The Unreasonable Effectiveness of Graph Theory: With apologies to Eugene Wigner

AnNet Workshop 2019

Dr. Phil Tee
email: phil@moogsoft.com; ptee2@asu.edu; p.tee@sussex.ac.uk

April 8, 2019

Moogsoft Inc, ASU Beyond, and University of Sussex

Unreasonable Effectiveness

- First coined by the great Hungarian-American Physicist Eugene Wigner (1902-1995) in an essay for the Richard Courant lectures [1]
- Posed a very simple question, "Just why is mathematics so good in the hands of scientists at predicting the world in which we live?"
- How good? Anomalous Magnetic Moment of an Electron $a_e = \frac{g-2}{4\pi}$ which measures the strength that the spin of an electron couples to a magnetic field

  $$a_e = 0.001\ 159\ 652\ 181\ 643(764) \text{ Calculated}$$
  $$a_e = 0.001\ 159\ 652\ 180\ 73\ \ (28) \quad \text{Measured!}$$

  Which is the distance San Francisco to London within the width of a human hair!!!!!!
- In this talk I will show the many unreasonably effective applications of Graph Theory to Fault Localization!

Summary

- I cover a very very brief introduction to key concepts in Graph Theory
- Fault localization, is the motivating practical problem
- In particular root cause is the relationship between the signals a network can send and what is going on
- I survey a series of applications of Graph Theory to the problem of Fault Localization

Background

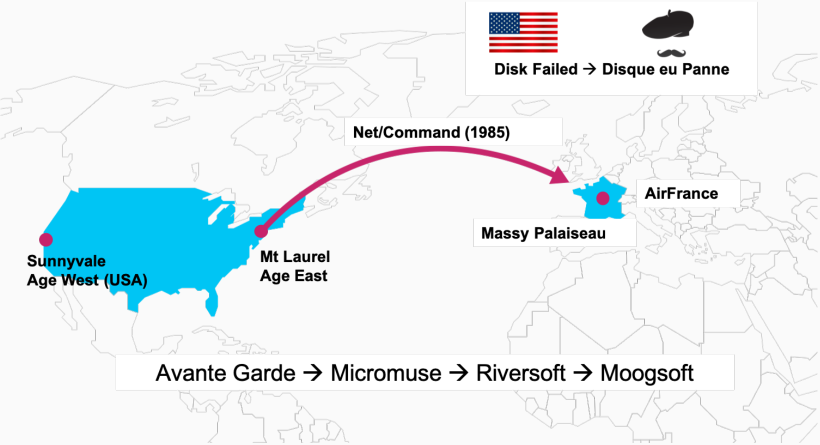
A little bit on me...

- Nearly 30 years building software for Fortune 2000 companies; Currently CEO of Moogsoft Inc
- 2 IPOs, 2 trade sales, one on way to full scale (Moogsoft)
- And...
- PhD in Informatics, speciality information theory of graphs and network science
- Publications in IEEE, EPJ-B, Entropy, Complex Networks, on graph entropy, network science, cancer genetics, and spacetime geometry [2, 3, 4]
- Adjunct Professor at the Beyond Center, ASU, and visiting researcher at University of Sussex

Fault Localization - The Very Big Picture

- An incident is a problem on a digital service that causes service interruption
- They are hopefully detected by network management systems (NMS)
- NMS systems collect status messages (logs) and typically poll for status by performing health checks with protocols such as SNMP
- Each status record is called an Event, and will have a source node and a timestamp as a minimum
- Events can be promoted to an Alert by the NMS to indicate a fault condition, which may or may not be actionable.
- Fault Localization is the task of working out which subset of alerts and their events underly the incident and hopefully point the way to a fix
- Excellent reviews of the field in Steinder *et al* [5, 6].

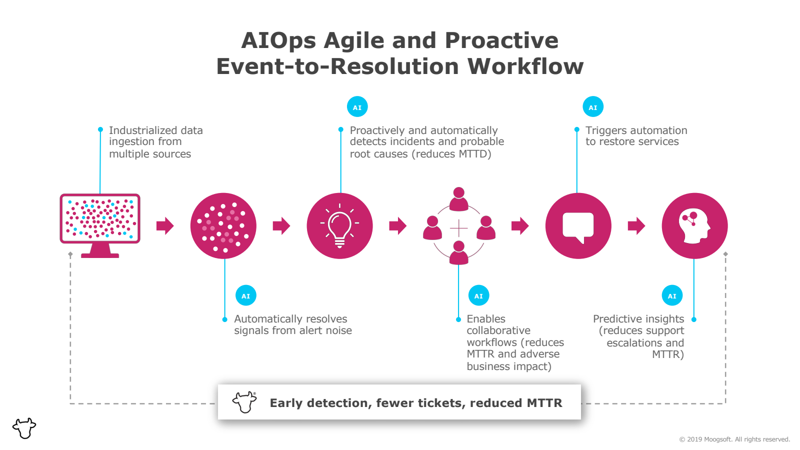
Fault Localization - A Very Brief History



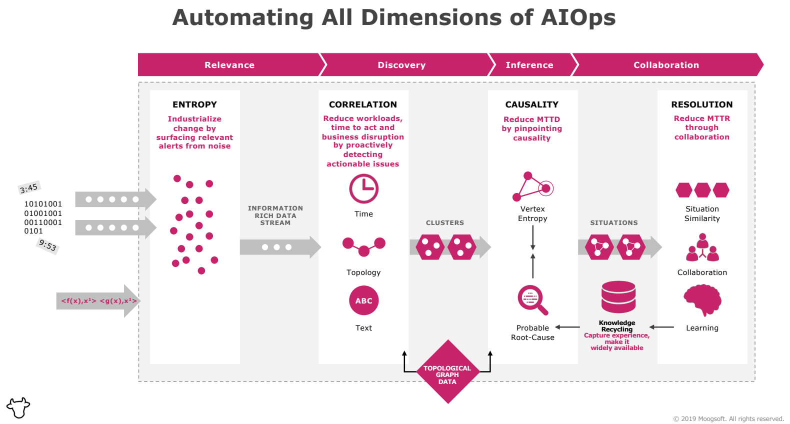
A little bit about Moogsoft

- Founded in January 2012 in London, HQ now in San Francisco
- 130 + customers across every segment of corporate IT
- Principle business is AIOps, the use of AI techniques to help with Systems and Network Management
- 50 + Patents, significant footprint in academic research

AIOps In Action



Many different algorithms in a pipeline...



Partnering with Academia to Produce IP!

Our +50 Machine Learning Patents Make AI Valuable and Consumable

Noise Reduction ← Correlation → Causality

Entropy	Time	Language	Topology	Vertex Entropy	Algorithmic Correlation Engine	Feedback	Probable Root Cause
Alert Significance	Timestamp Patterns	Text value similarities	Network proximity patterns	Critical node identification	Algorithms application with a deterministic policy	Learnings from Ops behavior	Root cause identification

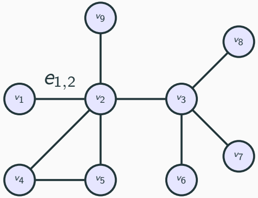
© 2019 Moogsoft. All rights reserved.

A Lightening Course in Graph Theory

Graph Theory Basics

Definitions

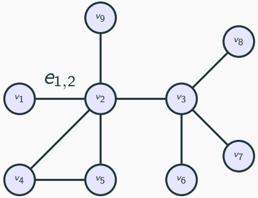
- A Graph $G(V, E) \equiv G(V, V \times V)$, is a collection of vertices $v_i \in V$, and edges $e_{ij} \in V \times V$.
- If $e_{ij} = e_{ji}$ the graph is said to be undirected, and if $\{e_{ii}\} = \emptyset$ the graph is said to be simple.
- For each node in the graph v_i we can compute the number of in directed edges k_i^{in} and out directed edges k_i^{out}, and for undirected graph simply the incident edges k_i, call the in-degree, out-degree or degree.



Graph Theory Basics

Definitions

- Graphs can be decomposed into sub-graphs $H(V,E) \subset G(V,E)$, implying $V_H \subset V_G$. A very important subset being the set of all 'triangles' in a graph (e.g. $\{v_2, v_4, v_5\}$).
- Another strong structural hint are the 'stable sets' of a graph, or decomposition of V into sets of vertices that are not adjacent (do not share an edge). The minimum number of such sets is the chromatic number χ_G of the graph; etymological origin coming from map coloring analogy. For example $\{\{v_1, v_4, v_6, v_7, v_8, v_9\}, \{v_2\}, \{v_3\}, \{v_5\}\}$ is such a decomposition of our graph.



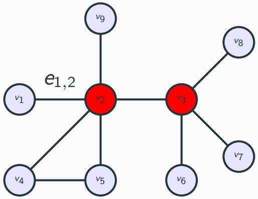
Graph Theory Basics

Perfection, Clustering, Diameter, Centrality!

- A 'Perfect' graph on $n = |V|$ nodes, K_n, is the maximally connected graph, and has $1/2n(n-1) = \binom{n}{2}$ edges.
- Clustering measures how 'Perfect' a graph is, by counting the number of triangles in a graph as a fraction of how many triangles there *could* be. It is often defined as:

$$C = \frac{3 \times \ \# \text{ triangles}}{\# \text{ open triples}}, \text{ globally}$$

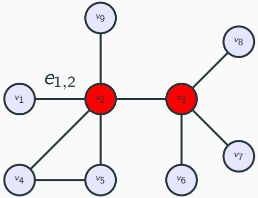
$$C_i = \frac{2 \times \ \# \text{ edges in 1-hop neighborhood}}{k_i(k_i - 1)}$$



Graph Theory Basics

Perfection, Clustering, Diameter, Centrality!

- Diameter of a graph is the longest shortest path (sequence of edges) between any two vertices. Somewhat related and occasionally used , is the Volume of the graph $vol(G) = \sum_i k_i = 2|E|$.
- Centrality measures the importance of a node for navigation in the graph, and is very important for identifying which nodes cause disconnection in a graph. Betweenness Centrality is the fraction of shortest paths between pairs of nodes that pass through a given node. In our graph the BC of $v_2 = 0.69$, and say $v_9 = 0.0$.



Graph Theory Basics

Graph/Matrix Duality

- Can define many matrix representations of a graph.
- Adjacency Matrix $\boldsymbol{A} = A_{ij} = 1, \exists e_{ij} \in E$, otherwise $A_{ij} = 0, A_{ii} = 0$.
- Lots of interesting properties, in particular A^n_{ij} is the number of n lengths paths between node i and j. Note links can be traversed more than once!
- Degree matrix $\boldsymbol{\Delta} = \Delta_{ij} = 0, i \neq j, \Delta_{ii} = k_i$.
- Laplacian $\boldsymbol{L} = L_{ij} = \Delta_{ij} - A_{ij}$.
- The Laplacian has interesting eigenvalues λ_i, with 0 occurring with at least multiplicity 1. The multiplicity of $\lambda_0 = 0$ is the number of disconnected subgraphs. The second smallest eigenvalue is called the algebraic connectivity of the graph, and the associated eigenvector the Fiedler vector. They are useful in assessing how easy it is to dismember the graph into disconnected "modules".
- The name "Laplacian" is intentional as in certain discrete applications it plays an analogous role to the Laplace ∇^2 operator. For example, if ϕ_i is the "heat" of a node in a graph, the heat diffusion equation for the whole graph can be written:

$$\frac{d\phi}{dt} + k\boldsymbol{L}\phi = 0 \tag{1}$$

Application 1: Rules Based Approaches

Rules and Graphs?

The correspondence between rules and decision trees...

- The fundamental idea is to go from a set of received events to the causal indicator of the problem.
- Begins with mapping the states of a managed system to a direct graph of states.
- In Figure 1 we depict the states as nodes in this graph, and the links represent rules that must be matched to transition the system between states.
- Typically these rules are statements about the attributes of a received event in the form of:

```
#
# Declare Event
#
event={
        "source" : host1, "check" : pingFail,
        "description": "Ping Failed on host1"
        };
# Check Transition 1
if event.type == "pingFail" then
do
     fireTransition( event.source, "State 1" );
done;
...
```

- Arrival in a given state generally triggers an operational notification/alert/incident

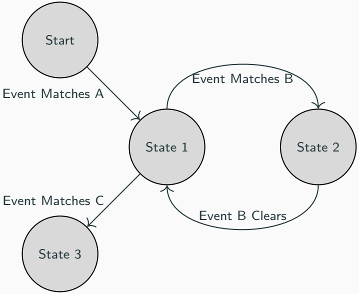


Figure 1: Example of a Basic State Graph in a Rules Based System (reproduced from [7])

Implementations and Limitations

- This approach is the single most popular and widespread implementation of Fault Localization.
- It is the core approach of products from IBM/HP/CA and most leading suppliers of operational support software
- The IT Infrastructure Library (ITIL), a set of manuals produced by the British Government [8], encodes the preferred operational procedures around such an approach.
- BUT, it relies upon being able to produce a model such as Figure 1, which is often not possible, and suffers badly from ambiguity in transition rules.
- In other work by my colleague Rob Harper and myself ([9]), we propose a different data driven approach to rules that overcomes these limitations.

Causal Graphs and Codebook

- Pioneered in the early 1990s, as is described in Kliger *et al* [10]
- A causality graph is a representation of cause ('Problems', which equate to incidents or root causes) and effect (Symptoms, which equate to events).
- In principle at any given time, the state of the system is deduced from the events (Symptoms) present, by selecting the minimal distance to a known problem. In the codebook in Table 1, the presence of S1,S2 but not S3 (event vector $(1, 1, 0)$ would indicate P1 as the most likely root cause.
- Still suffers from ambiguity (consider for example the event vector $(1, 0, 1)$.

Table 1: Codebook for Causality Graph in Figure 2, reproduced from [7]

Symptom	P1	P2	P3
S1	0.8	0.3	0.0
S2	0.9	0.1	0.0
S3	0.0	0.9	0.9

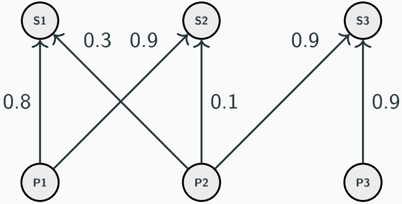


Figure 2: A Simple Codebook Causality Graph - reproduced from [7]

Fundamental Flaws in Rules; Graph Theoretical Viewpoint

Regardless of Approach - 5 Fatal Flaws

1. **Null Event:** A common use case is the non-occurrence of an event in a given timeframe. This is hard to represent as a static transition in a causality graph and corresponds to a non-deterministic structure of the state graph.
2. **Degenerate Causes:** Often different causes can cause the same set of alerts, or in the case of a causality graph two different symptom nodes can have the identical set of edges to a problem node. This is a type of graph 'automorphism' and contributes to a higher entropy of the graph (see next section). The state of the graph is not deterministic and root causes are not definable.
3. **Order of Arrival:** Because events cannot be guaranteed to arrive in a fixed order the causality graph will need to have each possible path from symptom to node. This will eventually result in a perfect graph and complete indeterminacy of causes from symptoms.
4. **Completeness:** Unless the causality graph completely covers all possible transitions the arrival of an unknown event will not be able to be processed by the graph. This is equivalent to adding disconnected nodes to the graph.
5. **Max Sat and Closure:** A well known *NP*-Hard/Complete problem is determining whether an arbitrarily large boolean in normal form has a solution. This is equivalent to knowing the maximum adjacency set of an arbitrary causality graph, also an *NP*-Hard/Complete problem.

Application 2: Entropy and Network Resilience

Graph Entropy and Network Resilience

What is Graph Entropy $H[G(V, E)]$

- Measures structural *information* in a graph. The more meshed a graph, the lower the entropy
- **Chromatic Entropy**
  - Defined using Chromatic number of the graph. Acts like **"negentropy"**
- **Körner or Structural Entropy**[1]
  - Closely related, uses non adjacent sets of vertices

[1] We assume in our treatment that vertex emission probabilities are all uniform

Graph Entropy a Measure of Redundancy

S_4 K_4 C_4 P_4

Table 2: Graph Types that Maximize and Minimize Entropy[2]

	Chromatic	Structural
Maximum	K_n	S_n
Minimum	S_n	K_n

[2]In all of our work we only consider connected, simple graphs

Global Measures Computationally too Complex

- It is only valid globally, no value for an individual node
- All are expensive to compute, and contain NP-complete problems

We need a vertex value such that $H[G(V,E)] \sim \sum_{v \in G} H(v)$

The Dehmer Vertex Informational Functional

- Dehmer ([11]) creates a framework for calculating graph entropy in terms of vertices
- Dehmer scopes locality of a node by the $j - Sphere$, $S_j(v)$ of a node v. This is all of the nodes j links distant to v. We include v (Dehmer doesn't) and constrain our analysis to $j = 1$.
- Introduces vertex information functional $f_i(v)$ of a node v, with vertex *probability* defined as

$$\boxed{p_i(v) = \frac{f_i(v)}{\sum_{v \in S_j(v)} f_i(v)}}$$

- Node entropy $H(v) = -p_i \log p_i$, and total graph entropy $H(G) = \sum_{v \in G} H(v)$

Introducing the Local Vertex Entropy VE and VE'

- We define an inverse degree entropy for a node $VE(v)$ as:

$$p_i(v_i) = \frac{1}{k_i} \text{ where } k_i \text{ is the degree of } v_i,\ VE(v_i) = \frac{1}{k_i}\log_2(k_i)$$

- And fractional degree entropy of a node $VE'(v)$ as:

$$p_i(v_i) = \frac{k_i}{2|E|},\ VE'(v_i) = \frac{k_i}{2|E|}\log_2\left(\frac{2|E|}{k_i}\right)$$

- These two measures do not take into account high degree nodes which are redundantly connected into the graph

Not all High Degree Nodes are Equal!

- To capture importance more accurately we suppress entropy for highly meshed nodes
- A highly meshed network has local similarity to the perfect graph K_n. The modified [3] clustering coefficient C_i of the neighborhood of a vertex i scales our metrics as:

$$\boxed{C_i = \frac{2|E^1(v_i)|}{k_i(k_i+1)},\ NVE(v) = \frac{1}{C_i}VE(v) \text{ and } NVE'(v) = \frac{1}{C_i}VE'(v)}$$

- And for the whole graphs:

$$\boxed{NVE(G) = \sum_{i=0}^{i<n} \frac{(k_i+1)}{2|E^1(v_i)|} \log_2(k_i)]}$$

$$\boxed{NVE'(G) = \sum_{i=0}^{i<n} \frac{k_i^2(k_i+1)}{4|E||E^1(v_i)|} \log_2\left(\frac{2|E|}{k_i}\right)}$$

[3] we include the central vertex in our version to avoid problematic zeros

Comparing NVE and NVE' to Global Entropy Measures

Table 3: Values of Normalized Entropy for Special Graphs

	NVE	NVE'
S_n	$\frac{n}{2(n-1)}\log_2(n-1)$	$\frac{1}{2}\log_2\{2(n-1)\}+\frac{n}{4}$
K_n	$\frac{n}{n-1}\log_2(n-1)$	$\log_2(n)$
P_n	$\frac{3}{4}(n-2)$	$\frac{1}{n-1}+\frac{3n-4}{2(n-1)}\log_2(n-1)$
C_n	$\frac{3}{4}n$	$\frac{3}{2}\log_2(n)$

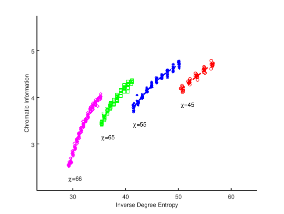
Table 4: Maximal and Minimal Total Vertex Entropy Graph Types

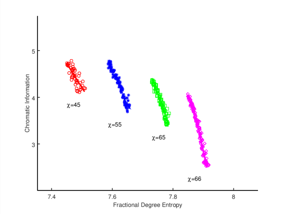
	NVE	NVE'
Maximum	C_n	S_n
Minimum	S_n	K_n

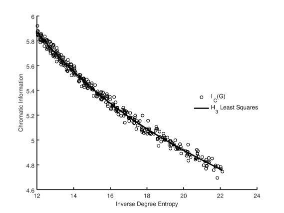
Close inspection of the minima and maxima indicate that NVE' has similar limit behavior to Structural entropy, and NVE to Chromatic entropy

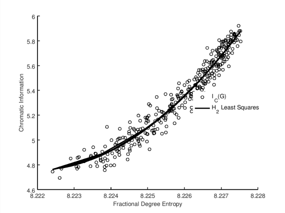
Investigating the Link Between Vertex and Global Entropy

Indeed Random Graph Sampling Highlights Further Correlation









Sampled sum of Vertex Entropies for Scale Free Graphs of constant —V—=300 and $G(N, p)$ Erdős-Rényi Graphs, with $p \in [0.3, 0.7]$ and $|V| = 100$ [12].

Graph Structure and Node Importance

- Network Science demonstrates some nodes are more critical (Barabási-Albert [13], [14])
- Betweenness Centrality is very accurate but expensive.
- Vertex entropy is cheap to compute, and nearly as good!

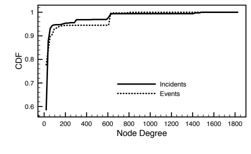


Figure 3: Distribution Events/Incidents by Node Degree [4].

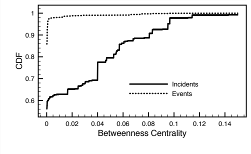


Figure 4: Distribution of Events/Incidents by Node Centrality [4]

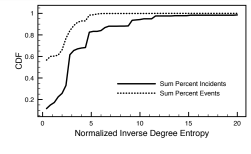


Figure 5: Distribution Events/Incidents by Normalized Inverse Degree [4]

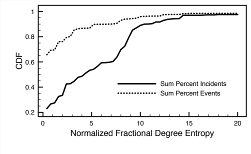


Figure 6: Distribution of Events/Incidents by Normalized Fractional Degree [4]

Vertex Entropy is Almost a Classifier!

- ROC curves for commercial event/incident data at various values of entropy.
- Performance is almost good enough to use as a classifier.

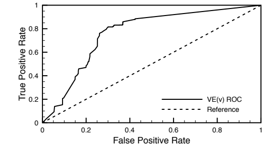


Figure 7: ROC curve for inverse degree entropy

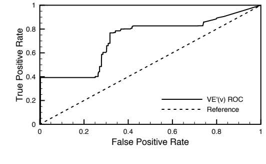


Figure 8: ROC curve for fractional degree entropy

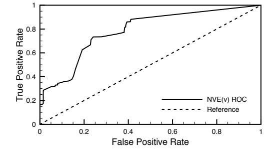


Figure 9: ROC curve for normalized inverse degree entropy

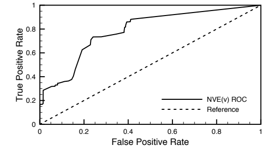


Figure 10: ROC curve for normalized fractional degree entropy

Vertex Entropy Take Aways...

- Effective alternative to heavy algorithms to compute centrality or entropy that are either *NP*-complete, or just plain expensive
- Combines combinatorics with information science to capture the complexity of a network and identify where failures are likely to cause impact
- As tested against commercial networks, VE is both effective as an event "conditioner", and *almost* good enough as a fault localizer in its own right!

What I left out...

Graphs, Graphs, Graphs

- There are many other approaches, that fundamentally boil down to graphs...
- A Neural Net, is essentially a directed signal propagation graph with weights numerically solved to present a known output for a known input. In some circumstances (Hopfield Network, Finite Boltzmann machines [15]) this is achieved by solving a graph Hamiltonian. The use of Neural Nets is another practical application of graphs!
- My colleague Dr Harper will be presenting a time based correlation technique that utilizes modularity detection in a graph obtained from a similarity matrix, again exploiting the correspondence between matrices and graphs...
- Older Bayesian Network [5] and Fault Propagation inferencing approaches rely upon representing causality as a directed acyclic probability graph (standard reference Pearl [16]). These approaches are not commercially popular and in general are *NP*-Hard computationally, and so are omitted.

Conclusion

- Graphs underpin many structures in fault localization techniques.
- They are the natural way to capture dependency of computer networks.
- The powerful correspondence between graphs and Linear Algebra makes them particularly useful!
- The structure of a graph can represent either the logic used to determine root cause, or alternatively the structure of the managed system.
- The powerful mathematics of Graph Theory, Information science, and Network Science can be used to provide novel new approaches to fault localization...
- All of this underpins the "Unreasonable Effectiveness of Graph Theory" for fault localization...

Acknowledgements

I would like to thank my colleagues Rob Harper and Will Cappelli for useful comments on the preparation of the talk. In particular thank you Rob for an excellent and fruitful collaboration over 20 years. In addition we have many friends and colleagues at Sussex, Oxford, Bath, ASU and UCSF. We have benefitted enormously from their help, friendship and collaboration.

I also have to thank my wife Christine for her tireless support of both careers!!!

Summary References

[1] E. P. Wigner, "The unreasonable effectiveness of mathematics in the natural sciences. Richard courant lecture in mathematical sciences delivered at New York University, May 11, 1959," *Communications on Pure and Applied Mathematics*, vol. 13, no. 1, pp. 1–14, 1960.

[2] P. Tee, I. Wakeman, G. Parisis, J. Dawes, and I. Kiss, "Constraints and Entropy in a Model of Network Evolution," *ArXiv e-prints*, dec 2016. [Online]. Available: https://arxiv.org/abs/1612.03115

[3] P. Tee, G. Parisis, and I. Wakeman, "Towards an Approximate Graph Entropy Measure for Identifying Incidents in Network Event Data," no. AnNet, pp. 1049–1054, 2016.

[4] P. Tee, G. Parisis, and I. Wakeman, "Vertex entropy as a critical node measure in network monitoring," *IEEE Transactions on Network and Service Management*, vol. PP, no. 99, pp. 1–1, 2017.

References ii

[5] M. L. Steinder and A. S. Sethi, “A survey of fault localization techniques in computer networks,” *Science of Computer Programming*, vol. 53, no. 2, pp. 165–194, nov 2004.

[6] A. Dusia and A. S. Sethi, “Recent Advances in Fault Localization in Computer Networks,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 3030–3051, 2016.

[7] P. Tee, “The role of graph entropy in fault localization and network evolution,” Ph.D. dissertation, 2017. [Online]. Available: http://sro.sussex.ac.uk/71052/

[8] *Service Operation Book*. The Stationery Office, 2007. [Online]. Available: https://www.itil.org.uk/so.htm

References iii

[9] R. Harper and P. Tee, "Cookbook, a recipe for fault localization," in *IEEE/IFIP Network Operations and Management Symposium: Cognitive Management in a Cyber World, NOMS 2018*, 2018, pp. 1–6.

[10] S. Kliger, S. Yemini, and Y. Yemini, "A coding approach to event correlation," *... Network Management IV*, 1995.

[11] M. Dehmer, "Information processing in complex networks: Graph entropy and information functionals," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 82–94, 2008.

[12] P. Tee, G. Parisis, I. Wakeman, and L. Berthouze, "Relating Vertex and Global Graph Entropy in Randomly Generated Graphs," *Entropy*, vol. 20, no. 7, p. 481, 2018.

[13] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Review of Modern Physics*, vol. 74, no. January, 2002.

[14] R. Albert, H. Jeong, and A. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–82, 2000.

[15] W. Kinzel, "Neural Networks: An Introduction. Von B. Müller und J. Reinhardt, Springer Verlag Berlin 1990, 266 S., 83 Abb., geb. DM 78,-," *Physik in unserer Zeit*, vol. 22, no. 5, pp. 229–229, 1991. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/piuz.19910220514

[16] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.