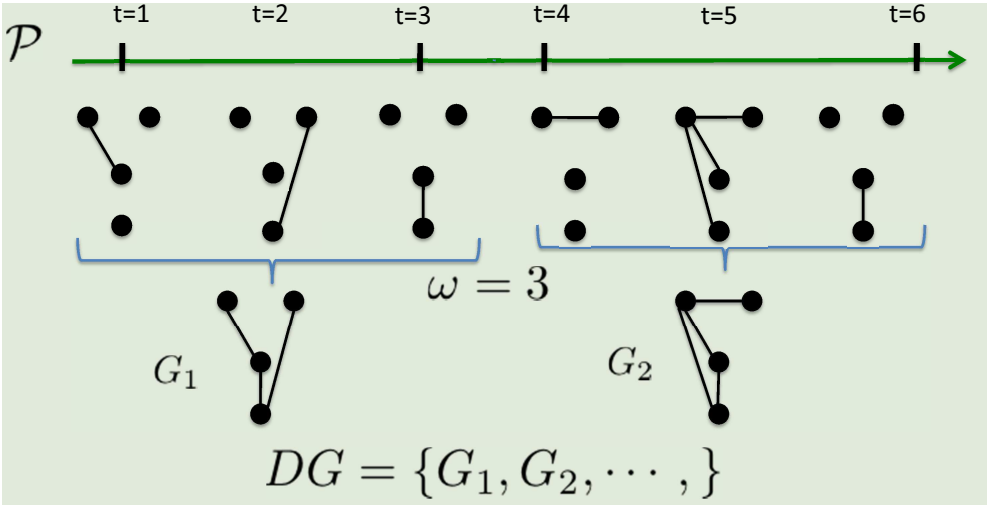
Temporal Errors: Data Collection/ Sampling error.

Topological Errors:
Representing continuous behavior discretely leads to missing interactions that should be present and recording spurious interactions that are not meaningful.

Temporal Scale of Dynamic Graphs



Temporal Scale of Dynamic Graphs

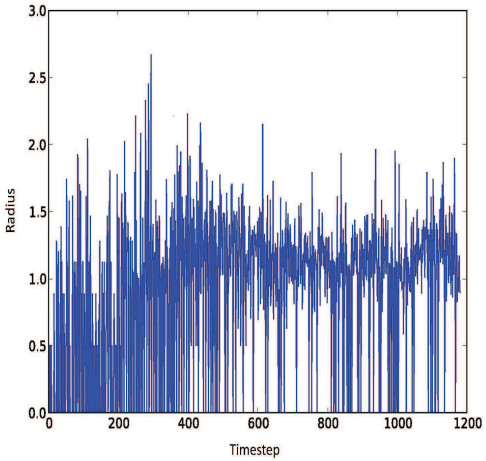


A dynamic Graph is a time series of Graph snapshots. Each snapshot represents a state of the system over an interval of time such as a minute, a day, or a year in the life of the system.

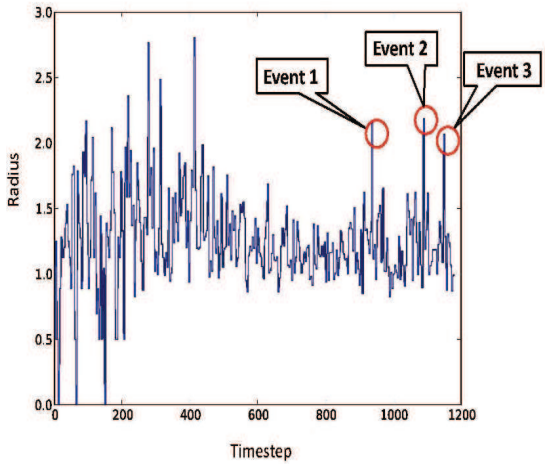
The duration of the snapshot represents the temporal scale of the dynamic Graph since all the interactions are lumped together discarding their order in time.

Temporal Scale of Dynamic Graphs

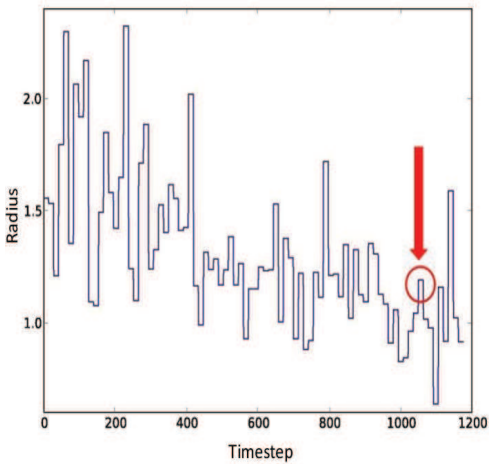
Online communications: Even though individual communications last only seconds or minutes, aggregation at the level of hours or days might be needed to find the correct timescale.



(a) $\omega = 1$ day



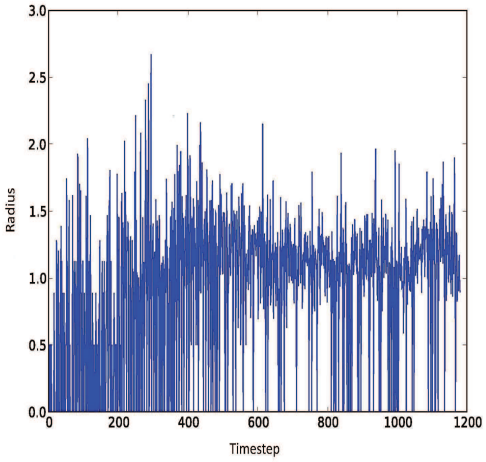
(b) $\omega = 5$ days



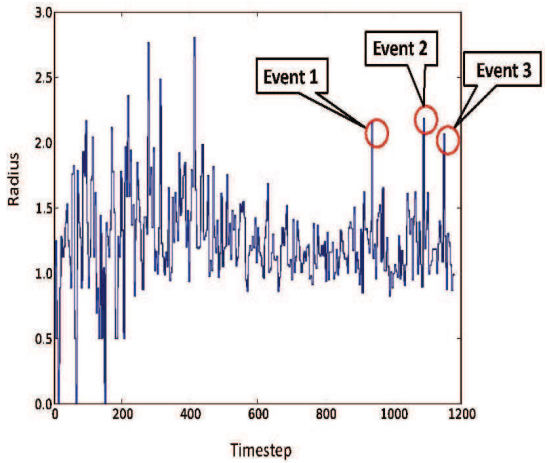
(c) $\omega = 12$ days

Enron Email is a publicly available dataset of e-mails sent between employees of the Enron corporation. Each email address represents a vertex and an email exchange represents an edge.

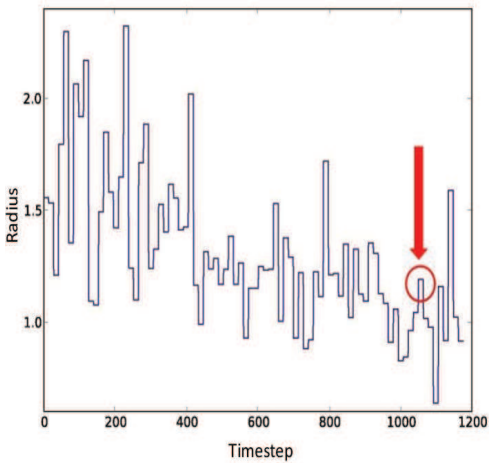
Temporal Scale of Dynamic Graphs



(a) $\omega = 1$ day



(b) $\omega = 5$ days



(c) $\omega = 12$ days

Enron Email Dataset.

Event 1 represents the time when Karl Rove sold off his energy stocks,

Event 2 represents the unsuccessful attempt of Dynegy to acquire the bankrupt Enron,

Event 3 represents the resignation of Enron's CEO.

Temporal Scale of Dynamic Graphs

Animal social interactions:

For example, grooming interactions of baboons usually have a temporal scale ranging from seconds to minutes,
mother to infant or peer to peer relationships have a scale extending over years,
an individual troop membership, splitting or formation of new troops extends from years to decades.

Temporal Scale of Dynamic Graphs

Human social interactions: Patterns of interaction of conversations, friendships, and kinship occupy different temporal scales.

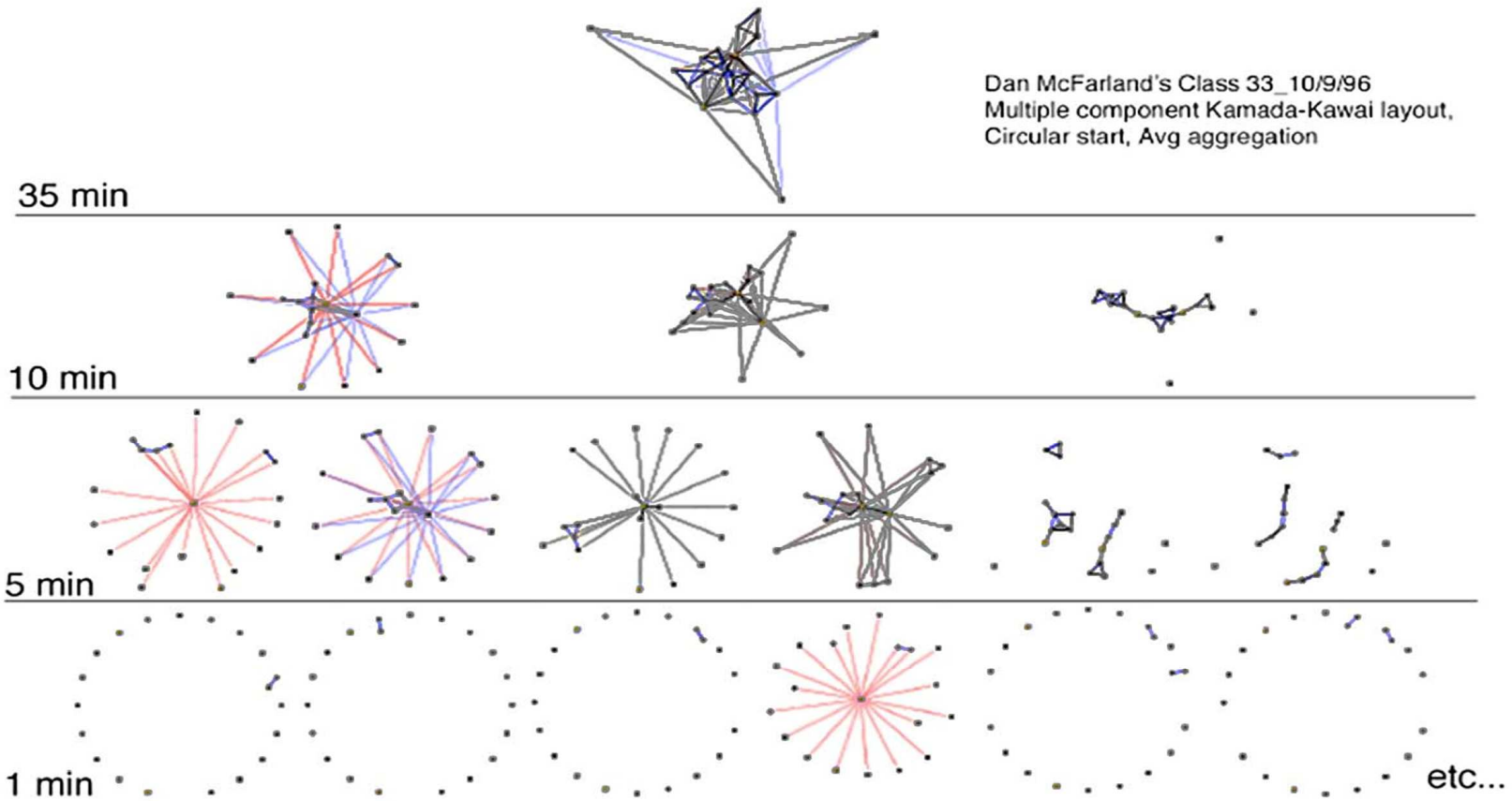
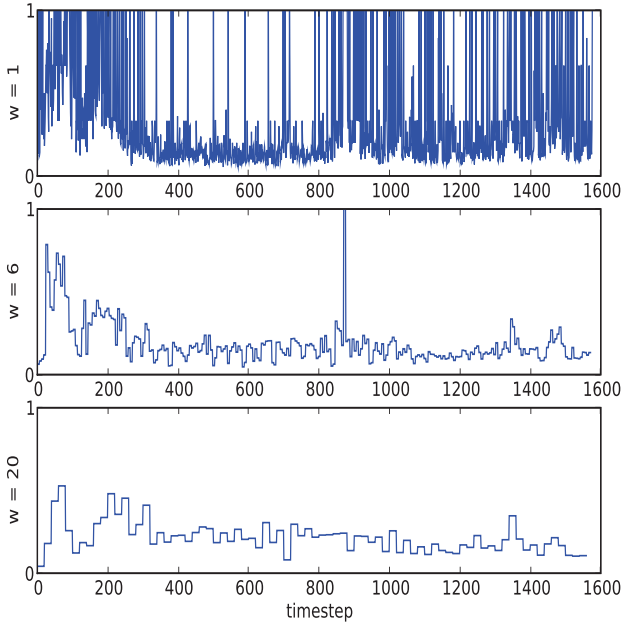
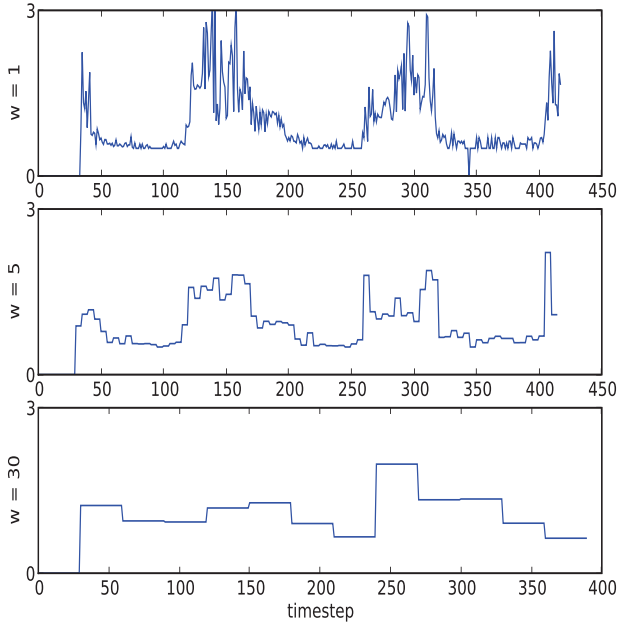


Figure 1. Interaction data from McFarland's classroom observations viewed at various levels of time aggregation from 35 minutes (one entire class period) to 1 minute (two to three turns of interaction).

Temporal Scale of Dynamic Graphs





Reality Mining network consists of social interactions among 90 MIT students and faculty over a nine month period with spatial proximity between people (through bluetooth connection) implying a social interaction.

Haggle Infocomm network consists of social interactions among attendees at an IEEE Infocom conference. There were 41 participants and the duration of the conference was 4 days.

Dynamic Graph

A temporal stream of edges is a sequence of edges (over a fixed vertex set $V = \{1, \ldots, N\}$) ordered by their time labels:

$$E = \{(ij, t) | ij \in V \times V, t \in [1, \ldots, T]\}$$

Dynamic Graph

Let $\mathcal{P}$ be a partition of the timeline $[1, \ldots, T]$:

$$\mathcal{P} = p_1, p_2, \ldots, p_k \;=\; [t_0, t_1), [t_1, t_2), \ldots, [t_k, T]$$

Dynamic Graph

Let $\mathcal{P}$ be a partition of the timeline $[1, \ldots, T]$:

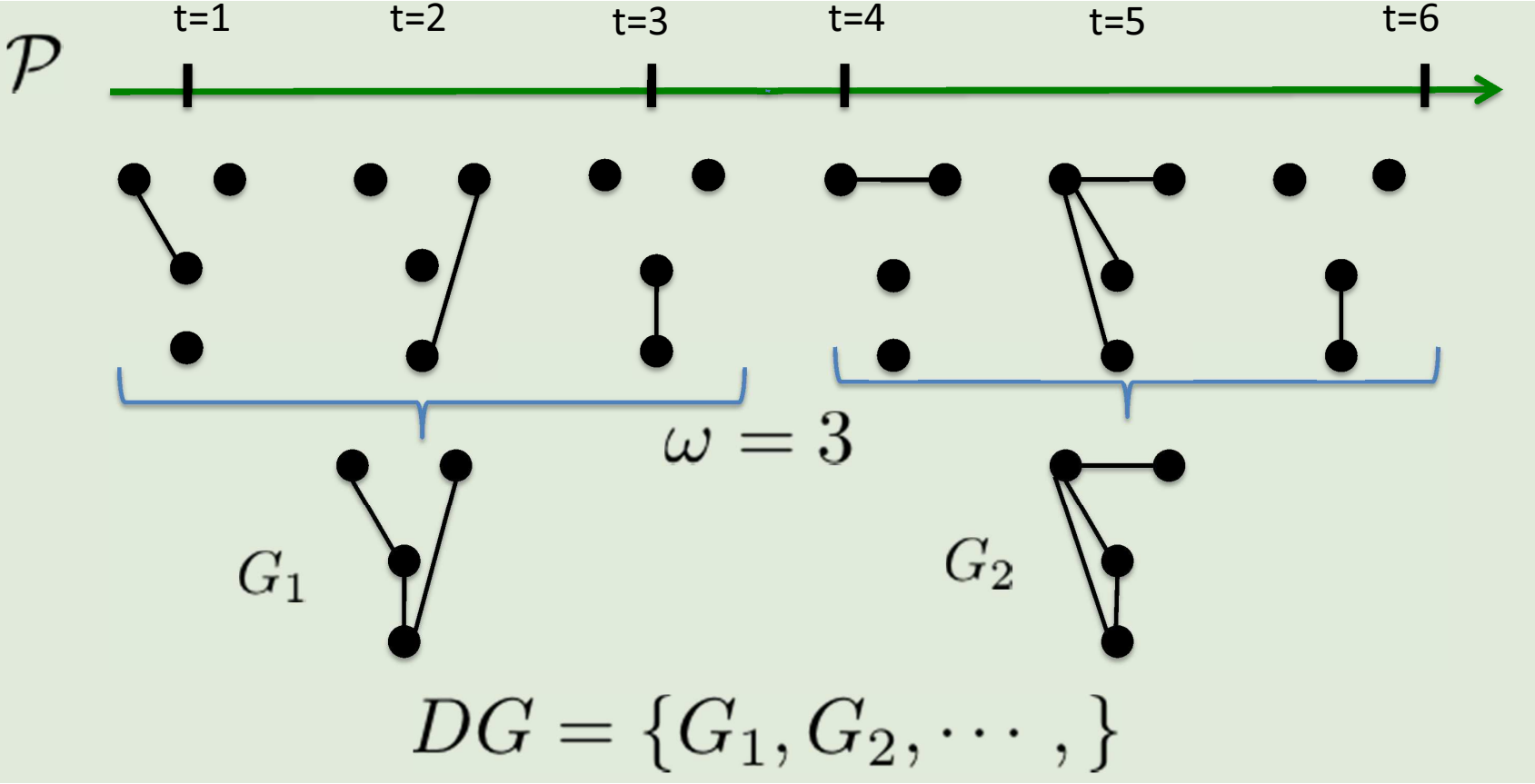
$$\mathcal{P} = p_1, p_2, \ldots, p_k \;=\; [t_0, t_1), [t_1, t_2), \ldots, [t_k, T]$$

A dynamic Graph is a sequence of graphs defined over the edge-stream E and a fixed partition $\mathcal{P}$ of $[T]$:

$$\langle G_1, G_2, \ldots, G_i, \ldots G_{|P|} \rangle$$

with $E(G_i) = \{(ij, t) | t \in p_i\}$ and
each G_i is associated with the ith interval p_i in $\mathcal{P}$.

Dynamic Graph



Some Related Work

Empirical Evidence:

Fourier Transform Analysis of Graph parameters

Clauset, A. and Eagle, N.: Persistence and periodicity in a dynamic proximity network, DIMACS 2007.

Dynamic Graph Visualization

Moody, J., McFarland, D., and Bender-deMoll, S.: Dynamic network visualization, American Journal of Sociology, 2005.

Empirical Analysis of Graph Parameters

Krings, G., Karsai, M., Bernharsson, S., Blondel, V. D., and Saramaki, J.: Effects of time window size and placement on the structure of aggregated networks, CoRR 2012.

Some Related Work

Heuristics:

Change detection in interaction streams

Sun, J., Faloutsos, C., Papadimitriou, S., and Yu, P. S.: Graphscope: parameter-free mining of large time-evolving graphs, Proc. 13th ACM SIGKDD, 2007.

Community detection in Biological Data

Berger-Wolf, T., Tantipathananandh, C., and Kempe, D.: Dynamic Community Identification, In: Link Mining: Models, Algorithms, and Applications, 2010.

Temporal scale detection via linear Graph functions

Sulo, R., Berger-Wolf, T., and Grossman, R.: Meaningful selection of temporal resolution for dynamic networks, Proc. 8th Workshop on Mining and Learning with Graphs, 2010.

Axioms

What properties should a temporal scale satisfy?

Motivated by axiomatic approaches to “Clustering”:

Impossibility result of Kleinberg 2002.

Quality-based axioms of Ackerman, Ben-David 2008.

Axioms

Q, a (quality) function that gives a numerical value to a particular partition of the timeline and the corresponding dynamic Graph indicating its "quality".

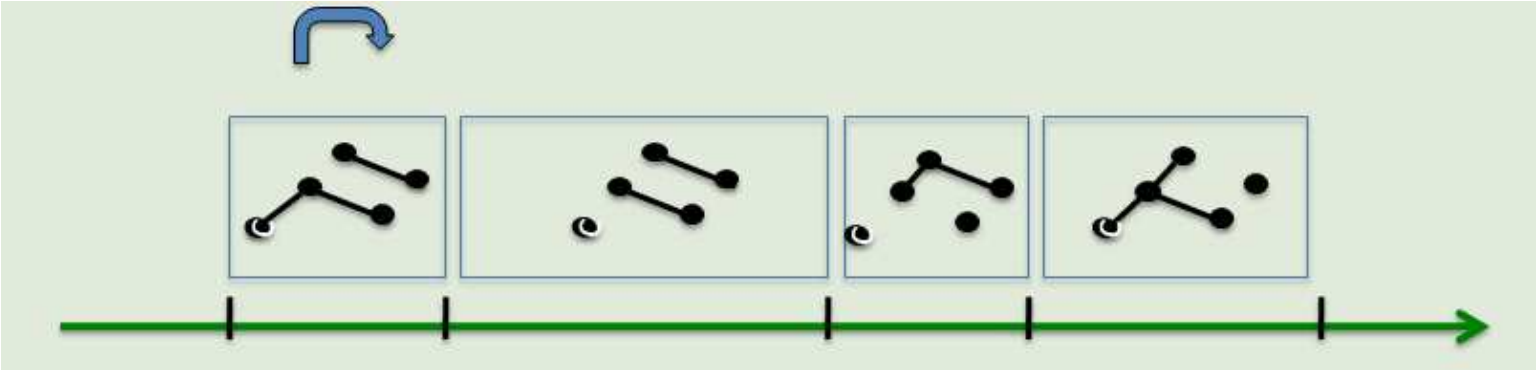
What properties must Q satisfy?

Axioms

Within Interval Order Invariance:

For an optimal partition, permutations of interactions within the same interval do not drastically change the quality of the dynamic graph.

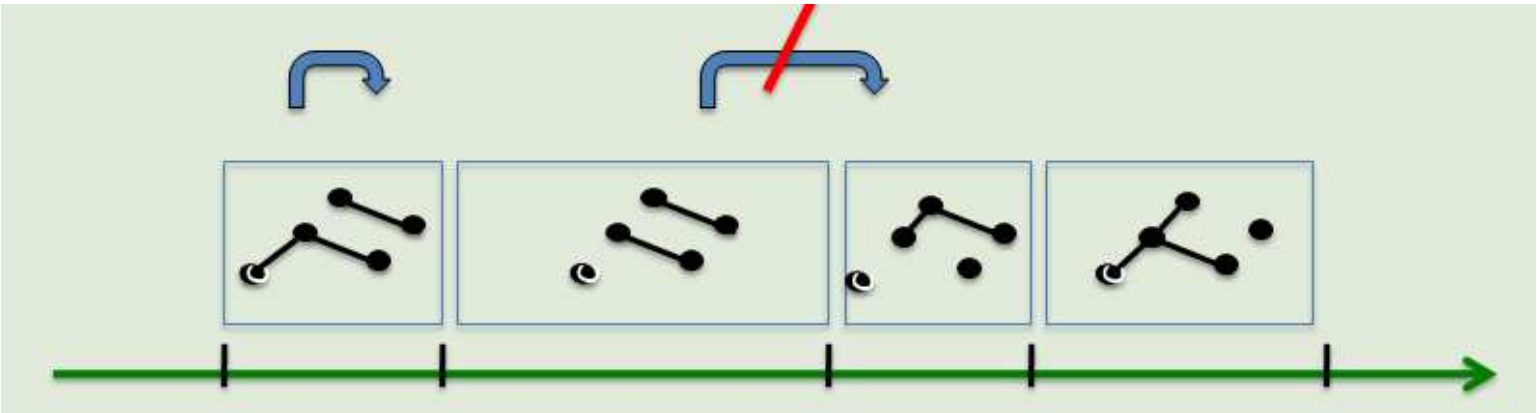
Some interactions are observed happening in a particular order might be an artifact of looking at them at too fine of a temporal resolution.



Axioms

Across Interval Order Criticality:

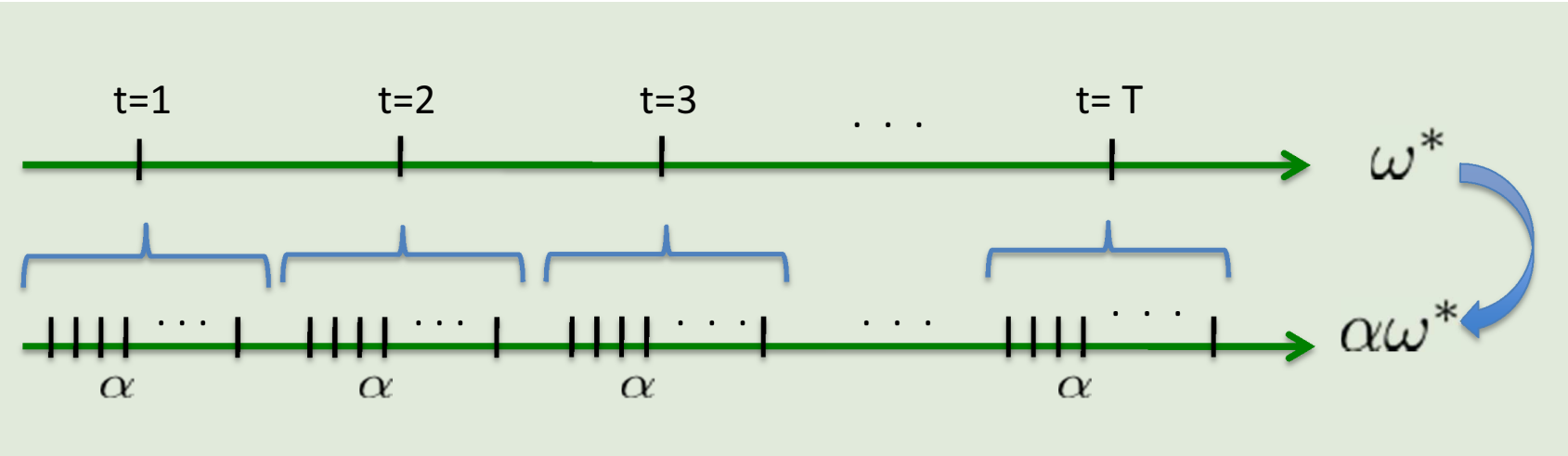
For an optimal partition, permutations of edges across different intervals will change the quality of the partition.



Axioms

Measure Unit Invariance:

Uniform scaling of the oversampling factor does not change the quality of the dynamic Graph.



Axioms

Constant Stream:

The constant stream (same set of edges at each moment of time) has no time scale, the optimal partition is the whole timeline.

Axioms

Stream with no Temporal Scale:

The quality function is the same for any partition of the stream with no temporal scale, a temporal version of the Erdos-Renyi random graph (noise).

Axioms

Temporal Shift Invariance:

A shift of the time line of a temporal stream, does not drastically change the quality of the dynamic Graph. The optimal partition of the stream is independent of the time line's starting point.

A Persistence Based Approach

Interactions observed fleetingly are often not interesting and they usually indicate that the data collection process is noisy.

Interactions that persist for a while, truly represent what is more essential for the underlying system.

What is, then, the “right” temporal scale that can capture the persistence of structure in time, while smoothing out temporal and topological noise?

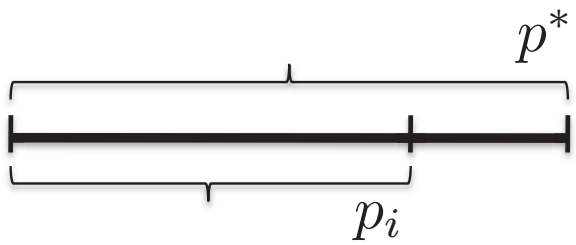
A Persistence Based Approach

Instead of a global quality function (for the the whole partition of the timeline), we will use a local quality function, q, for intervals within the partition.

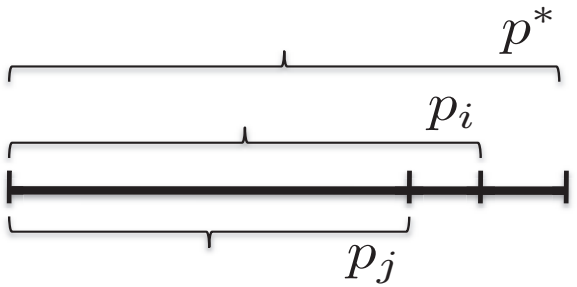
Axioms:

(a) Internal Consistency: $q(p_i) \approx q(p^*)$

(b) Local Monotonicity: $q(p_i) \geq q(p_j)$



(a)



(b)

A Persistence Based Approach

Local temporal persistence is measured via changes in edge frequency values as a proxy.

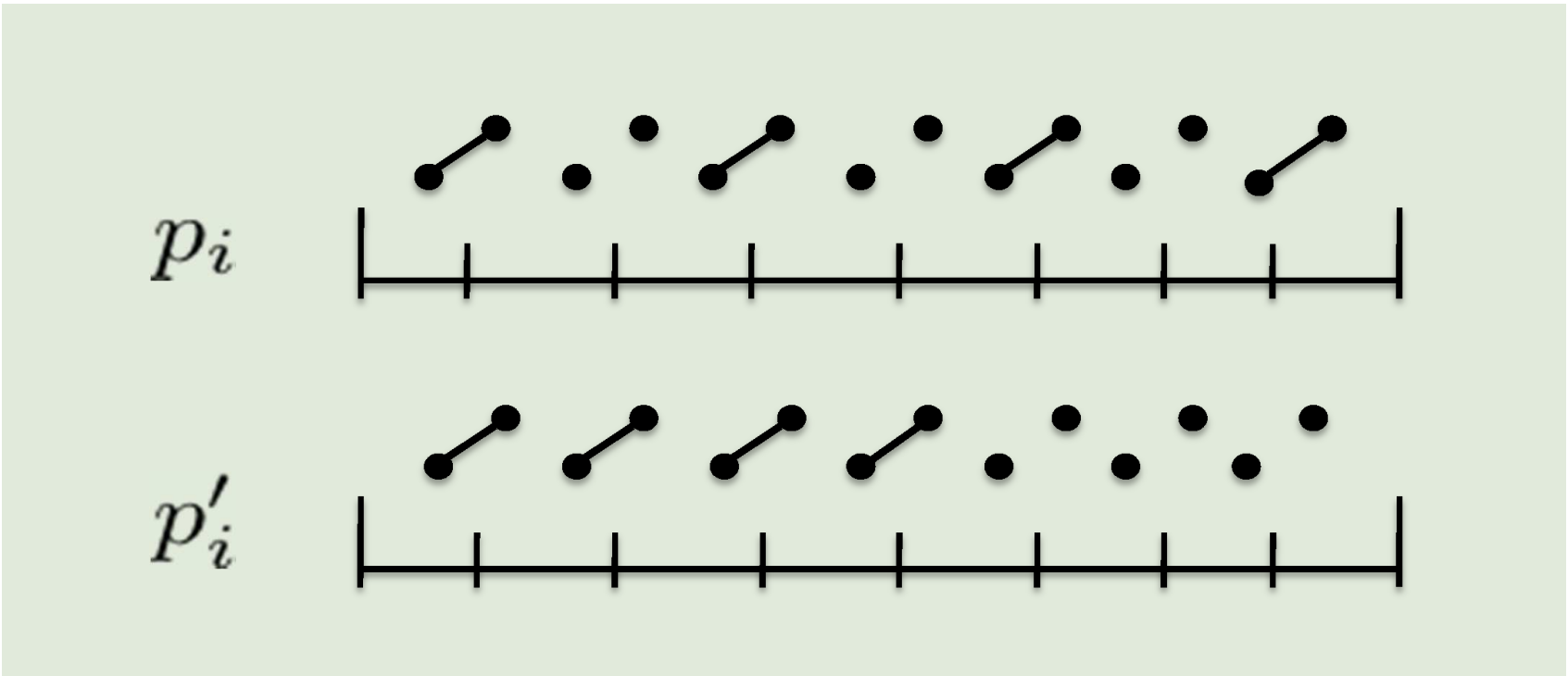
The Graph structure that persists over time is a manifestation of more or less the same set of edges occurring consistently.

At greater computational cost, “edge” can be replaced by any fixed substructure.

Using edges minimizes any assumptions about any substructures in the data.

A Persistence Based Approach

High persistence of Graph structure implies persistence of edge frequency values, but the converse is not necessarily true.



A Persistence Based Approach

$freq(p)$ is the frequency vector (of length $|E|$) representing the number of times each edge occurs in the interval p of the partition.

$fd(p_i)$ is the difference (via an l_p norm) between $freq(p_i)$ and $freq(p_{i+1})$, frequency vectors of consecutive intervals - not disjoint, overlap is controlled by local parameter w.

A Persistence Based Approach

Let *LM* be the set of *local* maxima:

$$LM = \{i : fd(i) \geq fd(j), i - r \leq j \leq i + r\}$$

where r is the radius for locality.

A Persistence Based Approach

Two types of intervals (l, r) whose quality we want to capture:

Type 1: $(l, r) \cap LM = \emptyset$. There are no local maxima inside interval (l, r).

The quality function q_1 of Type 1 intervals:

$$q_1 = \frac{\min\{fd(l), fd(r)\} - \min\{fd(x) : l < x < r\}}{r - l}.$$

A rectangle with left side $x = l$, right side $x = r$, top $y = \min\{fd(l), fd(r)\}$, and bottom $y = \min\{fd(x) : l < x < r\}$. q_1 is the slope of its diagonal.

A Persistence Based Approach

Type 2: $(l, r) \cap LM \neq \emptyset$. There are local maxima inside interval (l, r).

Let $m \in LM$ be the value in (l, r) such that $fd(m)$ is maximized. The quality function q_2 of Type 2 intervals:

$$q_2 := \frac{\min\{fd(l), fd(r)\} - fd(m)}{r - l}.$$

A rectangle with left side at $x = l$, right side at $x = r$, bottom at $y = fd(m)$ and top at $y = \min\{fd(l), fd(r)\}$.
Intuitively, when this box is deeper, we have a better interval.

Algorithm

1. Generate potential breakpoints using local maxima:

(i) Compute Type 1 Intervals

(ii) Compute Type 2 Intervals

Algorithm

2. Synchronize Type 1 and Type 2 intervals to generate a partition:

Algorithm

2. Synchronize Type 1 and Type 2 intervals to generate a partition:

(i) Take the union of Type 1 and Type 2 intervals and their corresponding q values.

Algorithm

2. Synchronize Type 1 and Type 2 intervals to generate a partition:

(ii) Sort the intervals by their q-values in non-increasing order, with ties broken arbitrarily.

Algorithm

2. Synchronize Type 1 and Type 2 intervals to generate a partition:

(iii) Initialize the set of breakpoints $B := \emptyset$.

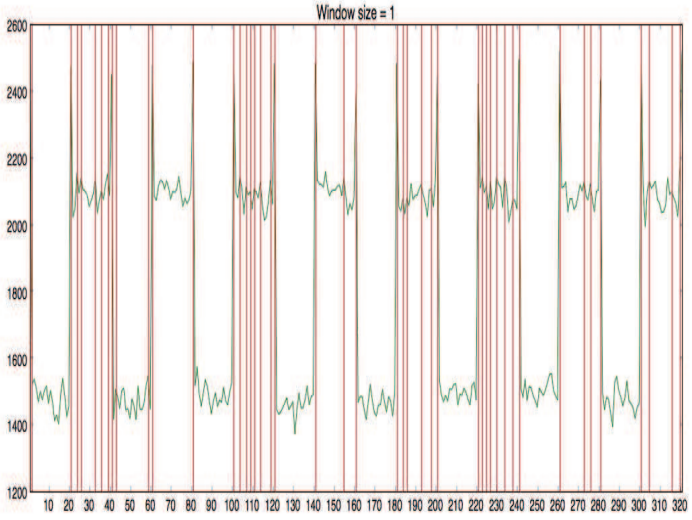
Iterate: Starting with the interval with the highest quality value (either q_1 or q_2), add the endpoints of the corresponding interval. Let $[l, r]$ be the next unprocessed interval. If the endpoints of the unprocessed interval fall inside any of the intervals already added to B, ignore the interval and move to the next unprocessed interval.

Algorithm

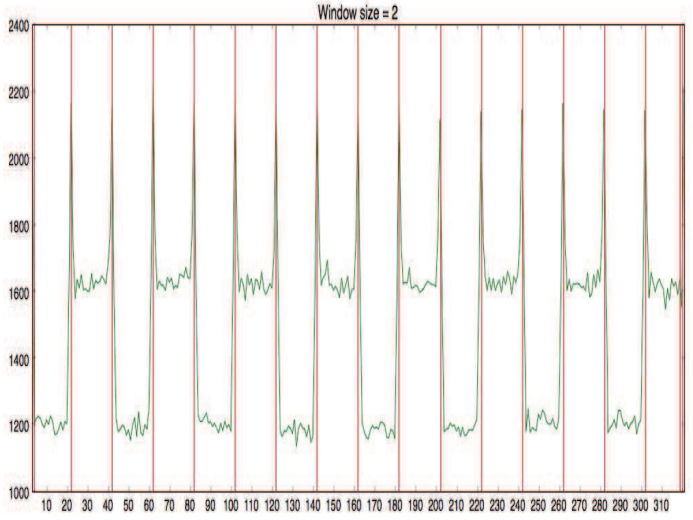
2. Synchronize Type 1 and Type 2 intervals to generate a partition:

(iv) When the procedure quits: if $B = \{b_1, \ldots, b_k\}$ with $b_1 < \ldots < b_k$, then our final answer is the set of intervals $[0, b_1), [b_1, b_2), \ldots, [b_k, T]$.

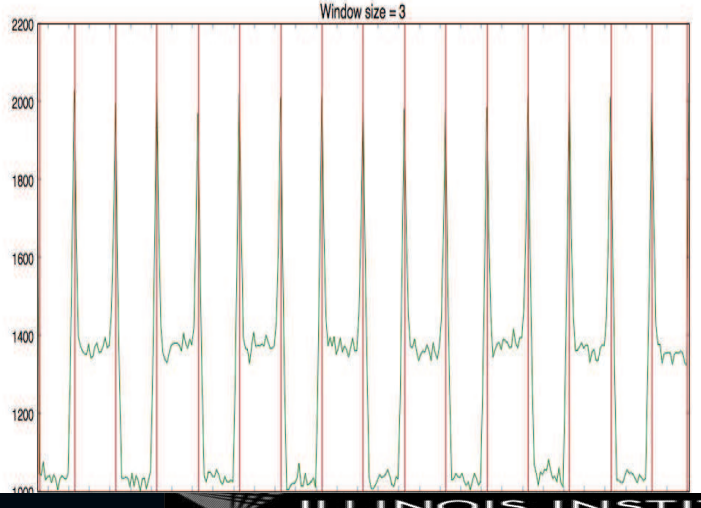
Computational Results



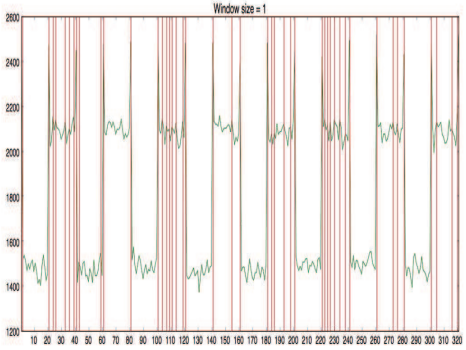
(a) $\omega = 1$



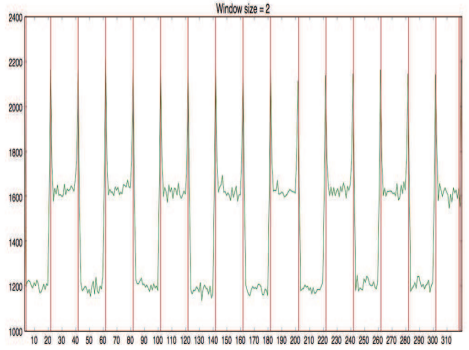
(b) $\omega = 2$



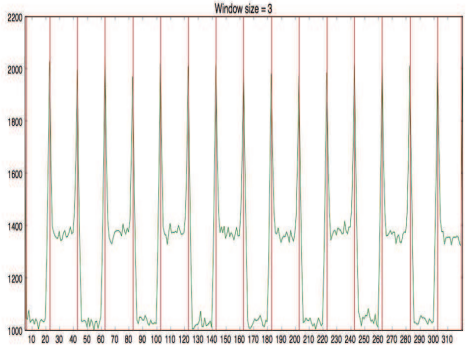
Computational Results



(a) $\omega = 1$



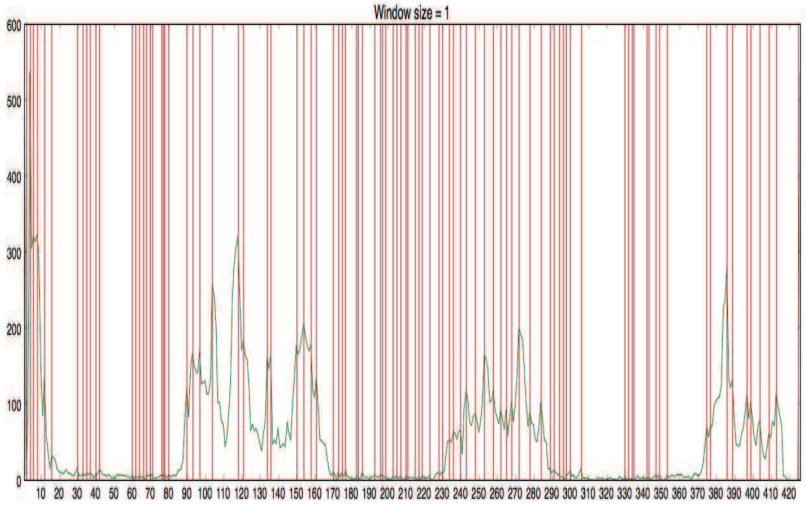
(b) $\omega = 2$



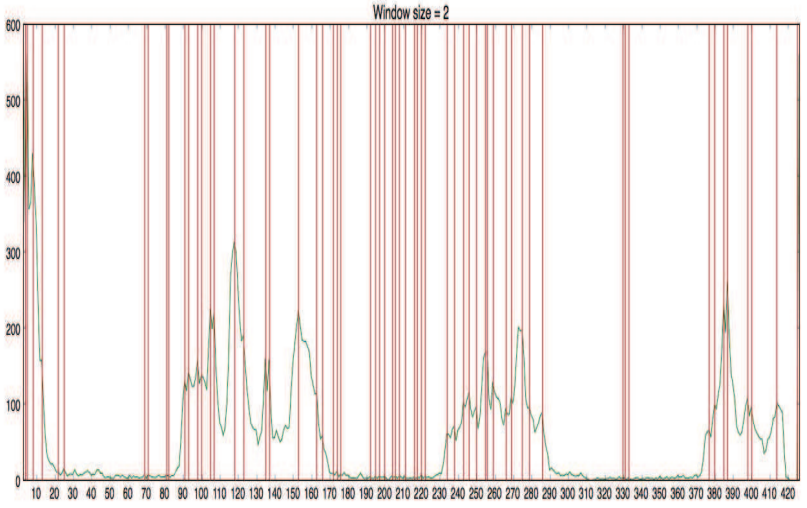
(c) $\omega = 3$

Synthetic data generated by using two alternating probability distributions: the beta distribution and gaussian distribution (behaves like noise) - alternating every 20 steps. Captured by the Algorithm.

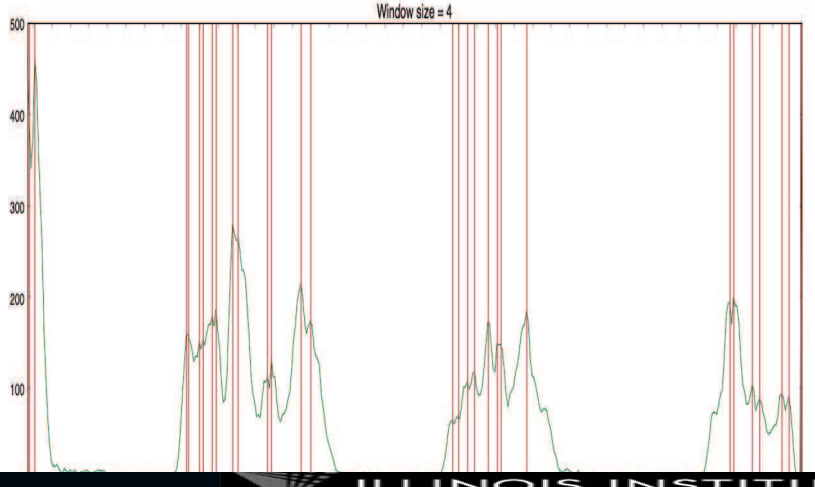
Computational Results



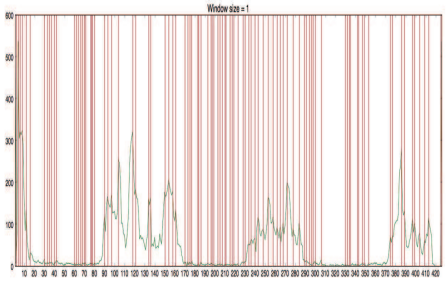
(a) $\omega = 1$



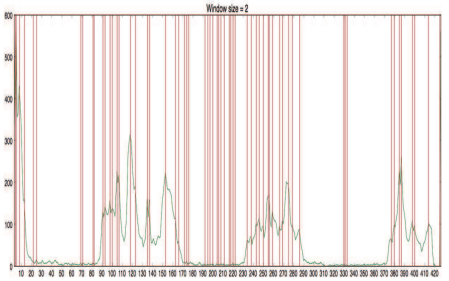
(b) $\omega = 2$



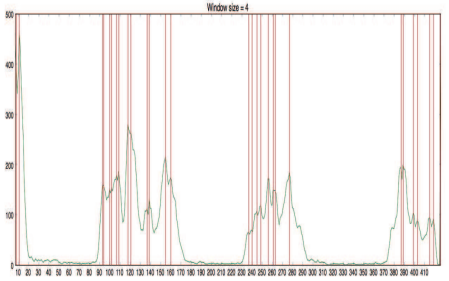
Computational Results



(a) $\omega = 1$



(b) $\omega = 2$



(c) $\omega = 4$

Haggle (IEEE conference) data stream over a 4 day period: For $w = 4$ (40 minute intervals), we see a clear separation between the day and night frequency patterns, some of the finer partitions correspond to intervals of length 20 minutes, 30 minutes, 50 minutes (talk periods) and about 3-4 hours (morning/afternoon sessions).

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Find a global quality function that captures “many” of the axioms.

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Find a global quality function that captures “many” of the axioms.

Show that the solution of our algorithm satisfies “some” of the global axioms.

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Find a global quality function that captures “many” of the axioms.

Show that the solution of our algorithm satisfies “some” of the global axioms.

In any algorithm, how can we determine the threshold value of window size (w) beyond which temporal scaling is meaningless/ useless?

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Find a global quality function that captures “many” of the axioms.

Show that the solution of our algorithm satisfies “some” of the global axioms.

In any algorithm, how can we determine the threshold value of window size (w) beyond which temporal scaling is meaningless/ useless?

How to determine best-fit subgraph within each interval in order to minimize topological errors?

Future Work/ Open Questions

Conjecture: There is no global quality function that satisfies all the axioms.

Find a global quality function that captures “many” of the axioms.

Show that the solution of our algorithm satisfies “some” of the global axioms.

In any algorithm, how can we determine the threshold value of window size (w) beyond which temporal scaling is meaningless/ useless?

How to determine best-fit subgraph within each interval in order to minimize topological errors?

New Algorithms. New applications.