Dynamic Pattern Detection with Connectivity and Temporal Consistency Constraints

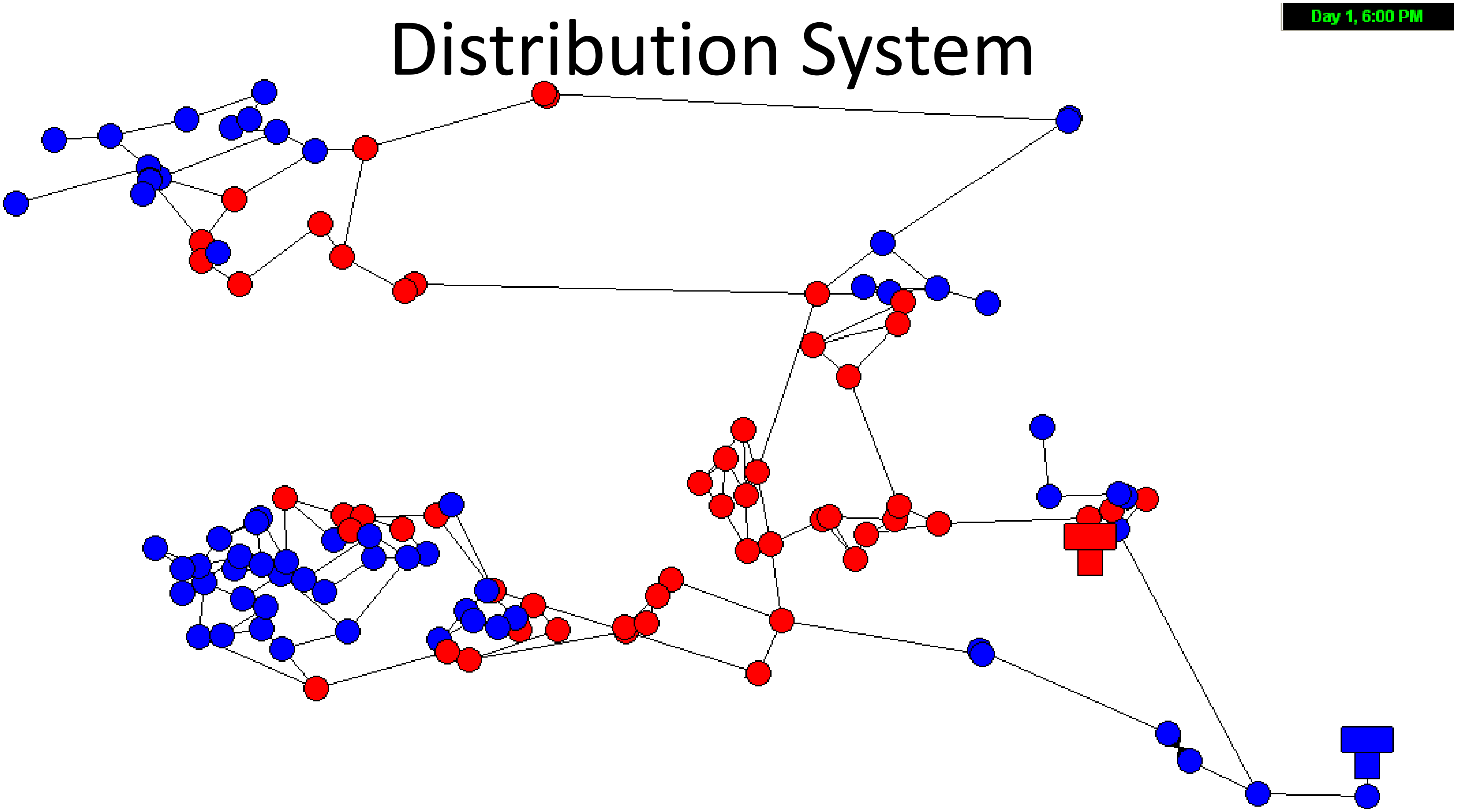
Skyler Speakman & Daniel B. Neill
Event and Pattern Detection Lab
Heinz College, Carnegie Mellon University

This work was partially supported by NSF grants:
IIS-0916345, IIS-0911032, and IIS-0953330

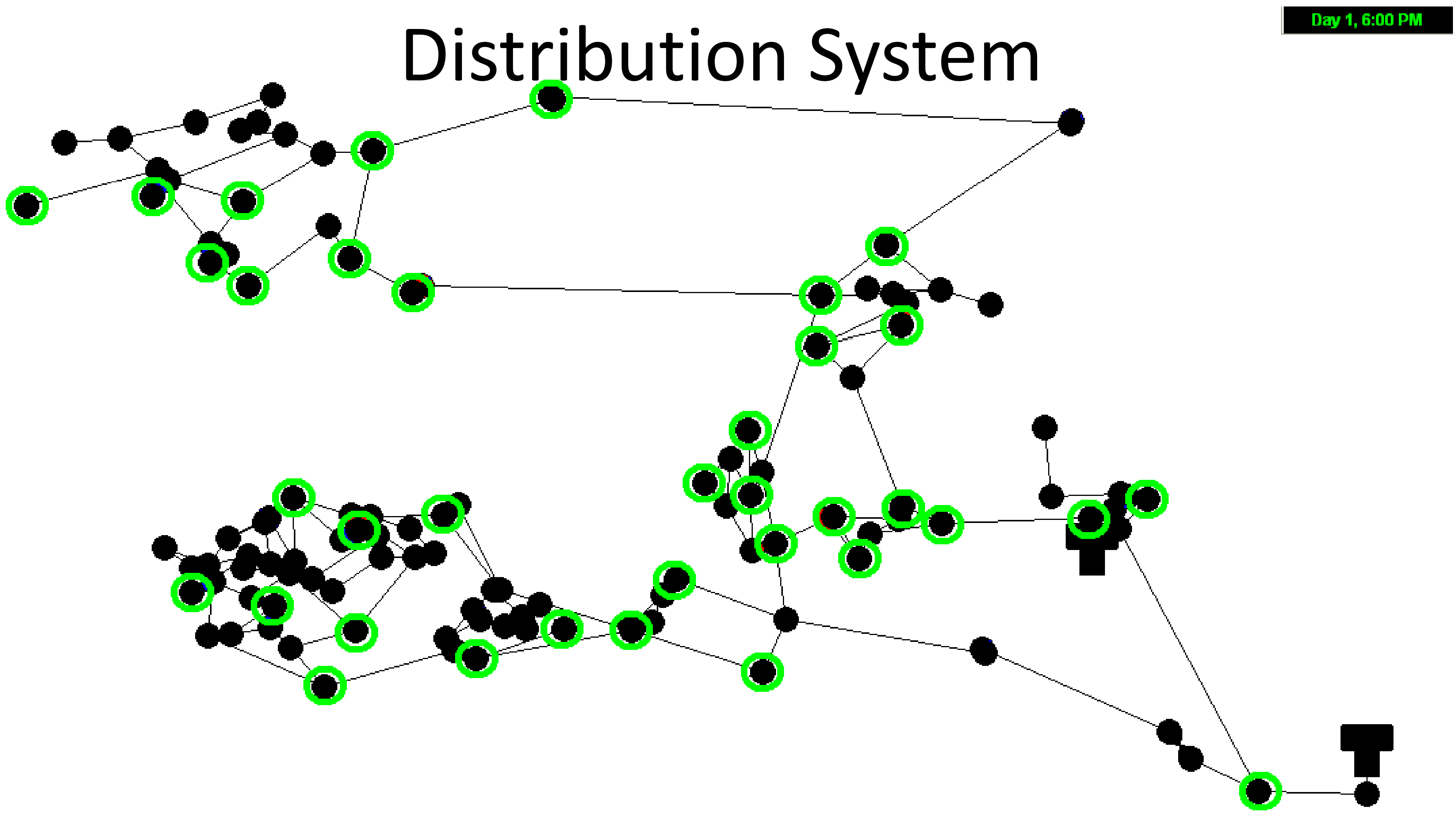
Event & Pattern Detection

National Science Foundation
WHERE DISCOVERIES BEGIN

Spreading Contaminants in a Water Distribution System



Spreading Contaminants in a Water Distribution System

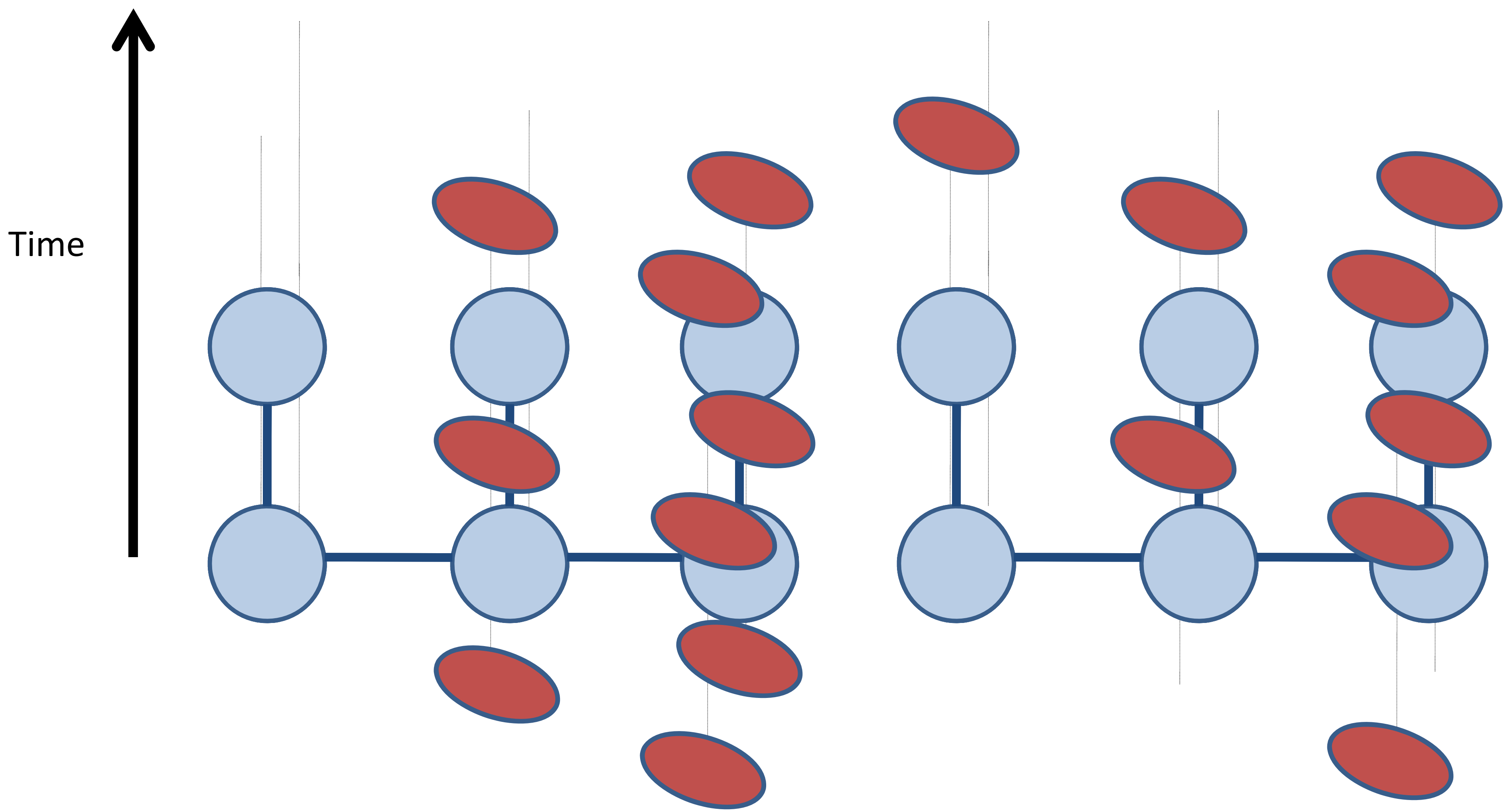


Static Pattern

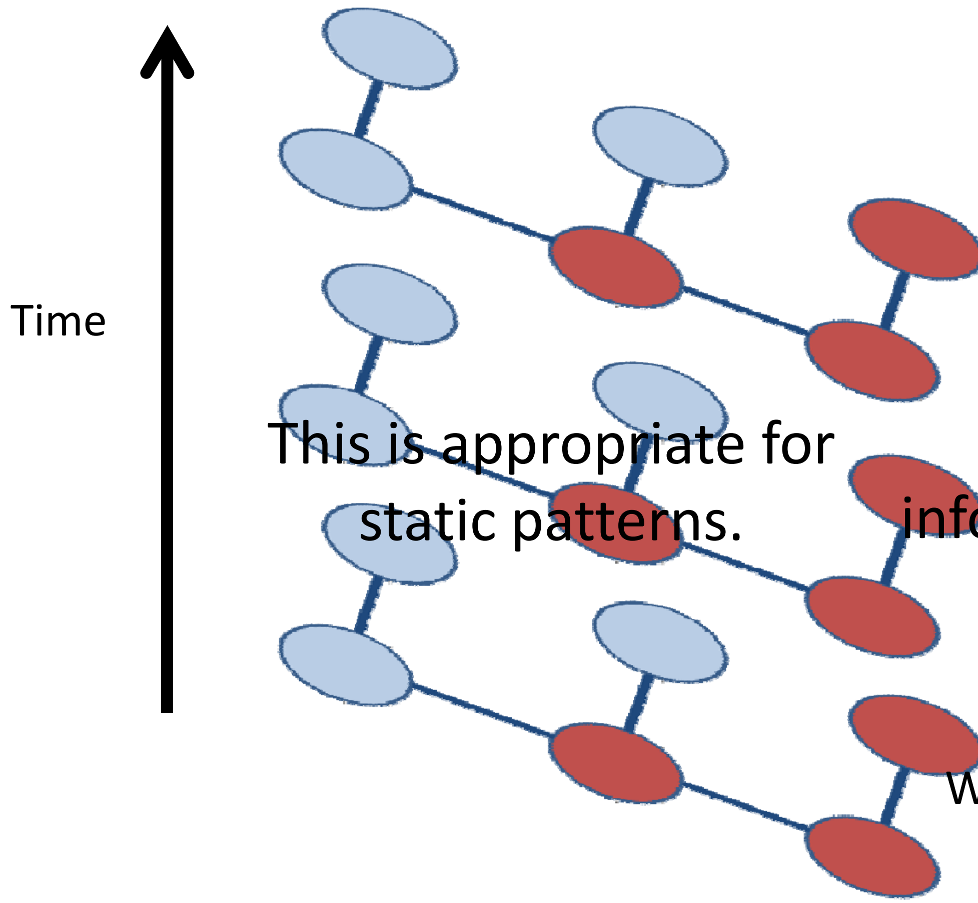
The subset of affected nodes
does not change over time

Dynamic Pattern

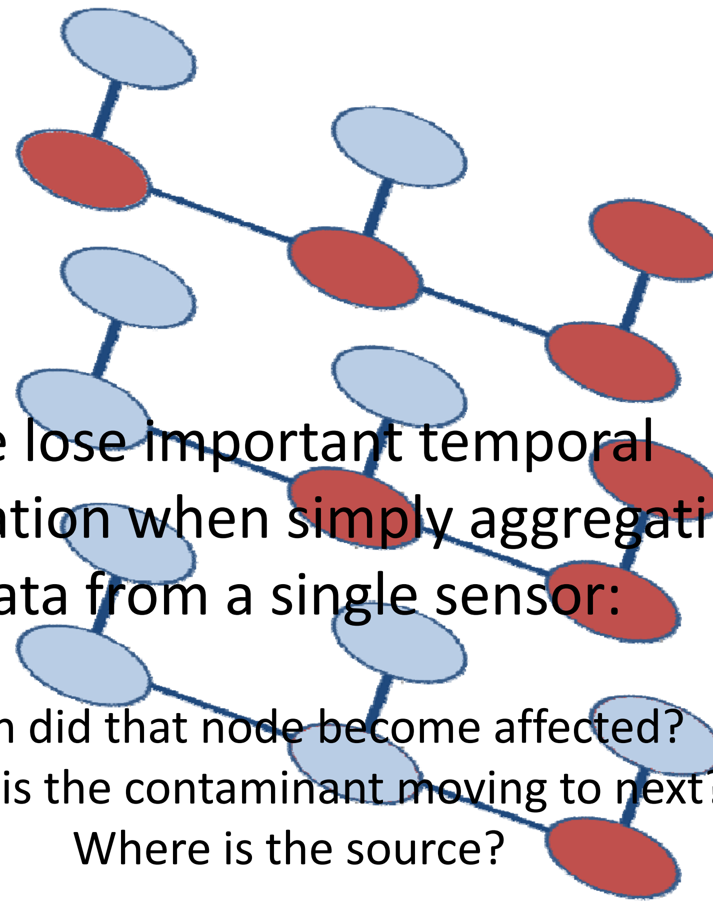
The subset(s) of affected
nodes ***change*** over time



Temporal Information: Aggregating



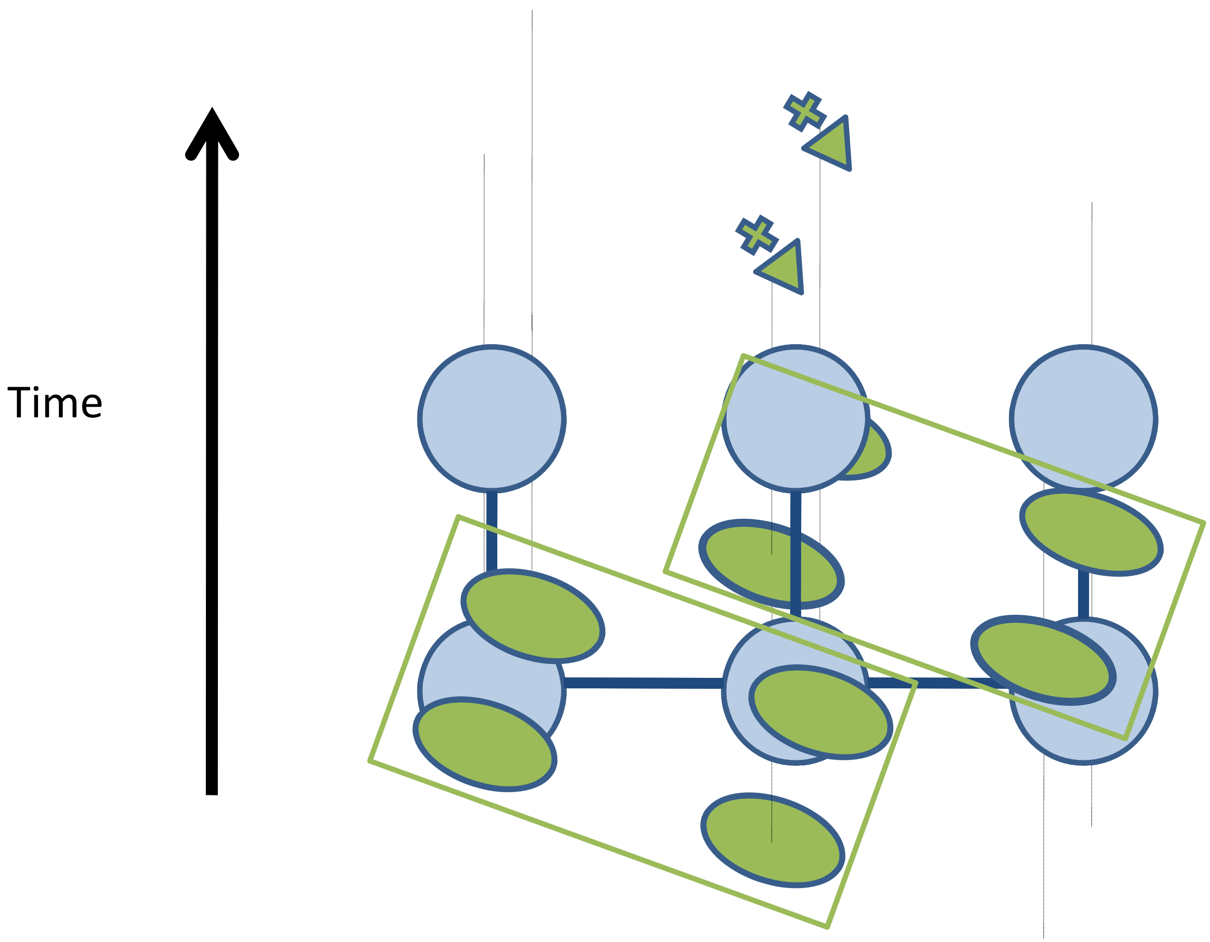
This is appropriate for static patterns.



We lose important temporal information when simply aggregating data from a single sensor:

When did that node become affected?
Where is the contaminant moving to next?
Where is the source?

Temporal Consistency



Goals of this Presentation

Provide a framework for incorporating
"soft constraints"
(i.e. temporal consistency)
without violating the properties that allow for the efficient search over the network

Describe properties of the ***expectation-based binomial*** scoring function which is appropriate when dealing with binary sensor data

Show empirical results which demonstrate the utility of incorporating soft constraints when detecting ***dynamic patterns*** such as contaminant plumes in a water distribution system

Pattern Detection as a Search Over Subsets

Pattern Detection can be framed as a search over subsets of the data with the goal of finding the subset which best matches a probabilistically modeled pattern.

This “match” is quantified by a scoring function, typically a ***likelihood ratio***.

Computational Problems: Infeasible to perform exhaustive search for more than 30 records

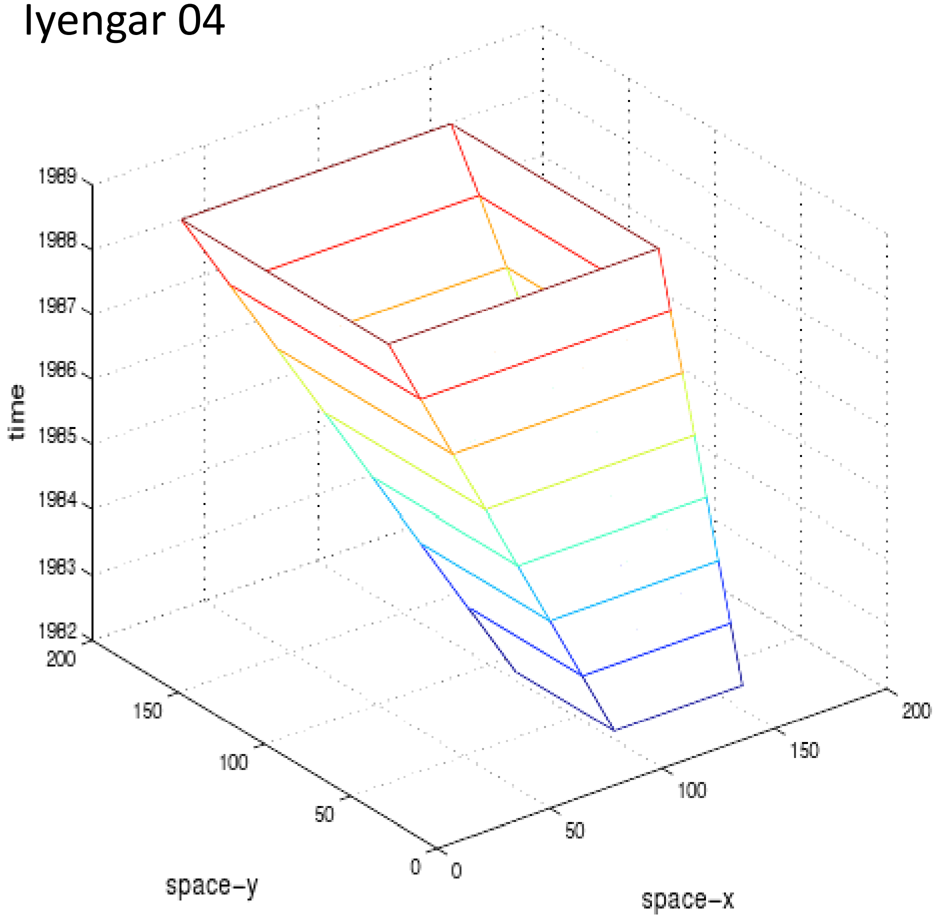
Linear-time Subset Scanning (LTSS)
property allows for exact, efficient identification of “highest scoring” subset without an exhaustive search

Neill 2010

GraphScan applied LTSS to only consider ***connected*** subsets. Increases power to detect patterns that affect a subgraph of a larger network.

Speakman & Neill 2010

Iyengar 04



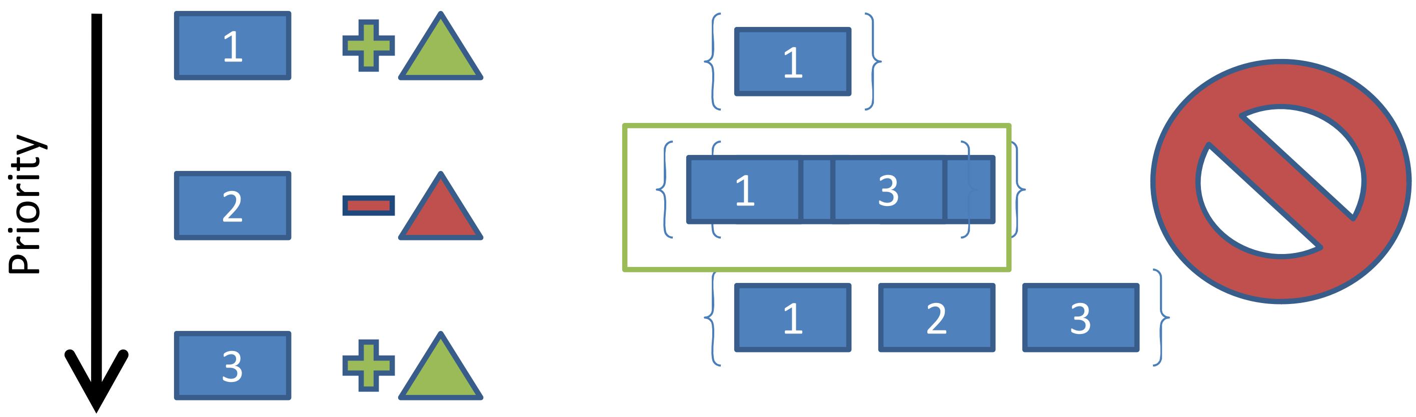
Use geometric shapes (truncated cones) to aggregate temporal information

Assumes linear growth or movement of the pattern

Potential shapes to search over grow exponentially and therefore relies on a heuristic to approximate best subset

(Neill, 2010)

The highest scoring subset is guaranteed to be one of the following subsets



Decreases the search space from 2^N to N

Naively altering the scoring function to enforce soft constraints violates LTSS

PROBLEM:	We must alter the ***scoring function*** instead of restricting the search space.

When applied directly, these constraints ***violate*** the ***LTSS property*** of the scoring function and make exact, efficient search ***impossible***.

SOLUTION:	Interpret the scoring function as a ***sum*** of ***contributions*** from each record in the subset.

Maximizing the scoring function is then equivalent to selecting all records that are making a ***positive contribution***.

INSIGHT:	When treated as an additive function, ***further terms*** may be introduced without interfering with the maximization step.

Enforcing Soft Constraints with LTSS

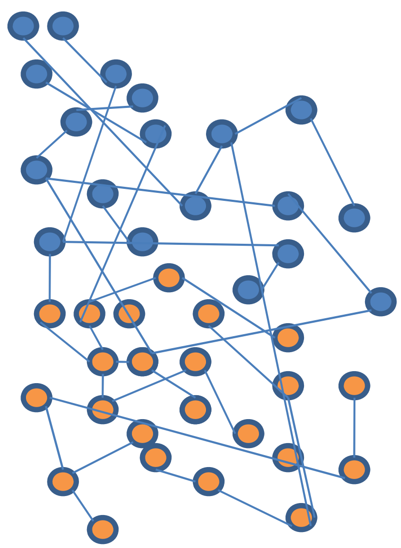
Scoring Function

$$F(S) = \log \frac{P(Data \mid H_1(S))}{P(Data \mid H_0)}$$

$$H_0 : c_i \sim Bernoulli(p_0)$$

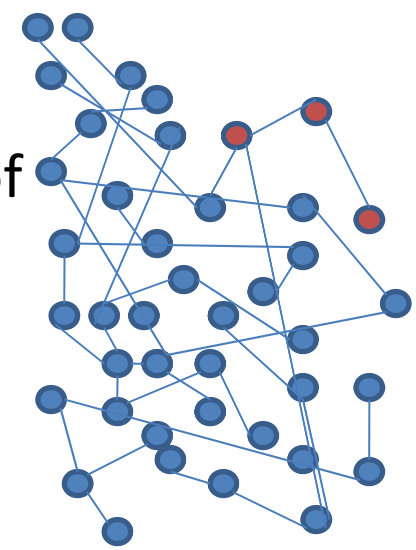
$$H_1 : c_i \sim Bernoulli(qp_0) \qquad 1 < q < \frac{1}{p_0}$$

$$F(S) = \max_q \log \frac{P(Data \mid H_1(S))}{P(Data \mid H_0)}$$



Large number nodes with a moderate risk

Small number of nodes with a high risk



Adding Soft Constraints to the Scoring Function

$$F(S) + \sum_{s_i \in S} \Delta_i$$

SOLUTION: Interpret the scoring function as a ***sum*** of ***contributions*** from each record in the subset.

Maximizing the scoring function is then equivalent to selecting all records that are making a ***positive contribution***.

$$F(S) = \max_{q} \sum_{s_i \in S} F(s_i \mid q)$$

INSIGHT: When treated as an additive function, ***further terms*** (i.e. soft constraints) may be introduced without interfering with the maximization step.

$$F(S) = \max_{q} \sum_{s_i \in S} \left[F(s_i \mid q) + \Delta_i \right]$$

Enforcing Soft Constraints with LTSS

PROBLEM:

We must alter the ***scoring function*** instead of restricting the search space.

When applied directly, these constraints ***violate*** the ***LTSS property*** of the scoring function and make exact, efficient search ***impossible***.

SOLUTION:

Interpret the scoring function as a ***sum*** of ***contributions*** from each record in the subset.

Maximizing the scoring function is then equivalent to selecting all records that are making a ***positive contribution***.

INSIGHT:

When treated as an additive function, ***further terms*** (i.e. soft constraints) may be introduced without interfering with the maximization step.

$$F(S) = \max_{q} \sum_{s_i \in S} F(s_i \mid q)$$

Enforcing Soft Constraints with LTSS

PROBLEM:

We must alter the ***scoring function*** instead of restricting the search space.

When applied directly, these constraints ***violate*** the ***LTSS property*** of the scoring function and make exact, efficient search ***impossible***.

SOLUTION:

Interpret the scoring function as a ***sum*** of ***contributions*** from each record in the subset.

Maximizing the scoring function is then equivalent to selecting all records that are making a ***positive contribution***.

INSIGHT:

When treated as an additive function, ***further terms*** (i.e. soft constraints) may be introduced without interfering with the maximization step.

$$F(S) = \max_q \sum_{s_i \in S} \left[F(s_i \mid q) + \Delta_i \right]$$

Enforcing Soft Constraints with LTSS

PROBLEM:

We must alter the ***scoring function*** instead of restricting the search space.

When applied directly, these constraints ***violate*** the ***LTSS property*** of the scoring function and make exact, efficient search ***impossible***.

SOLUTION:

Interpret the scoring function as a ***sum*** of ***contributions*** from each record in the subset.

Maximizing the scoring function is then equivalent to selecting all records that are making a ***positive contribution***.

INSIGHT:

When treated as an additive function, ***further terms*** (i.e. soft constraints) may be introduced without interfering with the maximization step.

Demonstration with Expectation-based Binomial

$$F(S) = \max \log \frac{P(Data \mid H_1(S))}{P(Data \mid H_0)}$$

$$H_0 : c_i \sim Bernoulli(p_0)$$

$$H_1 : c_i \sim Bernoulli(qp_0) \qquad 1 < q < \frac{1}{p_0}$$

$$F(S) = \max_{1<q<1/p_0} \log \prod_{s_i \in S} \frac{(qp_0)^{c_i}(1-qp_0)^{N_i - c_i}}{p_0^{\,c_i}(1-p_0)^{N_i - c_i}}$$

$$F(S) = \max_{1<q<1/p_0} \sum_{s_i \in S} c_i \log q + \sum_{s_i \in S} (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right)$$

$$F(S) = C \log\left(\frac{C}{Np_0}\right) + (N - C) \log\left(\frac{1 - C/N}{1-p_0}\right) \text{ for } C > Np_0 \text{, 0 otherwise}$$

#Trials

#Triggers

Baseline rate

Demonstration with Expectation-based Binomial

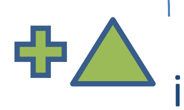
$$F(S) = \max_{1<q<1/p_0} \sum_{s_i \in S} c_i \log q + \sum_{s_i \in S} (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right)$$

$$F(S) = \max_{1<q<1/p_0} \sum_{s_i \in S} \left[c_i \log q + (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right)\right]$$

$$F(S \mid q) = \sum_{s_i \in S} \left[c_i \log q + (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right)\right]$$

Contribution from each sensor in subset

$$F(S \mid q) = \sum_{s_i \in S} \left[c_i \log q + (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right) \quad \right]$$



Log-likelihood F(s_i | q)

Reward /Penalty from constraints

Demonstration with Expectation-based Binomial

$$F(S) = \max \log \frac{P(Data \mid H_1(S))}{P(Data \mid H_0)}$$

$$F(S) = \max_{1<q<1/p_0} \log \prod_{s_i \in S} \frac{(qp_0)^{c_i}(1-qp_0)^{N_i-c_i}}{p_0^{c_i}(1-p_0)^{N_i-c_i}}$$

Contribution from each sensor, for a fixed q

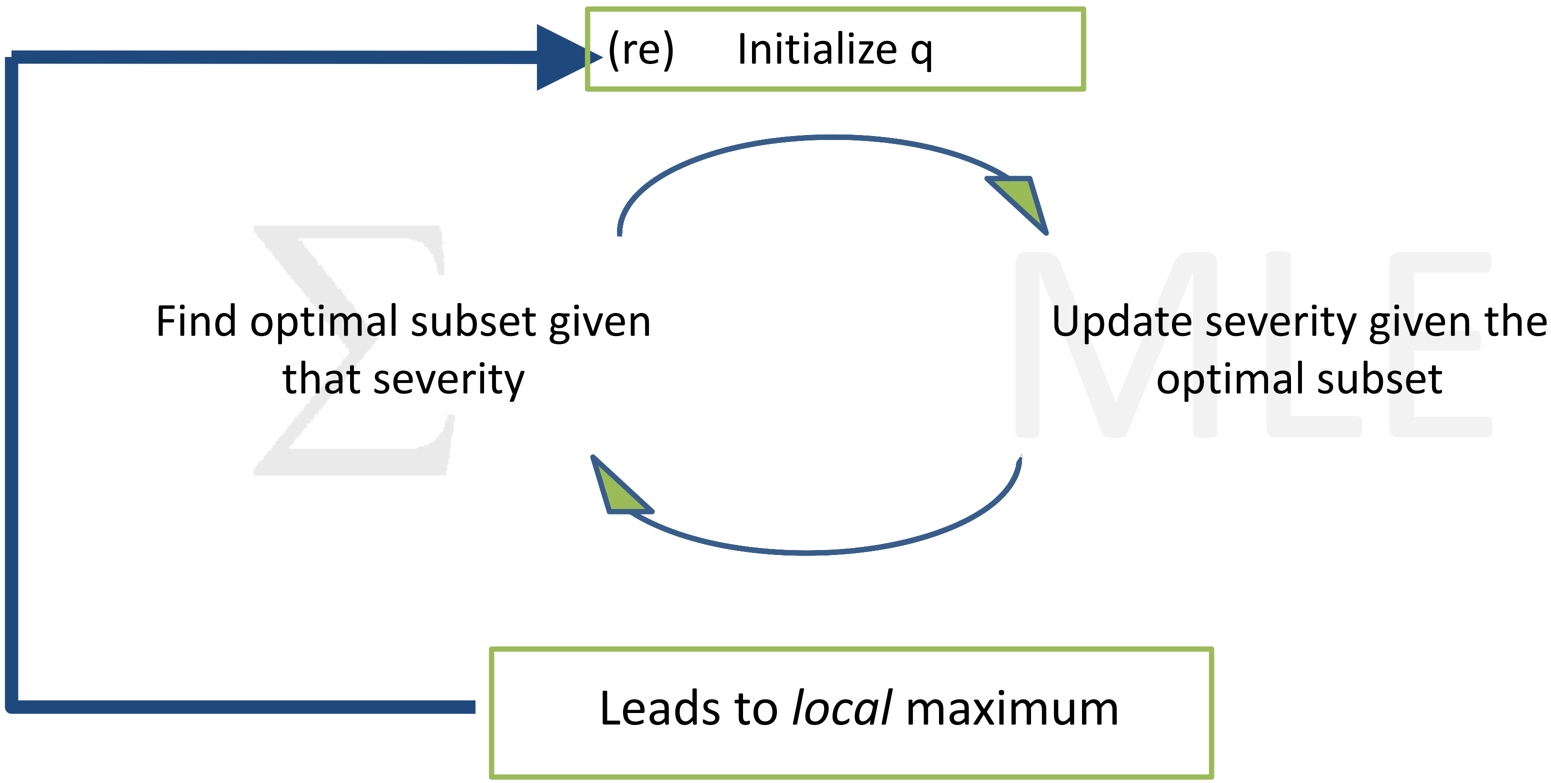
$$F(S \mid q) = \sum_{s_i \in S} \left[c_i \log q + (N_i - c_i) \log\left(\frac{1-qp_0}{1-p_0}\right) + \Delta_i \right]$$

Log-likelihood $F(s_i \mid q)$

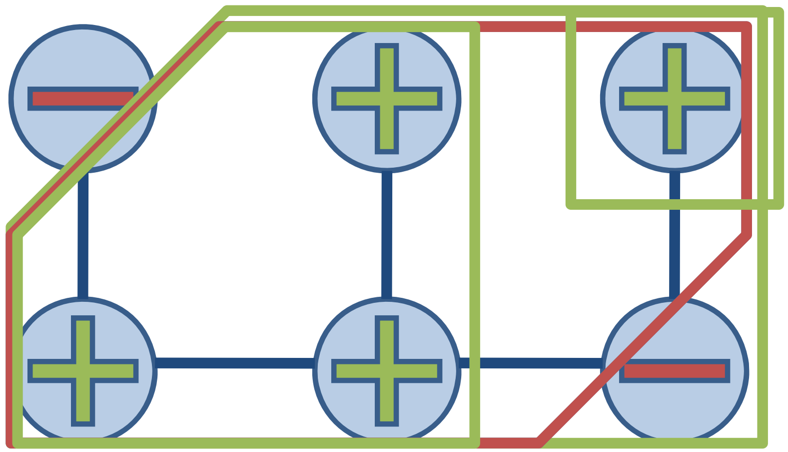
Reward /Penalty from constraints

From *Fixed* q to *All* q

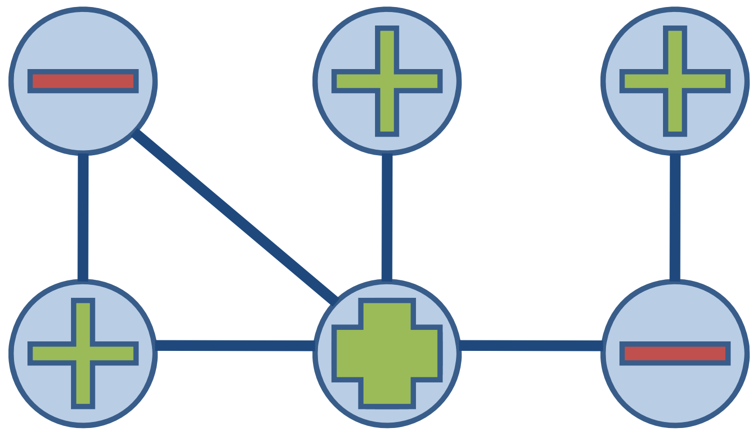
Our goal is to maximize F(S) ***over all q***



Additive GraphScan



Additive GraphScan



Additive GraphScan is 10x faster than regular, even though operating with multiple iterations

We represent groups of subsets as a string of
0's, 1's, and ?'s

Priority Ranking	1	2	3	4	5	6
Bit String	1	0	0	1	?	?

The above bit string represents 4 possible subsets:
{1,4} {1,4,5} {1,4,6} {1,4,5,6}

A Naïve approach would search all 2^N subsets and is computationally infeasible

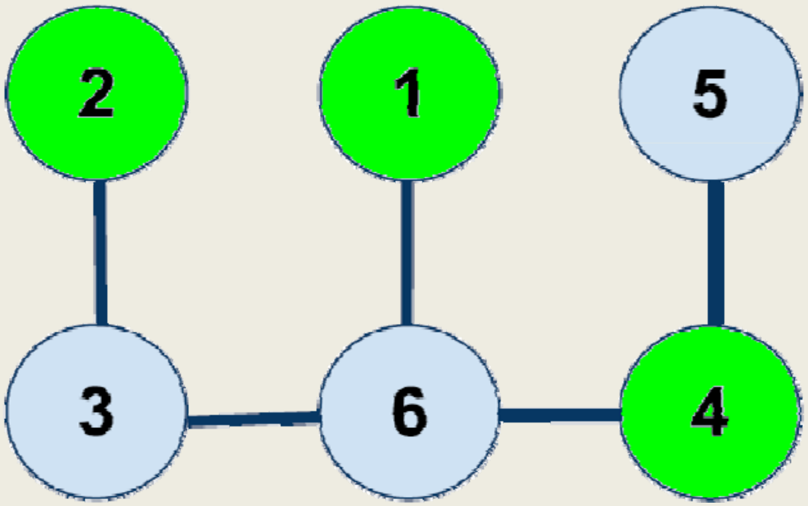
1	2	3	4	5	6
?	?	?	?	?	?

→

1	2	3	4	5	6
1	?	?	?	?	?
0	?	?	?	?	?

→

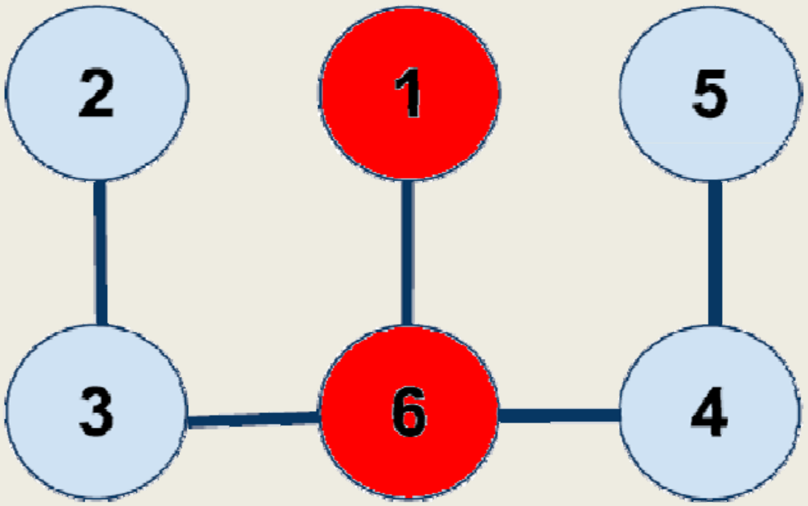
1	2	3	4	5	6
1	1	?	?	?	?
1	0	?	?	?	?
0	?	?	?	?	?



	1	2	3	4	5	6
S_1	1	?	?	?	?	?
S_2	0	1	?	?	?	?
~~S_3~~	~~0~~	~~0~~	~~1~~	~~?~~	~~?~~	~~?~~
S_4	0	0	0	1	?	?
~~S_5~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~?~~
~~S_6~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~

Seed nodes have higher priority than all of their neighbors

We can rule out bit strings whose highest priority node is not a seed node

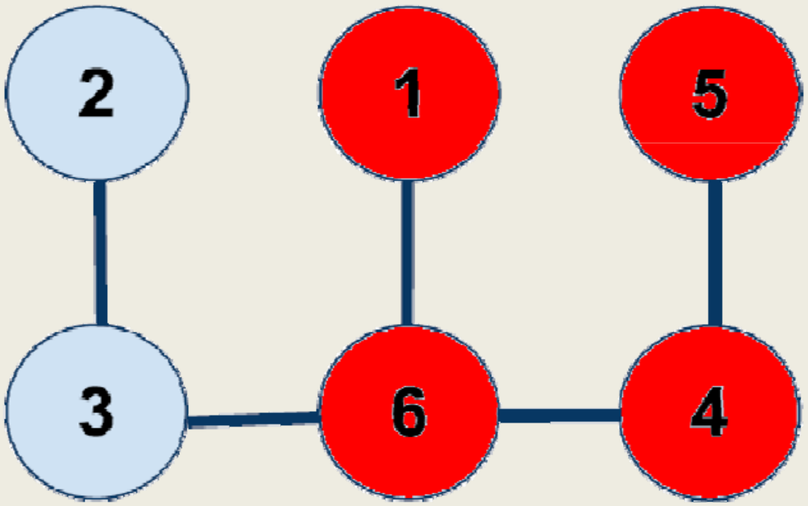


	1	2	3	4	5	6
S_1	1	?	?	?	?	?
S_2	0	1	?	?	?	**0**
~~S_3~~	~~0~~	~~0~~	~~1~~	~~?~~	~~?~~	~~?~~
S_4	0	0	0	1	?	**0**
~~S_5~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~?~~
~~S_6~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~

Seed nodes have higher priority than all of their neighbors

We can rule out bit strings whose highest priority node is not a seed node

If we rule out a high priority node, we can also rule out all of its lower priority neighbors...



Seed nodes have higher priority than all of their neighbors

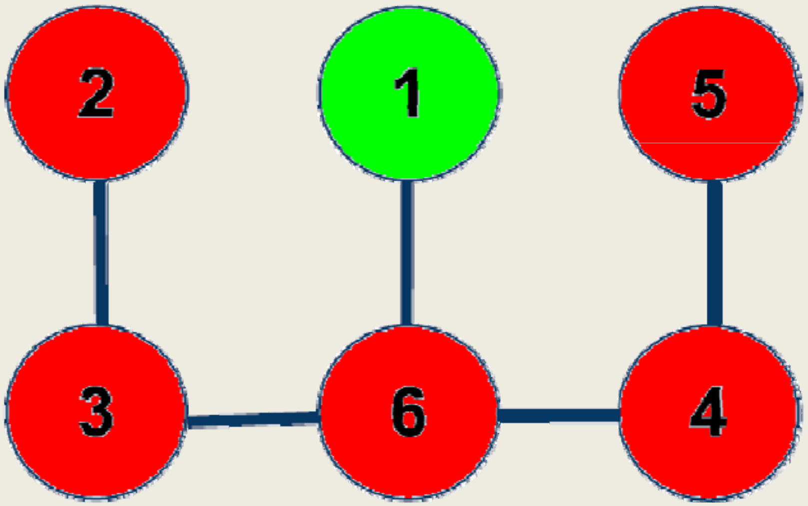
We can rule out bit strings whose highest priority node is not a seed node

	1	2	3	4	5	6
S_1	1	?	?	?	?	?
S_2	0	1	?	**0**	**0**	0
~~S_3~~	~~0~~	~~0~~	~~1~~	~~?~~	~~?~~	~~?~~
S_4	0	0	0	1	?	0
~~S_5~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~?~~
~~S_6~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~

If we rule out a high priority node, we can also rule out all of its lower priority neighbors…

…and any additional nodes that are disconnected when these nodes are ruled out

Propagation of bit strings:



	1	2	3	4	5	6
S_1	1	?	?	?	?	?
S_2	0	1	?	0	0	0
~~S_3~~	~~0~~	~~0~~	~~1~~	~~?~~	~~?~~	~~?~~
S_4	0	0	0	1	?	0
~~S_5~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~?~~
~~S_6~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~

Pull off $\mathbf{S_1}$ and consider the two cases of including or excluding node 2

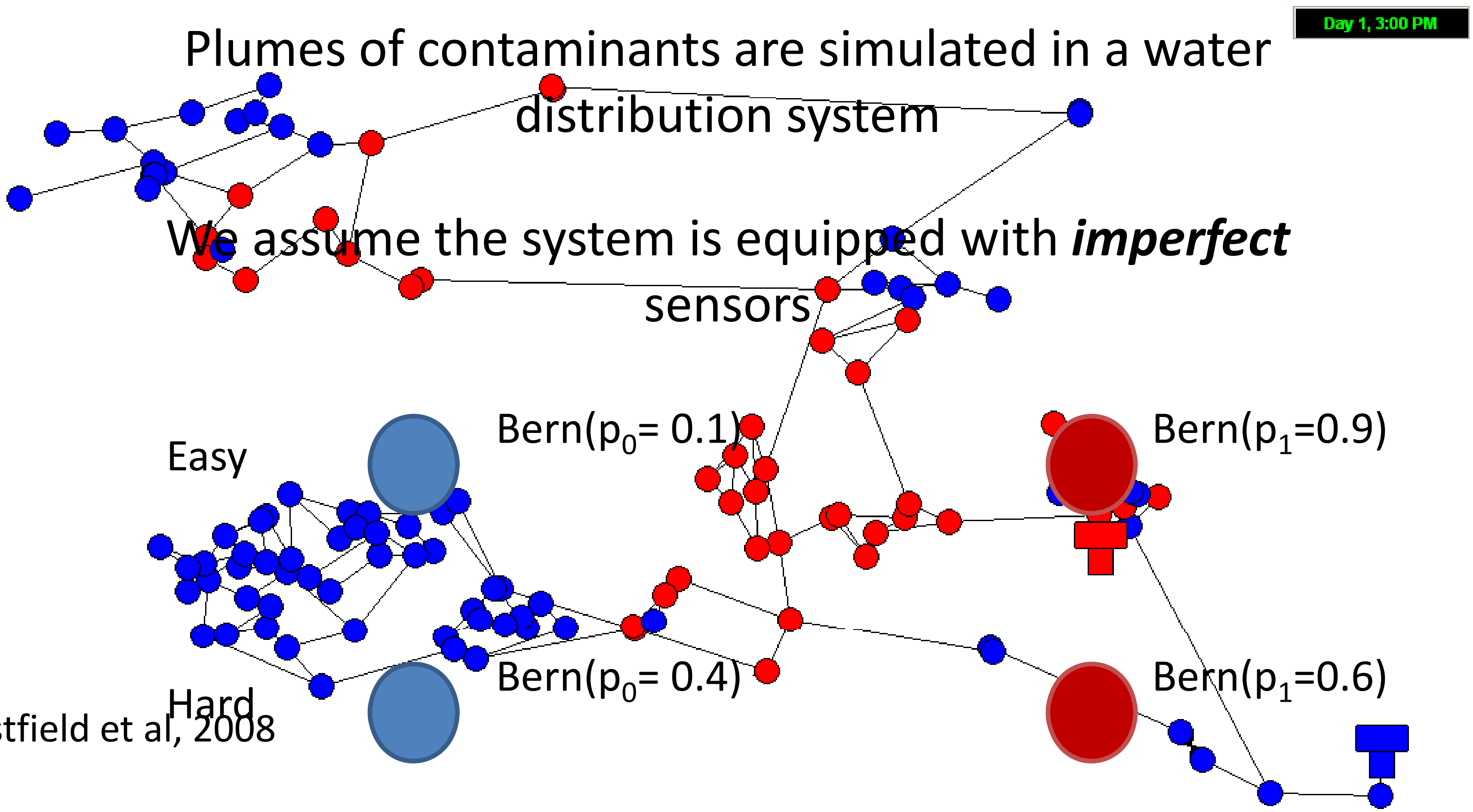
Including node 2 implies including nodes 3 and 6
$\mathbf{S_{1a}}$: 1 1 1 ? ? 1

Excluding node 2 implies excluding nodes 3, 4, 5, and 6
$\mathbf{S_{1b}}$: 1 0 0 0 0 0

Data: Battle of the Water Sensor Networks

Plumes of contaminants are simulated in a water distribution system

We assume the system is equipped with ***imperfect*** sensors



Easy

Bern(p_0= 0.1)

Bern(p_1=0.9)

Hard

Bern(p_0= 0.4)

Bern(p_1=0.6)

Ostfield et al, 2008

Competing Methods

Upper Level Sets:
A heuristic that is not guaranteed to find the most anomalous subgraph

Patil & Taillie, 2004

ULS

GraphScan:
Determines the most anomalous subgraph *without further constraints*

Speakman & Neill, 2010

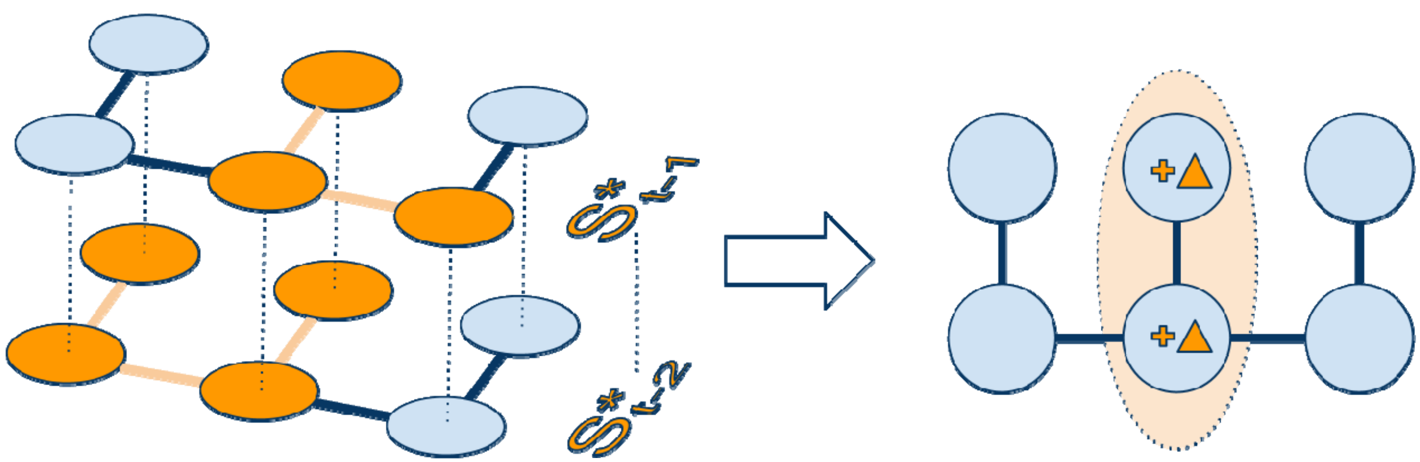
GS

Competing Methods

GraphScan with basic temporal consistency

$$F(S) = \max_q \sum_{s_i \in S} \left(F(s_i \mid q) + \Delta_i \right) \qquad \Delta_i = \begin{cases} +\Delta \text{ if } s_i \in \Omega \\ 0 \text{ otherwise} \end{cases}$$

ADD-GS



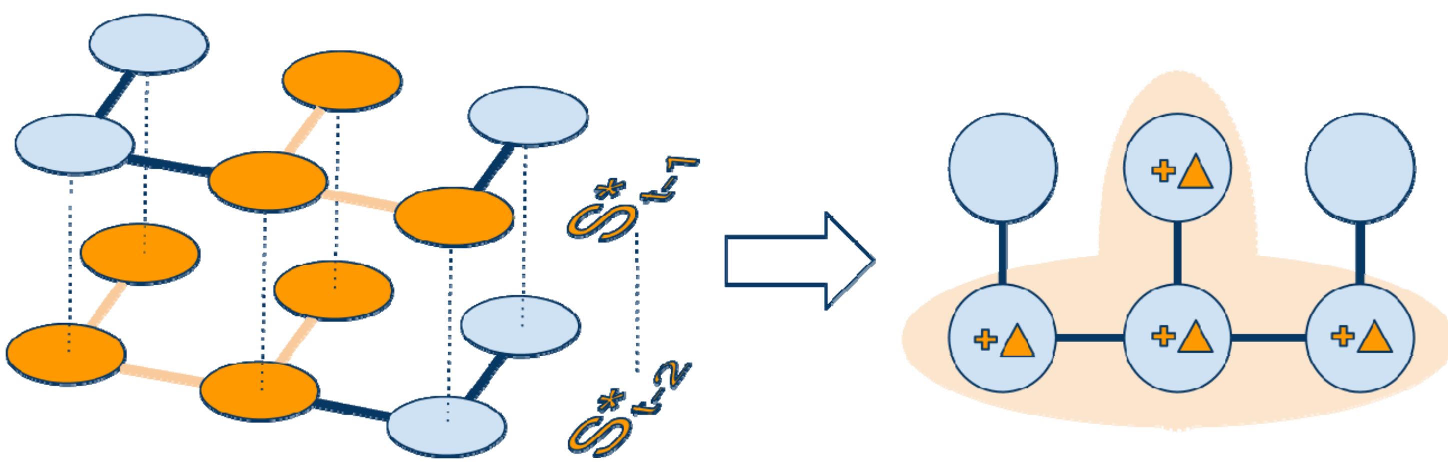
Competing Methods

GraphScan with temporal consistency: Overlap and Neighbors

$$F_{pen}(S) = \max_{q} \sum_{s_i \in S} \left(F(s_i \mid q) - \lambda + \Delta_i \right)$$

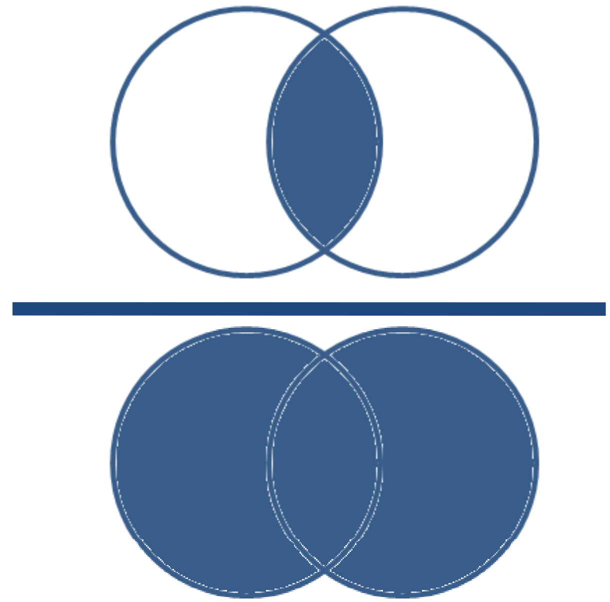
$$\Delta_i = \begin{cases} +\Delta \text{ if } s_i \in \Omega_{neighbor} \\ 0 \text{ otherwise} \end{cases}$$

GS-SON



Results: Spatial Overlap

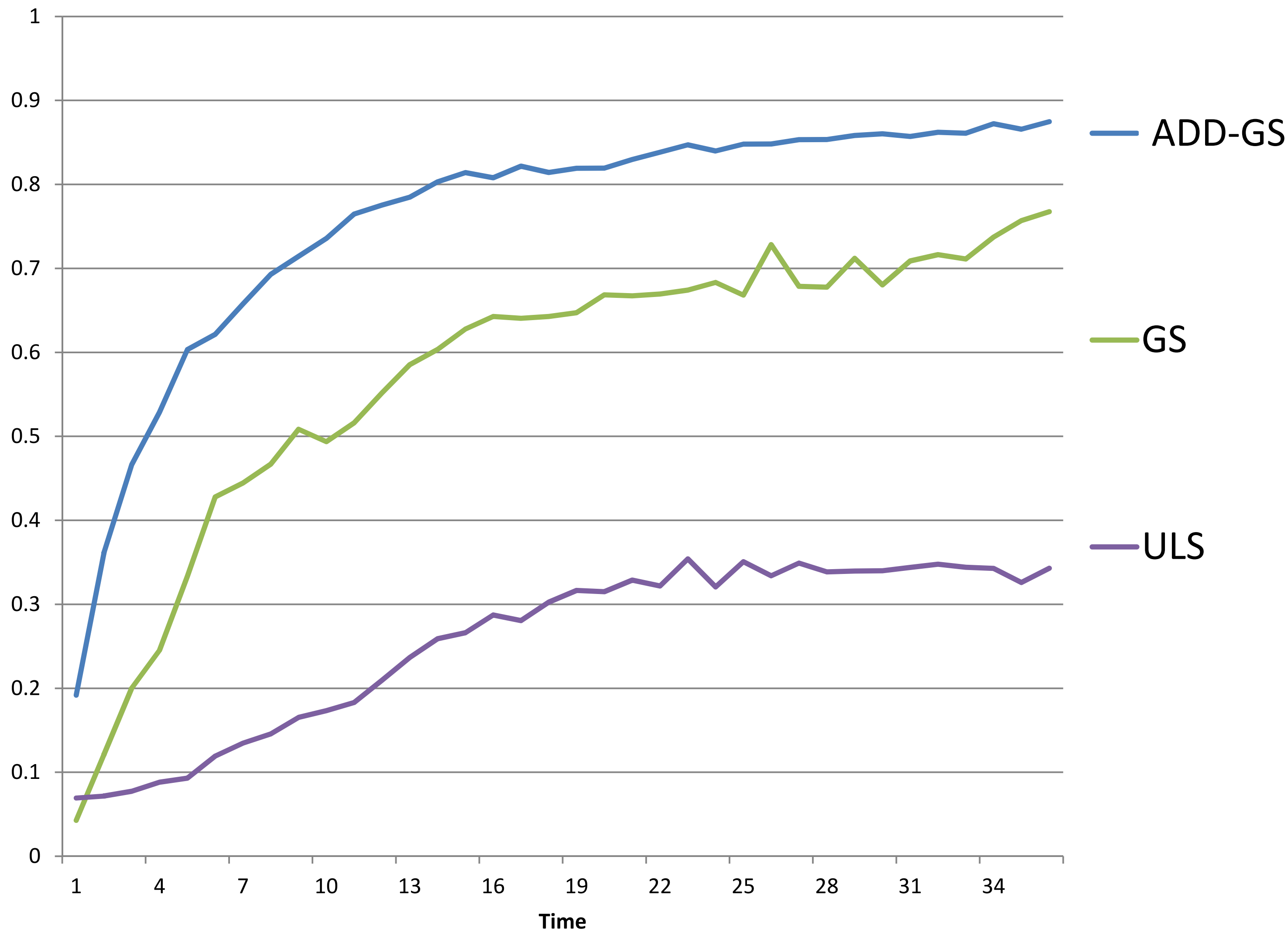
$$Overlap = \frac{A \cap B}{A \cup B} =$$



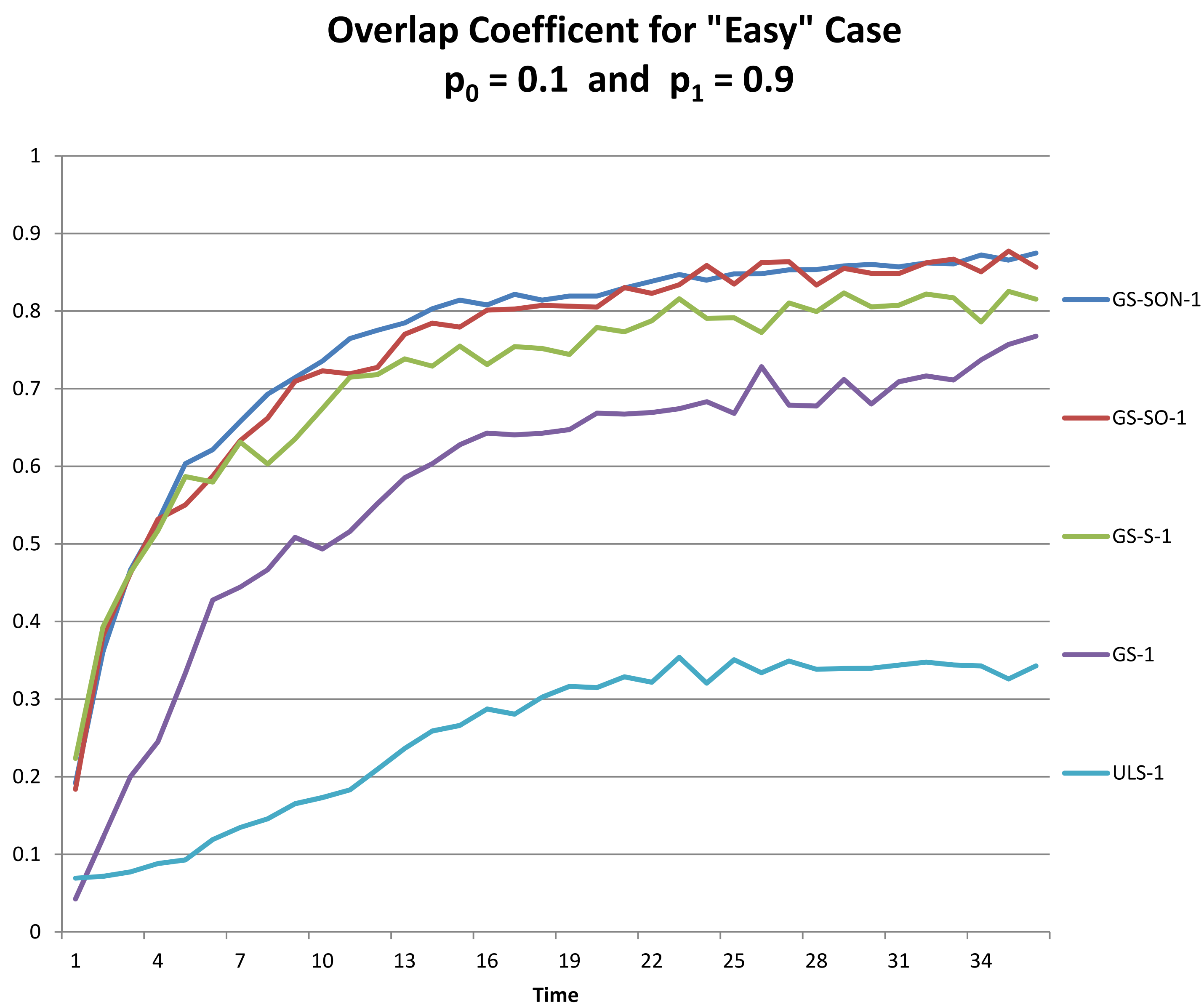
$$Overlap = 1$$ Perfect Match

$$Overlap = 0$$ Completely Disjoint

Overlap Coefficent for "Easy" Case
p_0 = 0.1 and p_1 = 0.9

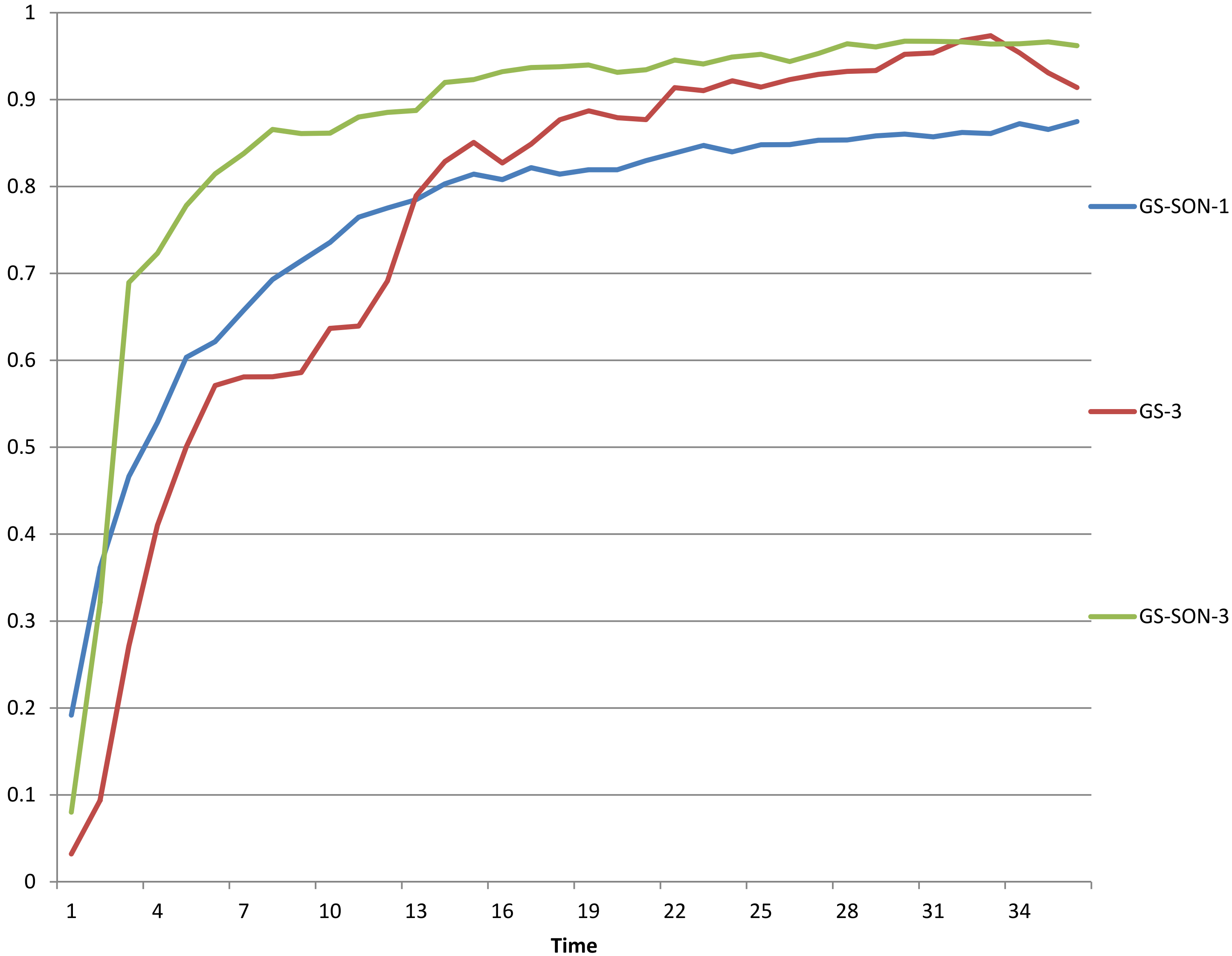


Hours until Detection	% Detected
7.66	100%
9.65	97.5%
15.4	92.4%

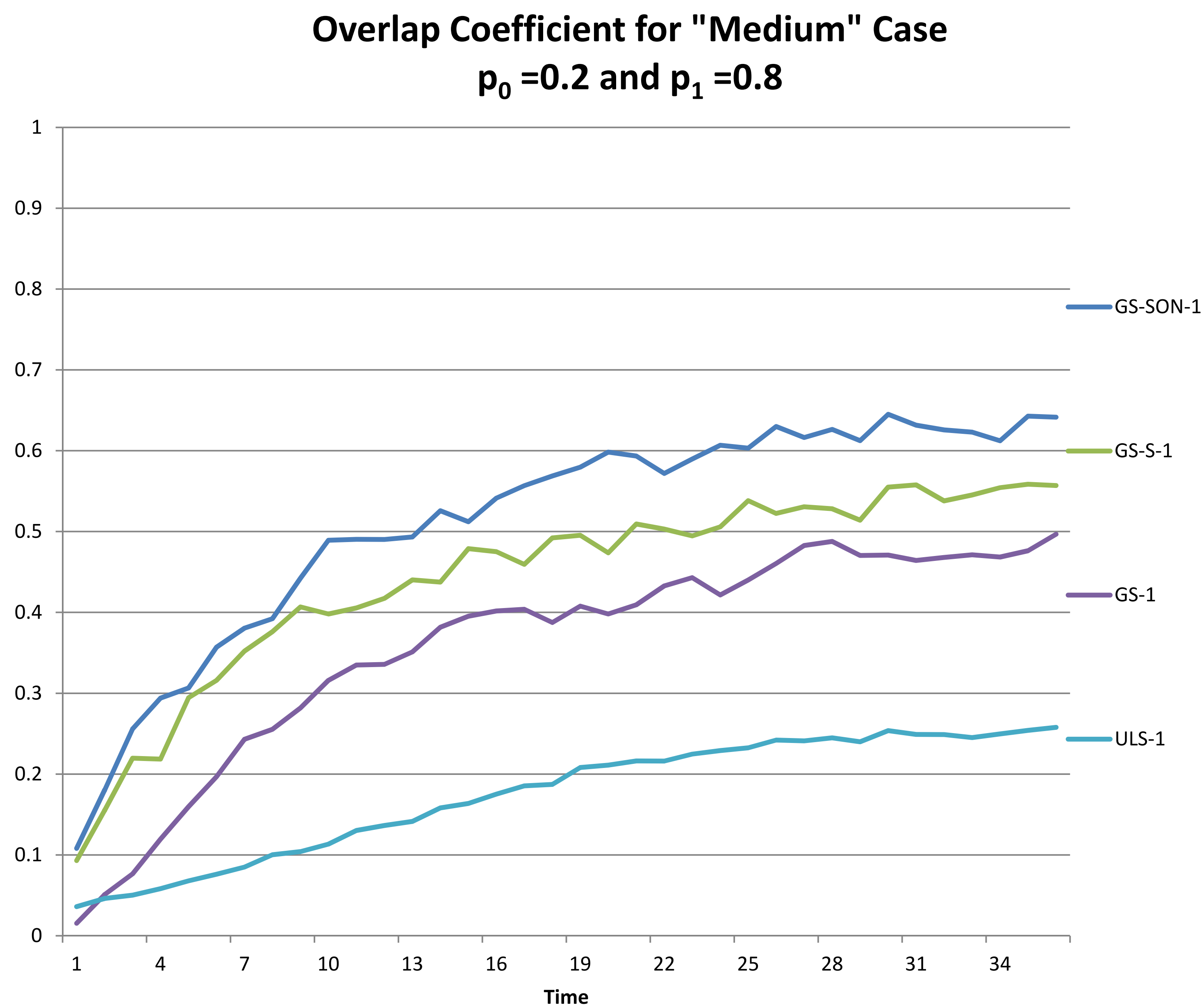


Hours	% Detected
7.66	100%
7.82	100%
10.27	97.6%
9.65	97.5%
15.4	92.4%

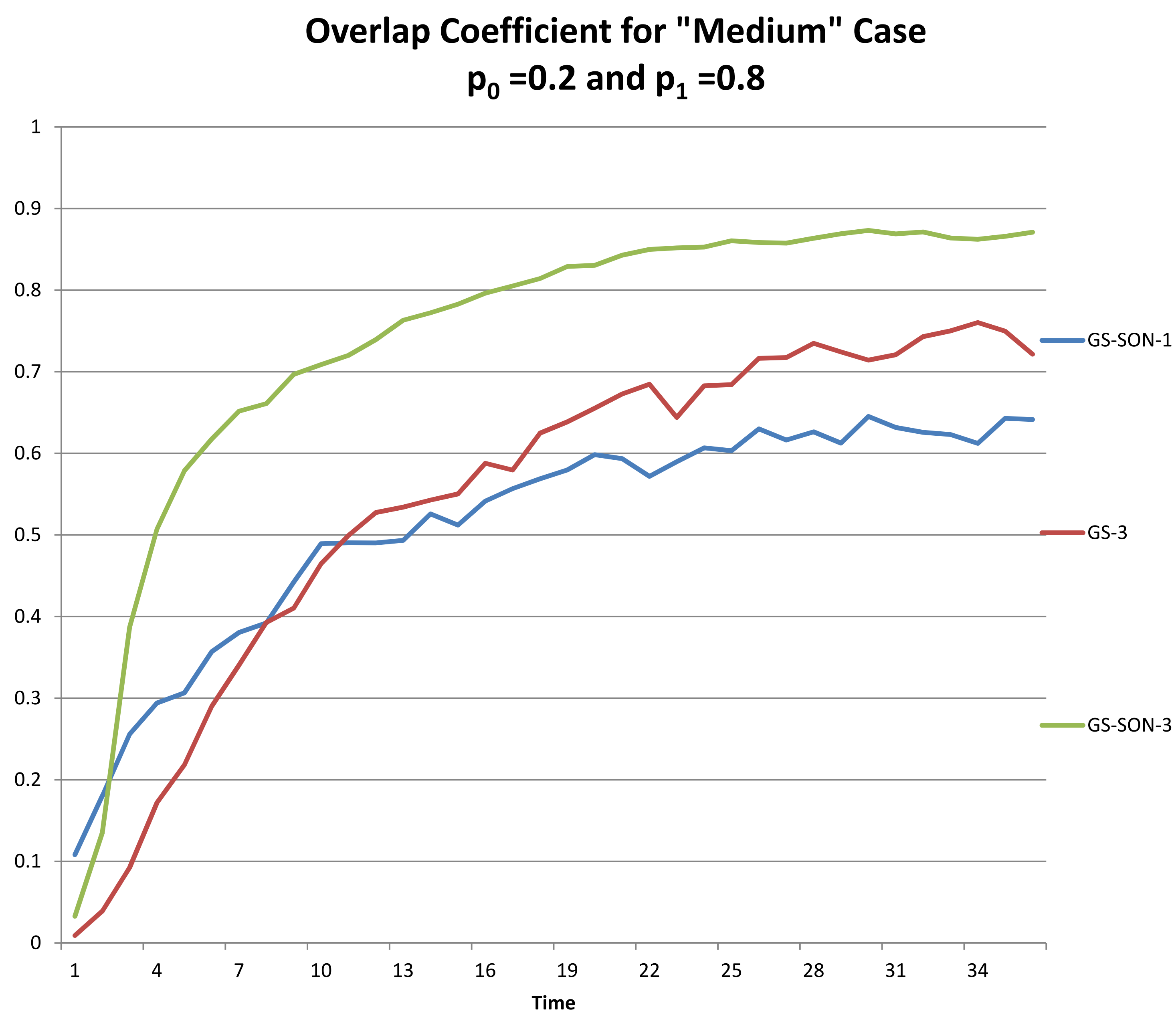
Overlap Coefficent for "Easy" Case
$p_0 = 0.1$ and $p_1 = 0.9$



Hours	% Detected
7.66	100%
8.2	100%
4.6	100%

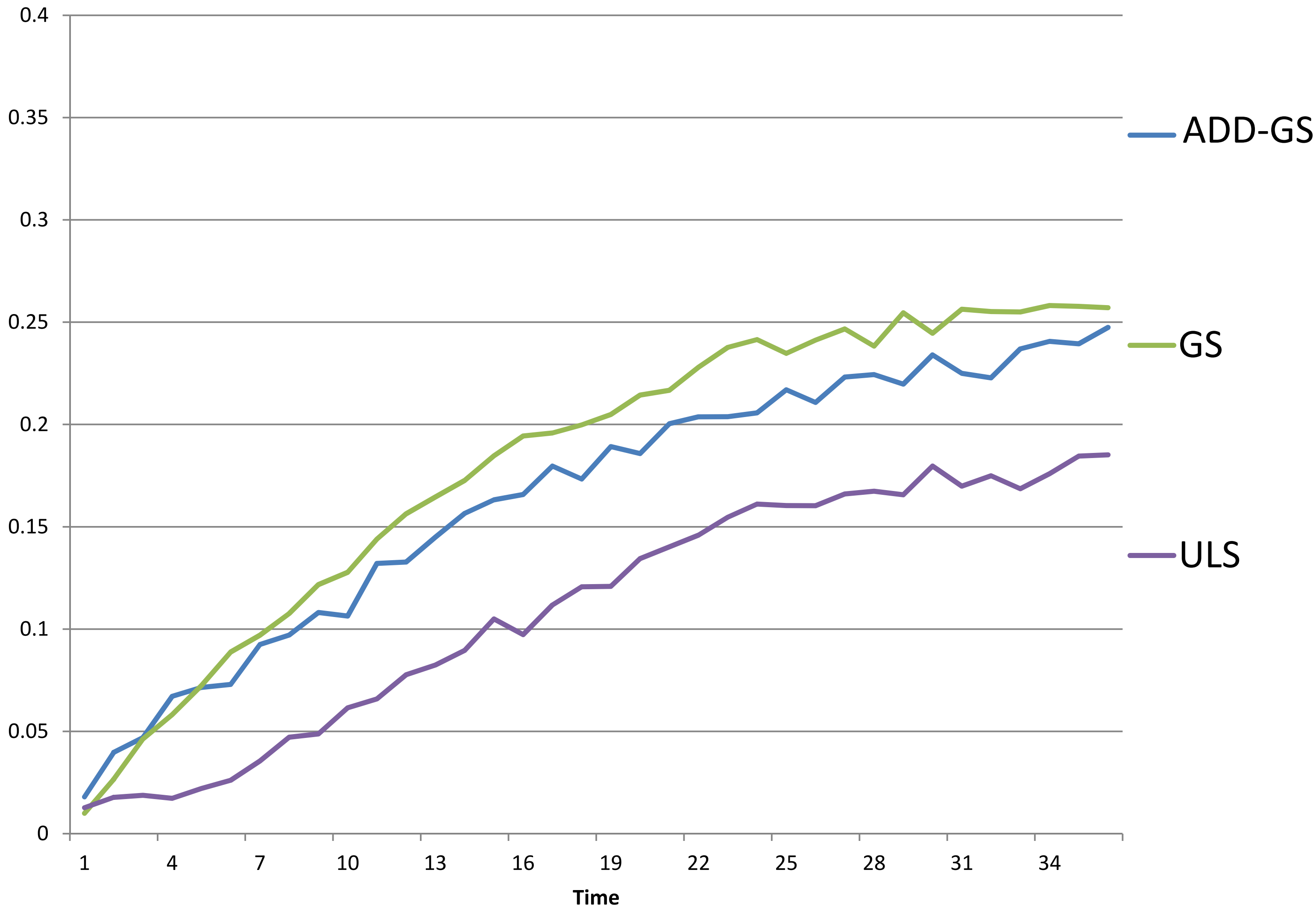


Overlap Coefficient for "Medium" Case
p0 =0.2 and p1 =0.8
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
4
7
10
13
16
19
22
25
28
31
34
Time
GS-SON-1
GS-S-1
GS-1
ULS-1



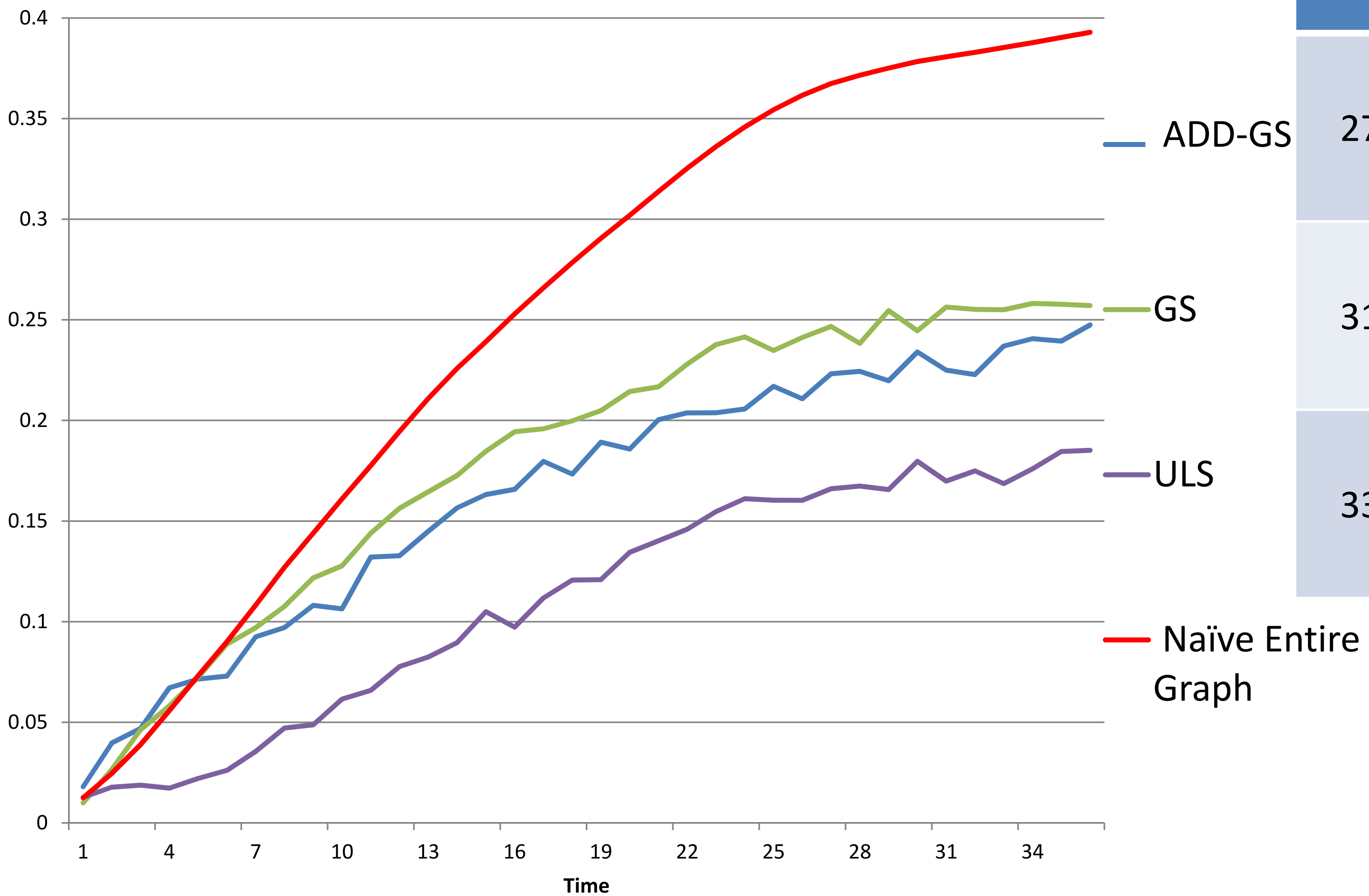
Overlap Coefficient for "Medium" Case
p0 =0.2 and p1 =0.8
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
4
7
10
13
16
19
22
25
28
31
34
Time
GS-SON-1
GS-3
GS-SON-3

Overlap Coefficient for "Hard" Case
$p_0 = 0.4$ and $p_1 = 0.6$

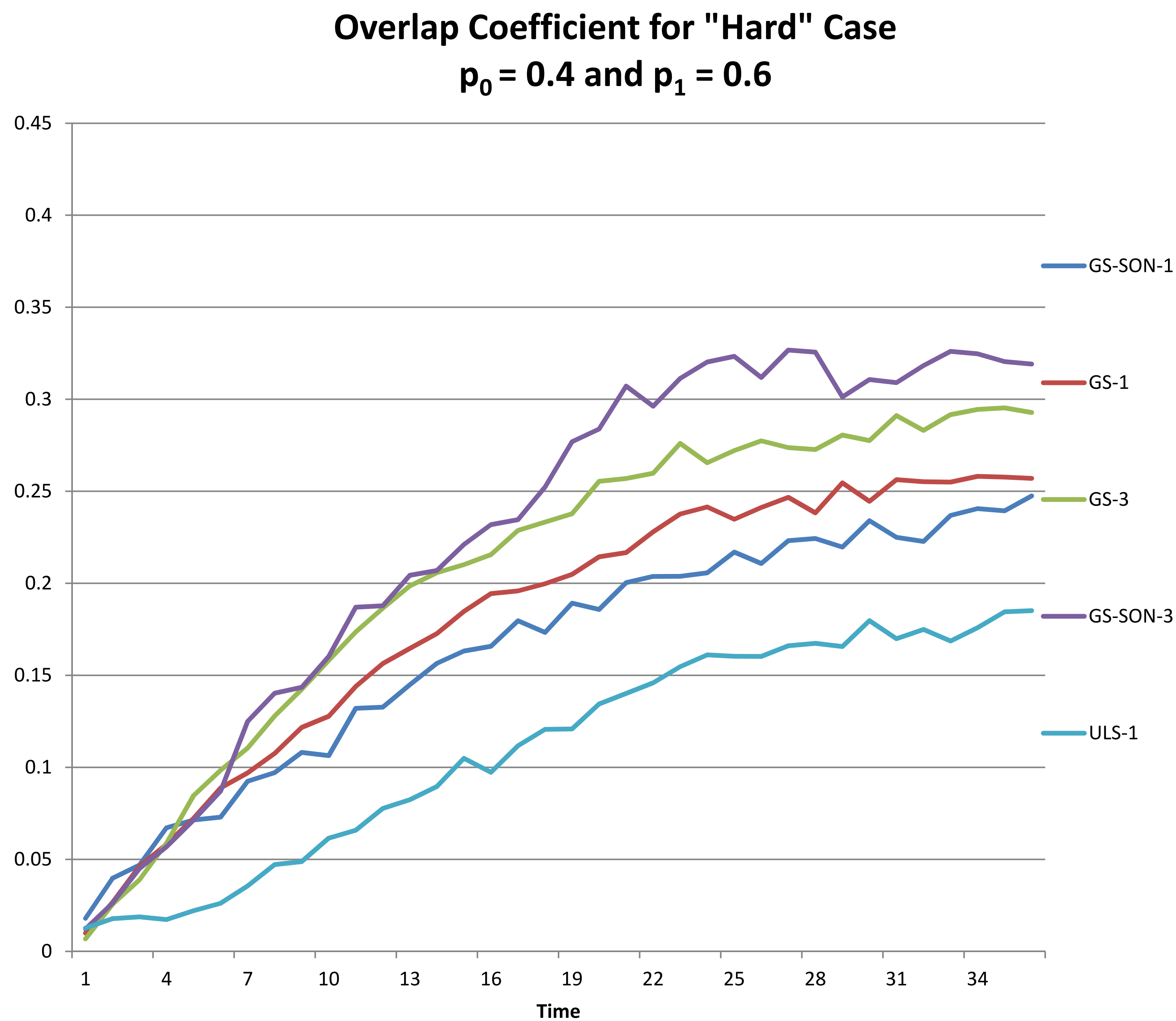


	Hours Until Detection	% Detected
ADD-GS	27.4	52.5%
GS	31.0	26.8%
ULS	33.5	15.8%

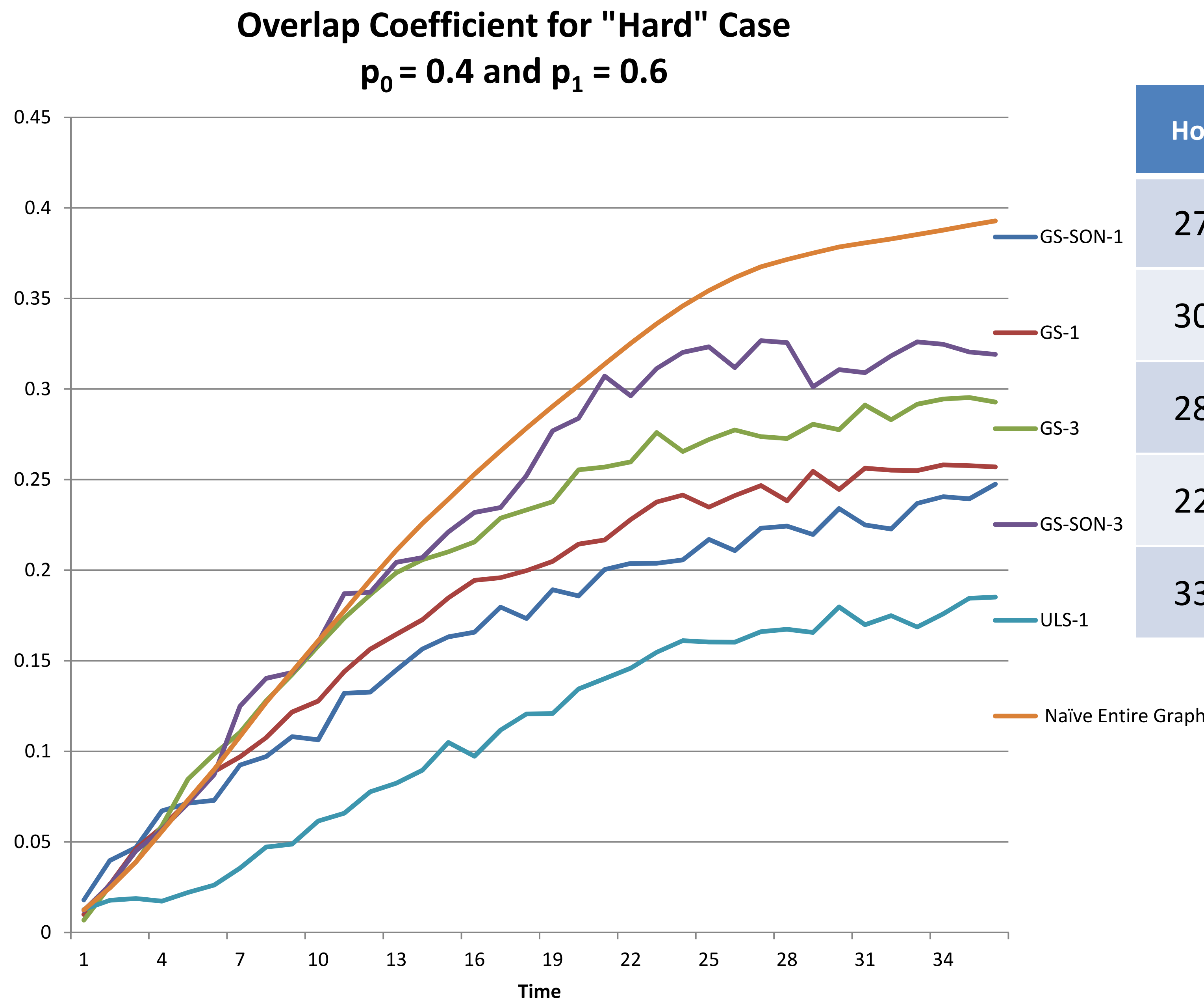
Overlap Coefficient for "Hard" Case
$p_0 = 0.4$ and $p_1 = 0.6$



Hours Until Detection	% Detected
27.4	52.5%
31.0	26.8%
33.5	15.8%



Hours	% Detected
27.4	52.5%
30.6	26.8%
28.1	48.0%
22.6	76.8%
33.5	15.8%



Hours	% Detected
27.4	52.5%
30.6	26.8%
28.1	48.0%
22.6	76.8%
33.5	15.8%

Conclusions

Provided a framework that allows soft constraints to influence the scoring function and give preference to subsets of desired temporal consistency while still allowing an efficient search for the highest scoring connected sub-graph

Applied the EBB scoring function with ***Additive GraphScan*** to the task of detecting contaminants in a water distribution system

Empirical results showed temporal consistency constraints ***reduced the time to detect*** the contaminants and ***increased spatial accuracy*** of the methods

Future work:
Forward & Backward Consistency

This work assumed information was only being passed "forward" in time

We can also share current information with the past

Alter past subsets to be consistent with current information

Use the altered subsets from the past to make more informed searches in present

"Interlink" time sharing to ensure a smoother consistency through time

Also useful for source tracing


healthmap.org/en/
Map
Satellite
News / Media
Google News
Moreover Technologies
Wildlife Disease Node
Baidu 新闻
SOSO资讯
Scientific Reports
ProMED Mail
GeoSentinel
EuroSurveillance
Community Contributions
HM Community News Reports
Eyewitness Reports
Official Sources
World Health Organization
World Org. for Animal Health
Food and Agriculture Org
HealthMap
Global health, local information
Activity Index
Low
High
Country level
Province or local level
Canada
North Atlantic Ocean
North Pacific Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
México
Brasil
Argentina
Kazakhstan
India
Egypt

Thank you!

speakman@cmu.edu
neill@cs.cmu.edu