

2016-01-01

Information Entropy Measures Applied To Hierarchial Complex Technical And Socio-Technical Systems

Satya Aditya Akundi

University of Texas at El Paso, saakundi@gmail.com

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Education Commons](#), and the [Electrical and Electronics Commons](#)

Recommended Citation

Akundi, Satya Aditya, "Information Entropy Measures Applied To Hierarchial Complex Technical And Socio-Technical Systems" (2016). *Open Access Theses & Dissertations*. 796.

https://digitalcommons.utep.edu/open_etd/796

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

INFORMATION ENTROPY MEASURES
APPLIED TO HIERARCHIAL
COMPLEX TECHNICAL AND SOCI-TECHNICAL SYSTEMS

SATYA ADITYA AKUNDI

Doctoral Program in Electrical and Computer Engineering

APPROVED:

Eric D. Smith, Ph.D., Chair

Tzu-Liang (Bill) Tseng, Ph.D.

Virgilio Gonzalez, Ph.D.

Arturo Olivarez, Ph.D.

Scott Starks, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Copyright ©

By

Satya Aditya. Akundi

2016

Dedication

*I dedicate this dissertation research to my beloved parents, **Akundi Murali Krishna** and **Akundi Kusuma Gowri**.*

*I would also like to especially acknowledge and express my gratitude to **Dr. Eric D. Smith** for his continued guidance and supervision in completing this endeavor.*

INFORMATION ENTROPY MEASURES
APPLIED TO HIERARCHIAL
COMPLEX TECHNICAL AND SOCI-TECHNICAL SYSTEMS

By

SATYA ADITYA. AKUNDI

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

AUGUST 2016

Abstract

A significant increase of Systems-of-Systems (SoS) is currently observed in the social and technical domains. With increasing constituent components, Systems of Systems are becoming larger and more complex. Recent research efforts have highlighted the importance of identifying innovative statistical approaches for analyzing complex systems to better understand how they work. This research is aimed towards developing an Information Entropy based framework to analyze complex technical and social systems. Entropy in terms of information theory can be seen as the expected amount of information observed in an event. A parallel can be drawn between information entropy and system complexity, where, as a system evolves or changes its state, the information entropy will also change, thereby identifying entropy in terms of the systems components and their interactions.

Towards the research goal of identifying a framework and characterizing system complexity with information entropy, work has been carried out in exploring the potential application of entropy in three different application areas to illustrate its applicability and to establish the use of information entropy within the broad horizon of complex systems. The case studies identified in the application areas used in this research help to lay a basic foundation for identifying a framework geared towards characterizing complexity and criticality, in order to analyze and assess complex systems in different operational domains.

Table of Contents

Abstract.....	v
List of Figures.....	viii
List of Tables	xi
Chapter 1 - Entropy	1
Introduction.....	1
Entropy – Thermodynamic Sense.....	2
Entropy – Statistical Sense	3
Entropy – Disorder Sense	3
Entropy – Information Sense	4
Entropy according to Thermodynamics.....	5
Entropy according to Statistical Mechanics	6
Entropy according to Information Theory	7
Motivation.....	17
Research Methodology	20
Document Organization.....	23
Chapter 2 – Application of Entropy towards Design Structure Matrix Clusters	26
Design Structure Matrix (DSM)	26
DSM Categories.....	28
DSM Analysis.....	29
Example- U-Print 3D Printer	31
Application of Entropy towards DSM Clusters	39
Proposed Methodology for Calculating Clustering Entropy	39
Chapter 3 – Entropy Based Complexity Measure For Software Based Control Flow Graphs.....	53
Software Engineering	53
Software Quality and Complexity Measures	55
Concept of Entropy on Software Metrics	57
Proposed Metric Definition	61
Proposed Metric Characterization	63
Proposed Metric Validation	67
Proposed Metric Evaluation	73
Chapter 4 – Application of Entropy in Engineering Education.....	82
Introduction to Engineering Education.....	82
Interactions and Classroom Structures	84
Assessment of Classroom Structures.....	89
Information Processing Theory	92
Concept of Entropy.....	93
So what is Entropy?	95
Application of Entropy to Educational Settings	95
Entropy based Assessment Model	98
<i>Traditional Classroom Setting</i>	101
<i>Peer-based Learning Structure</i>	103
<i>Flipped Setting – Self-Organizing</i>	105
Entropy based Classroom Structural Assessment Framework	107
Chapter 5 – Information Theory Applied Towards Complex Networks.....	110
Introduction to Complex Networks	110

Information Theory in Complex Networks	121
Entropy to understand Interaction Dynamics	121
Transfer Entropy to understand the transfer of information in network over time	123
Network Models Used to Apply Metrics	125
Chapter 6 - Conclusion	174
Case Studies Revisited.....	177
Entropy Applied to Software Based Control Flow Graphs	177
Application of Information Entropy for Classroom Structural Assessment.....	178
Information Theory applied to Complex Networks.....	179
Research Questions Explored	182
Research Contributions.....	184
References.....	185
Web Pages	197
Term Definitions.....	198
Appendix A.....	199
Appendix B.....	204
Vita	211

List of Figures

Figure 1: Illustration of a simple Carnot cycle	5
Figure 2: Game Evolution for a 2-Dice game	12
Figure 3: Game Evolution for a 4-Dice game	13
Figure 4: Game Evolution for a 10-Dice game	14
Figure 5: Game Evolution for a 100-Dice game	15
Figure 6: State Distributions for 2, 4, 10 and 100 Dice	16
Figure 7: Primary Research Approach	21
Figure 8: Framework identified to understand Complex systems	21
Figure 9: Complexity Classification of domains considered and respective Case Studies	22
Figure 10: Example digraph form.....	28
Figure 11: DSM Models	29
Figure 12: 3D Printer Decomposition.....	33
Figure 13: 3D Printer Component Interaction Map.....	35
Figure 14: DSM Showing interactions across components of a 3D printer	36
Figure 15: Clustered DSM with corresponding network map.....	37
Figure 16: Decomposition of Probabilities.....	42
Figure 17: Cluster Analysis (adapted from Eppinger et al., pg.26, fig 2.6)	44
Figure 18: Probability Decomposition for 2 Non-Overlapping clusters in Figure 17.....	47
Figure 19: Probability Decomposition for 2 Overlapping clusters in Figure 17	48
Figure 20: U-Print® SE personal 3D printer clustering analysis based on its un-clustered DSM	50
Figure 21: Software Engineering process & its associated phases	54
Figure 22: Basic primitive definitions (Note: All the starting nodes are assumed to have inflow of 1) ...	64
Figure 23: CFG for Metric Application.....	65
Figure 24: Architectural relation of proposed metric to software complexity characterization (dashed arrow indicates dependency relationship)	67
Figure 25: Two programs with same functionality and different implementations along with their control flow graphs [Venkatachalam et al., 2012].	76
Figure 26: Illustrations of CFG's for (P), (Q) and (P;Q)	77
Figure 27: Program bodies P, Q, R and their respective Control Flow Graphs	78
Figure 28: Program bodies (P; R) and (Q; R) and their respective Control Flow Graphs	79
Figure 29: A Complex Synergistic Classroom	83
Figure 30: Emergent Model.....	85
Figure 31: Traditional Classroom Setting.....	86
Figure 32: A Simple Flipped Classroom Setting (Note: EMA stands for External Media Access).....	88
Figure 33: Student Knowledge Structure	99
Figure 34: Hierarchical Representation of a Traditional Classroom Structure	102
Figure 35: Hierarchical Representation of an ideal Peer based Learning Environment.....	104
Figure 36: Representation of Student based Flipped Learning Environment.....	107
Figure 37: Entropy based Classroom Structural Assessment Framework (ESAF)	109
Figure 38: A simple network consisting of 7 nodes and 11 edges	110
Figure 39: Abstract fundamental representation of Complex network	114
Figure 40: Scale Free Network of 500 nodes and its corresponding degree distribution.....	115
Figure 41: Scale Free Network of 1000 nodes and its corresponding degree distribution.....	115

Figure 42: Social Network of online email communication among HP employees [Huberman and Adamic, 2004]	116
Figure 43a: Facebook Network with 826 Nodes, 11361 Edges, & Average degree of 27.508 per node	117
Figure 43b: Facebook Network with 456 Nodes, 6163 Edges, and Average degree of 27.031 per node	117
Figure 44: Social Networks extracted from official Facebook pages of Hilary Clinton (left most in the figure) and Donald Trump (right most in the figure).	118
Preferential Attachment Network	126
Figure 45: Emergence of Network based on Preferential Attachment Phenomenon	127
Figure 46: Transfer Entropy values between the nodes of Preferential Attachment Network	129
Figure 47: Dynamic Entropy and R_0 Measures of Preferential Attachment Network Generated	131
Small World Networks	132
Figure 48: Small world Network Emergence generated using Netlogo software	132
Figure 49: Transfer Entropy values between the nodes of Small world Network.....	135
Figure 50: Dynamic Entropy and R_0 Measures of Small World Network Generated.....	136
Team based Assembly Networks.....	137
Figure 51: Transfer Entropy values between the nodes of Team based Assembly Network	140
Figure 52: Dynamic Entropy and R_0 Measures of Team Assembly Network Generated	141
Student Interaction based Network from Facebook platform.....	142
Figure 53: Raw extracted network based on student interaction in the Facebook group	144
Figure 54: Visualization of the network from Figure 50	145
Figure 55: Illustration of student group interactions after omitting the outliers.....	145
Figure 56: Network observed at time-step 1	146
Figure 57: Network observed at time-step 2.....	147
Figure 58: Network observed at time-step 3.....	147
Figure 59: Network observed at time-step 4.....	148
Figure 60: Network observed at time-step 5.....	148
Figure 61: Transfer entropy values between the nodes	150
Figure 62: Dynamic Entropy and R_0 Measures of Small World Network Generated.....	152
Figure 63: Screenshot of Netlogo Interface Developed	156
Figure 64: Conceptual view based on which the netlogo model is developed.....	156
Figure 65: Network evolution towards a self-organized critical state.....	158
Figure 66: Graphs illustrating Number of connection per node and the ratio of links appearing to disappearing over the network evolution.....	159
Figure 67 (a, b): Dynamic Entropy Measure of the Networks Generated (30(a)-Graph to the extreme left illustrates entropy change in network of 10 nodes without any perturbations; 30(b)-Graph to the extreme right illustrates the entropy change in network of 10 nodes where 5 external nodes were introduced to create an avalanche).....	160
Figure 68: Transfer Entropy Values across Contiguous Nodes.....	160
Figure 69: Node TE values of network generated ((A total of 15 Nodes over 316 time-steps considered)	161
Figure 70: TE Values among Network Nodes (A network with 10 initial nodes where a set of 5 new nodes are introduced twice in the network over its evolution for 880 time-steps).....	162
Figure 71: Information available mapped across the average node degrees observed (Green Line represent preferential attachment network, and Blue line represents Small World network)	164
Figure 72: Information available mapped across the average node degrees observed (Blue Line represents the network generated where perturbations are introduced, and Red line represents the generated network free of perturbations).....	164

Figure 73: Domains of heuristics and biases in the Expected Value Model as described by Smith [2006].	166
Figure 74 (a, b): Subjective Utility and Subjective Probability as defined by Prospect Theory [Smith, 2006]. (74 (a) - Left most figure shows subjective utility versus values based on prospect theory; 74 (b) – Right most figure shows the translation curve of objective to subjective probability values based on prospect theory)	167
Figure 75: Network Dynamics of a simple network with 4 nodes. At every time-step the corresponding network state is represented.	169
Figure 76: Expected utility calculated at every time-step of the network illustrated in figure 75	170
Figure 77: Expected Utility values with reference to corresponding average network clustering coefficient values	171
Figure 78: Netlogo model Interface incorporating Expected Utility Value, Clustering Coefficient, Dynamic Entropy, Average Degree Distributions	172
Figure 79: Principles of Complex System Science used in this Dissertation	175

List of Tables

Table 1: Calculation of Relative Frequency	9
Table 2: Calculation of Relative Frequency	9
Table 3: Characteristic differences of Statistical and Shannon Entropy	10
Table 4: Binary DSM representation of digraph in figure 8.....	28
Table 5: Cluster entropy for Overlapping clusters in figure 8.....	49
Table 6: Cluster entropy for Non-Overlapping clusters in figure 8.....	49
Table 7: Cluster entropy for U-Print® SE personal 3D printer clusters illustrated in figure 18	51
Table 8: Software Quality Metrics [McCall et al., 1977]	55
Table 9: Contributions to Software Complexity Metrics	57
Table 10: Entropy based metric value for CFG.....	66
Table 11: Complexity Metric Values of CFG in Appendix A.....	68
Table 12: Relative Rankings of Complexity Metric Measures in Table 4	69
Table 13: Complexity Metric Values of CFG in Appendix B.....	70
Table 14: Relative Rankings of Complexity Metric Measures in Table 13	71
Table 15: Metric comparison to other complexity measures using Weyukers criteria ('YES' indicates property is satisfies and 'NO' indicates property is not satisfied).....	80
Table16: Classroom Structures and their Associated Interaction Patterns.....	85
Table 17: Example Teaching approaches and Assessment Methods	90
Table 18: Characteristics of complicated and complex networks	113
Table 19: Time series showing the node degrees of the nodes generated at each time-step.....	128
Table 20: TE values calculated based on node degrees from table 2	128
Table 21: Dynamic Entropy and R_0 values for generated preferential attachment network model	131
Table 22: Time series showing the node degrees generated at each time-step of Small world network	133
Table 23: TE values calculated based on node degrees from table 22	134
Table 24: Dynamic Entropy and R_0 values for generated Small world network model.....	136
Table 25: Time series showing the degrees of the nodes generated at each time-step.....	138
Table 26: TE values calculated based on node degrees from table 8	139
Table 27: Dynamic Entropy and R_0 values for generated Team Assembly network model	141
Table 28: Node degree evolution over the time-steps captured. (Note: N1 to N13 represent the different nodes in the network; T1 to T5 represent the time-steps)	148
Table 29: Transfer Entropy values calculated based on table 28	149
Table 30: Centrality Measures of Nodes (Note: Values highlighted in red are the maximum & values highlighted in green are the minimum)	151
Table 31: Dynamic Entropy and R_0 values for captured student interaction network	152
Table 32: Comparisons drawn towards networks from the sandpile model.....	154
Table 33: information theoretic concepts and their application towards complex systems	181

Chapter 1 - Entropy

Introduction

The term “Entropy” was first coined by Rudolph Clausius in 1868. It is rooted from the Greek word “*Entropia*” which means “a change towards something” or “turning towards.” Rudolph Clausius put forth the concept of entropy in classical thermodynamics based on his analysis of the Carnot Cycle, according to which, work is defined as a function of change in temperature. Entropy is defined as “A measure of the possible number of combinations in which a system can be arranged often considered to be a measure of disorder or a measure of progression towards equilibrium” [Freedman, 2003].

Through these fundamental aspects, Entropy was initially explored and applied in physics and statistical mechanics. Entropy later evolved in Information Theory, and was applied in interdisciplinary applications, for example, in Transportation, Networks, Manufacturing, Biology, Economics and Social sciences.

The concept of entropy is often thought of as abstract and also at the same time difficult to present based on its different applications by various authors in many research fields. Throughout the literature, it is observed that many authors mostly seem to consider entropy to be either a state of disorder, or a loss of information. To better understand the distinct sense of the term *entropy*, a principled polysemy approach adapted from the field of linguistics by the authors Haglund, Jeppsson and Stromdahl in their paper “Different Senses of Entropy-Implications for Education” will help the readers to understand the basic co-existence of many meanings of the word Entropy [Haglund et al., 2010]. Five distinct meanings of the word entropy were identified by these authors [Haglund et al., 2010], viz. Thermodynamic Sense, Statistical Sense, Disorder

sense, Information Sense and Homogeneity Sense. In this research we use entropy based on information sense however, to get a better understanding of entropy all the five distinct meanings are explored.

Entropy – Thermodynamic Sense

Based on Clausius's study on heat engines according to Carnot's principle, Clausius coined the term "*entropy*," closely resembling the word energy. His investigation was based upon a process of the transformation of heat from a body at higher temperature to a body at lower temperature, with some production of work. For heat engines at ideal conditions, Clausius suggested that this process can be reversed. Formulating the Second Law of Thermodynamics, Clausius illustrated that there exists a state function, which he called Entropy, which relates the amount of heat involved in a transformation process to the net work done by a system. To expand upon the thermodynamic sense, Haglund *et al.*, cites the following descriptions provided by Davies [Davies, 2000, Pg.: 51]:

"When a physical process occurs, such as a piston-and-cylinder cycle in a steam engine, it is possible to compute how much entropy is produced as a result."

"In a closed system the total entropy cannot go down. Nor will it go on rising without a limit. There will be a state of maximum entropy or maximum disorder, which is referred to as thermodynamic equilibrium; once the systems has reached that state, it is stuck there."

Based on these excerpts on the thermodynamic sense, entropy can be interpreted as a state of a system that tends to increase towards a maximum value [Haglund et al., 2010]. Please refer to the section "Entropy according to Thermodynamics" in this chapter for more details.

Entropy – Statistical Sense

Developing upon the work of Clausius in thermodynamics, which identifies the term entropy as that a property that increases until it reaches equilibrium; Maxwell, Boltzmann and Gibbs try to explain the underlying reason why entropy increases, thereby giving rise to the statistical sense of entropy. Boltzmann discovered that entropy is related to the number of microstates of the particles in a closed system. Looking at a macroscopic view, the system tends to a state with highest possible number of microstates, thereby reaching maximum entropy. The statistical sense of entropy presumes a probabilistic approach of identifying, counting, and monitoring the transitions between the microstates of the system [Haglund et al., 2010]. Please refer to the section “Entropy according to Statistical Mechanics’ in this chapter for more details.

Entropy – Disorder Sense

Entropy in terms of disorder can be divided into a science and a non-science domain. From the perspective of the science domain, entropy can be defined as:

“A measure of disorder; the higher the entropy the greater the disorder”

“In thermodynamics, a parameter representing the state of disorder of a system at the atomic, ionic, or molecular level; the greater the disorder the higher the entropy”

“Entropy provides a quantity measure of disorder” ([Freedman, 2003] as cited in [Haglund et al., 2010])

“Disorder is designated by a quantity called entropy, which is denoted S ” ([Henriksson, 2001] as cited in [Haglund et al., 2010])

From the excerpts presented above it can be seen that entropy is considered to be a measure of disorder. Refer to [Haglund et al., 2010] on how entropy is looked as disorder in non-science domains. To understand the difference between statistical and disorder sense, the authors

Haglund et al, state that *“Unlike the statistical sense, the disorder sense typically does not prompt for a probabilistic approach, but uses a snapshot of a situation, which analogically speaking, represents one single microstate. Disorder is related to visually salient spatial configurations and messiness that does not take energy distribution into account.”*

Entropy – Information Sense

Entropy in terms of information theory can be seen as the average rate of information added by the next element, calculated by the considering the complete set of symbols and their probabilities. The information sense of entropy according to Haglund *et al* relates to the information needed to produce or interpret a message by using its elements such as digits, symbols, letters, words etc. This entropy model shares the relationship between a message and its constituent elements [Haglund *et al.*, 2010]. To better understand the information sense of entropy, refer to the following text extracted from Shannon’s paper on Mathematical theory of Communication [Haglund *et al.*, 2010]:

“If a source can produce only one particular message its entropy is zero, and no channel is required. For example, a computing machine set up to calculate the successive digits of π produces a definite sequence with no chance element [Shannon, 1948 & Shannon, 2001].”

Contrary to the statistical sense, where characteristics are shared by microstates, entropy in information sense can be used to predict the next or an upcoming message or a symbol in a message based on conditional probabilities, thereby stressing the Probabilistic prediction of the next symbol of a message. Please refer to the section “Entropy according to Information Theory” in this chapter for more details.

The main difference emphasized here is the characteristic of entropy according to statistical sense based upon a system’s description through its constituent elements’ relationships. The

Information Sense and the Disorder Sense do share this property where as the Thermodynamic Sense does not.

Entropy according to Thermodynamics

As previously mentioned, Entropy according to classical thermodynamics has its foundation based on Carnot's cycle. According to Carnot's principle, work is a function of a temperature difference. A system undergoing such an efficient cycle of converting thermal energy to useful work (Carnot's cycle) is known as a Carnot engine. To understand an ideal Carnot engine, let's consider the following example adopted from [Ben-Naim, 2008]. Suppose you have a vessel of volume v containing gas, any fluid or a liquid. Now assume this vessel is sealed with the help of a movable piston.

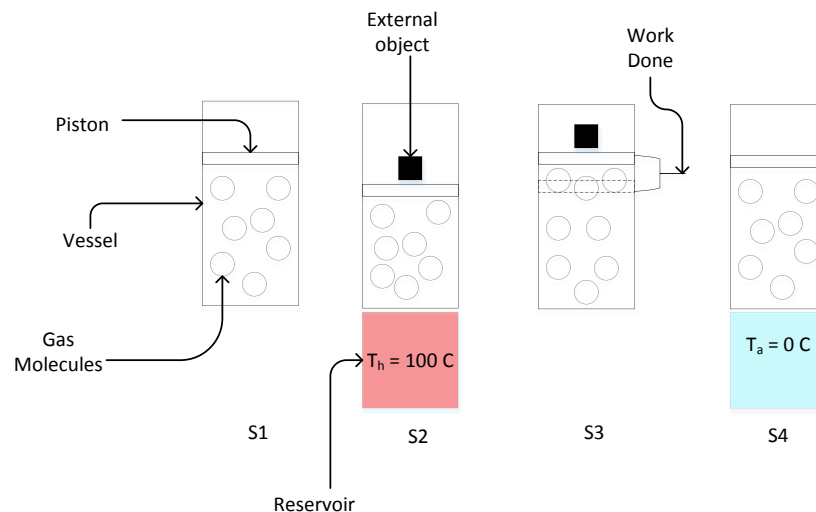


Figure 1: Illustration of a simple Carnot cycle

This system is now initially in State 1 (S1), thermally insulated at temperature (T_a) 0 degrees centigrade. A Carnot engine (in this case, the vessel) consists of the following steps:

- A weight or an object is placed on the piston. This enables the gas to compress and change its state from State 1 (S1) to State 2 (S2)
- Now, the vessel is placed on a heat reservoir at temperature (T_h) 100 degrees centigrade. This enables the thermal energy to flow from reservoir to the vessel, and they gradually attain thermal equilibrium. Now the system will have the same temperature T_h as the reservoir.
- As the temperature of the gas in the vessel increases, the molecules expand and move the piston, thereby doing some work to displace the object placed on the piston.
- To complete a full cycle the system should be brought back to its initial state at temperature T_a that is done by placing the vessel on a reservoir at temperature 0 degrees centigrade.

Clausius based his definition of entropy based on the similar process characterized by the steps mentioned above. The specific definition, according to Clausius, based on a reversible cycle process (thermodynamic equilibrium) is given by the function S which satisfies:

$$dS = \frac{\delta Q}{T}$$

Where; S is the Entropy, Q is the heat of the system and T is the temperature of the system.

Entropy according to Statistical Mechanics

Classical thermodynamic entropy is based on a system considered macroscopically. Ludwig Boltzmann developed the statistical definition of entropy. It is based on his analysis of the microscopic constituents of a system. In statistical mechanics the State variables help to characterize the thermodynamic state of the system also called as a macro state. This macro state

can be characterized upon its constituent microstates such as variables, position of particles, temperature, velocity and similar related attributes. At any given instant there are multiple combinations of microstates characterizing a macro state.

Ludwig Boltzmann established a relationship between macro states, microstates and entropy. Entropy here measures the probability of a system to be spread over different possible combinations of microstates. Based on the characterization by Boltzmann that for a given system achieving its equilibrium is the most probable state, Max Plank formulated entropy as:

$$S = K \log W$$

Where; S is Entropy, K is Boltzmann constant ($1.380 \times 10^{-23} \text{ JK}^{-1}$) and W is the number of possible micro states confirming to a macro state.

A more generalized equation of entropy given by J. Willard Gibbs considering the statistical correlation of distribution in microstates is formulated as:

$$S = -K \sum_i P_i \log P_i$$

Where; K is Boltzmann constant, P_i is the probability of a given microstate. We can see that, the statistical approach on entropy is based upon a probabilistic approach that considers the likelihood of occurrence of different inherent microstates. The multiplication by the Boltzmann's constant, measured in Joules/Kelvin indicates that the result obtained from the above equation is in the units of energy. This illustrates a distinguishable property of the statistical sense of entropy where it is similar to the thermodynamic sense.

Entropy according to Information Theory

Information theory, originally introduced by Shannon in the context to information transfer in communication lines, is used for research in many diverse fields such as linguistics, economics,

psychology and many other areas [Ben-Naim, 2008]. Information theory serves as an anchor to understand the concept of entropy. According to [Cover & Joy, 2012] Entropy based on information theory is defined as “A measure of uncertainty associated with a random variable”. The concept of information entropy was first introduced by Claude E. Shannon in his paper “A Mathematical Theory of Communication” in 1948. According to him entropy quantifies the expected value of information constituent in a message. Information entropy H is usually expressed as:

$$H = - \sum_{i=1}^n P_i \log P_i$$

Where; H is Entropy measure and P_i is the probability that a given message is transmitted. The units of H are Bits, Nats or Bans [Entropy (Information Theory)].

- Bits (if logarithm base 2 is used in the equation) : Basic unit of information which can only have two values, commonly represented using 0 or 1
- Nats (if natural logarithm is used in the equation): Unit of information based on natural logarithms
- Bans (if logarithm base 10 is used in the equation): Bans or Hartley are the units of information theory based on base 10 logarithms

A Text based example of Shannon entropy: Consider the following quotes from Aristotle

“A Friend to all is a Friend to none”.

“He who cannot be a Good Follower cannot be a Good Leader”

To calculate the text entropy of Quote 1:

Table 1: Calculation of Relative Frequency

<i>Word</i>	<i>Frequency</i>	<i>Relative Frequency</i>
A	2	2/9
Friend	2	2/9
To	2	2/9
All	1	1/9
Is	1	1/9
None	1	1/9

The value of H for the given text would follow the guidelines based on the Shannon entropy equation

$$H(\text{Text}_1) = - \left[\frac{2}{9} \log_2 \frac{2}{9} - \frac{2}{9} \log_2 \frac{2}{9} - \frac{2}{9} \log_2 \frac{2}{9} - \frac{1}{9} \log_2 \frac{1}{9} - \frac{1}{9} \log_2 \frac{1}{9} - \frac{1}{9} \log_2 \frac{1}{9} \right]$$

$$H(\text{Text}_1) = 2.503256 \text{ bits}$$

Now, to calculate the text entropy of Quote 2:

Table 2: Calculation of Relative Frequency

<i>Word</i>	<i>Frequency</i>	<i>Relative Frequency</i>
He	1	1/12
Who	1	1/12
Cannot	2	2/12
Be	2	2/12
A	2	2/12
Good	2	2/12
Follower	1	1/12
Leader	1	1/12

Similarly, the entropy can be calculated based on Shannon entropy formulation:

$$H(\text{Text}_2) = - \left[4 * \left(\frac{1}{12} \log_2 \frac{1}{12} \right) - 4 * \left(\frac{2}{12} \log_2 \frac{2}{12} \right) \right]$$

$$H(\text{Text}_2) = 2.91820 \text{ bits}$$

Based on the calculated entropy, it is observed that Text ₂ has more information content comparatively. From this example it can be seen that, entropy referred to as uncertainty by many authors, can also be referred to as *information*.

Exploring the underlying basics of entropy according to information theory and statistical mechanics, which are same in nature, would help to identify their conceptual differences. Understanding these differences would lay a foundation for application of entropy in different fields. Highlighted below are the characteristic differences of their formulation [Information Entropy vs Thermodynamic Entropy, Accessed 2015]:

Table 3: Characteristic differences of Statistical and Shannon Entropy

STATISTICAL ENTROPY	SHANNON ENTROPY
$S = K \log W$	$H = - \sum_{i=1}^n P_i \log P_i$
Logarithm function used in the equation helps to reduce the numerical quantities by scaling them with a constant factor.	The logarithm function in the equation identifies the importance of transmission symbols required to represent a source information
The base of logarithm has no significant meaning	The base of logarithm signifies the information carried by a single message
This measure multiplies the Boltzmann constant to the natural logarithm of the number of particles.	This measure multiplies the probability of being in a source state with the logarithm of the probability of being in the source state.
Assumes all the microstates are equally probable	Assumes all the messages are equally probable
Time considered over here is taken over a limit extending to infinity	Time here is considered to be at a given spontaneous point of time

Association of the term Entropy with number of configurations and probabilities based on the particle dynamics is unquestioned. However, with different coexistent meanings of entropy, the probabilistic relationship of the number of states in a system is not sufficient as it give rise to the following set of assumptions [Ben-Naim, 2008]:

- A system with many number of particles is associated with many microstates
- All these micro states are equally likely i.e. they have equal probability of occurrence
- At equilibrium, the associated microstates of the system are consistent.

These assumptions give rise to the following questions:

- What is it that is changing in a system when it emerges towards equilibrium?
- Why is it changing?

To further explore the concept surrounding Entropy, and answer these questions, an example of a simple dice game is presented. This example is adapted from “Entropy Demystified: The Second Law Reduced to Plain Common Sense with Seven Simulated Games” [Ben-Naim, 2008] to illustrate how a simple dice game facilitates the reader to understand the concept behind entropy. Consider a fair die where its outcomes are equally likely. Following are the assumptions to be considered:

- The Dice considered are marked in such a way that three of its faces are marked with “1” and the three other faces are marked with “0”. One can contemplate this to a coin with a fair probability of occurrence of either heads or tails (in this case 0 or 1).
- This assumption makes the game easier, where, during an experiment the outcome of the dice would be either 0 or 1 with an equal probability of $\frac{1}{2}$ instead of six possible outcomes.

An experimental run with the dice is bounded by the following rules:

- The game always starts with the initial configuration of all zeros
- A die is chosen at random to be thrown
- It is then thrown and returned to its place

- The outcome of the die would be either 0 or 1 with equal probabilities
- The sum (function of number of ones) based on the output of each die is captured

These rules help to run the experiments with different configurations, thereby facilitating the reader to understand the system (in this case, a system of dice) evolution. This game is simulated using Microsoft Excel software for 4 different configurations.

Configuration 1

In this configuration, 2 dice are considered where; the minimum sum would be Zero and the maximum sum Two. Figure 2 illustrates the plot of the sum based on the dice outcome and the number of experimental runs for two different trials. It can be observed for both trials that, over 100 experimental runs, the graph visits Sum=0 and Sum =2 almost equally.

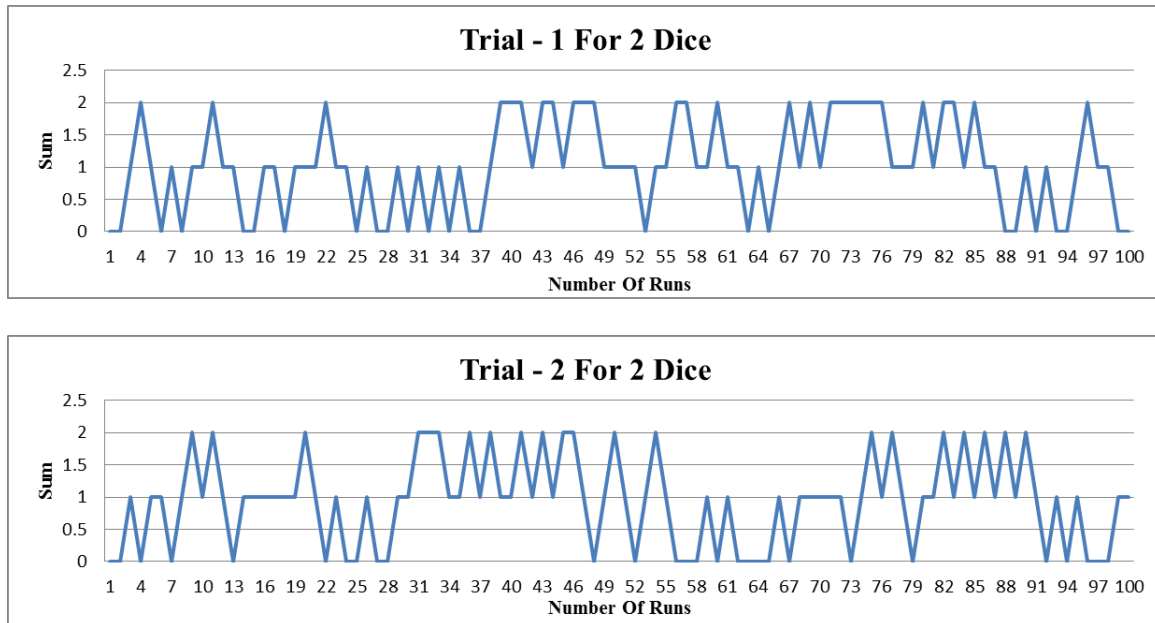


Figure 2: Game Evolution for a 2-Dice game

Configuration 2

In this configuration, 4 dice are considered where; the minimum sum would be Zero (All zeros) and the maximum sum Four (All ones). Figure 3 illustrates the plot of the sum based on the dice outcome and the number of experimental runs for two different trials. It can be observed for both trials that, over 100 experimental runs, the number of visits to Sum=0 is comparatively smaller to configuration 1.

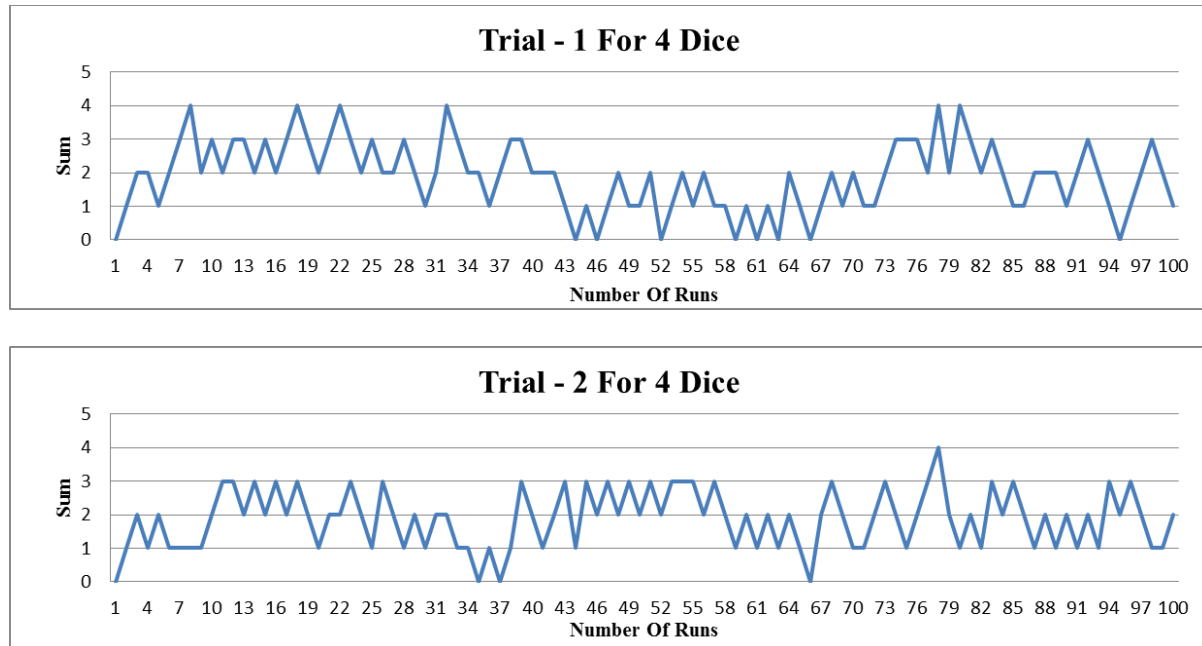


Figure 3: Game Evolution for a 4-Dice game

Configuration 3

In this configuration, 10 dice are considered where; the minimum sum would be Zero (All zeros) and the maximum sum Ten (All ones). Figure 4 illustrates the plot of the sum based on the dice outcome and the number of experimental runs for two different trials. It can be observed for both trials that, over 1000 experimental runs the overall trend of the game shifts up towards an equilibrium at $N/2$ where, N = number of Dice used in the game. These observations help to understand why the number of visit to the initial state of all zeros in the illustrated graphs

decreases as N increases and why the game outcome shifts to $N/2$. This phenomenon can be better observed in the next illustrated configuration.

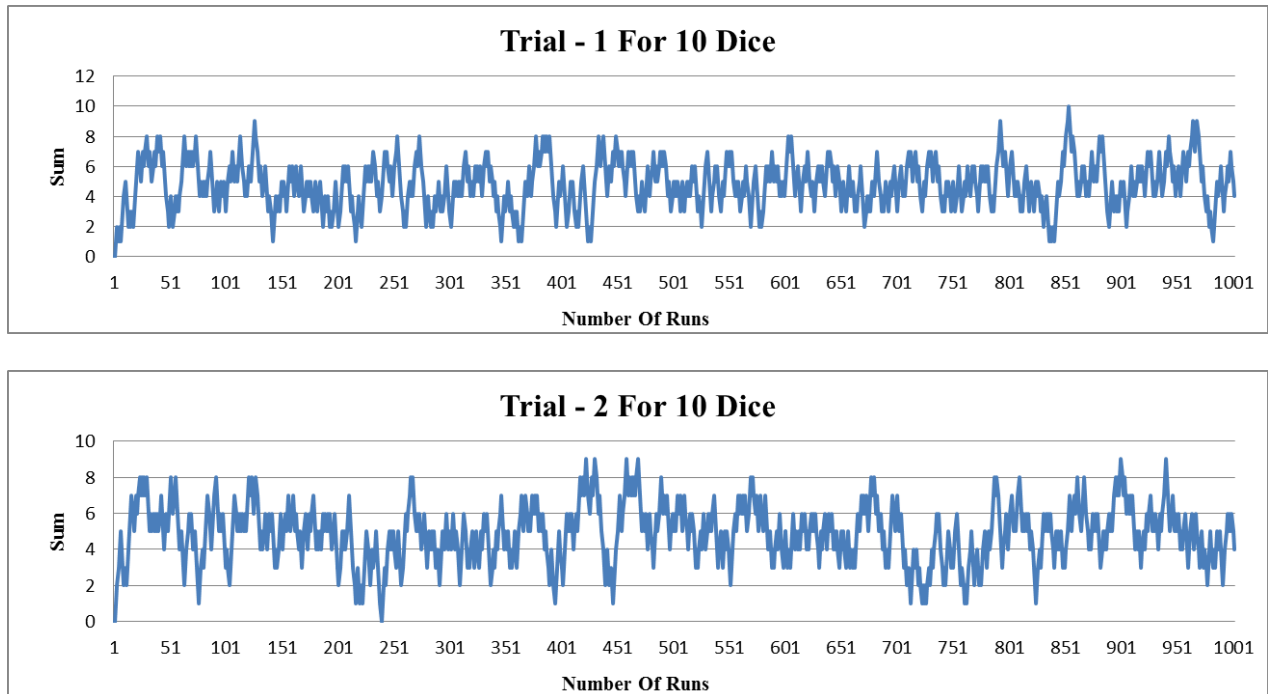


Figure 4: Game Evolution for a 10-Dice game

Configuration 4

In this configuration, 100 dice are considered where; the minimum sum would be Zero (All zeros) and the maximum sum Hundred (All ones). Figure 5 illustrates the plot of the sum based on the dice outcome and the number of experimental runs for two different trials. It can be observed for both trials that, over 1000 experimental runs the overall trend of the game shifts up towards an equilibrium at $N/2$ where, N = number of Dice used in the game.

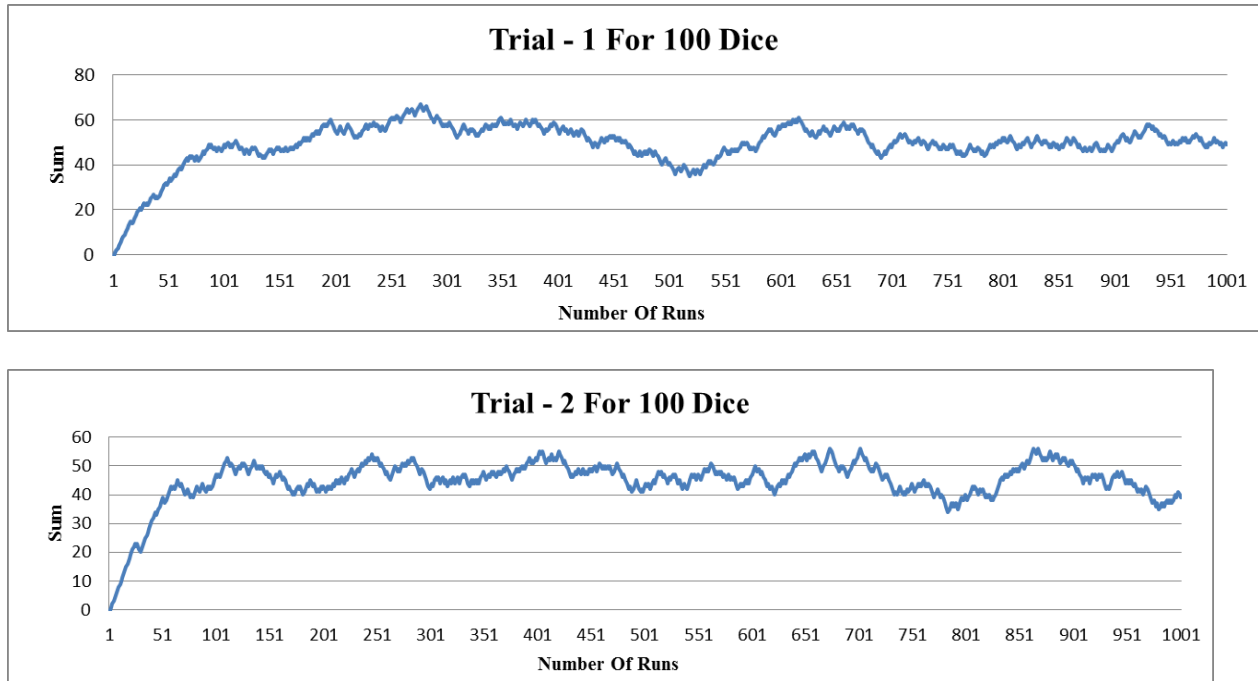


Figure 5: Game Evolution for a 100-Dice game

The two main features that can be observed based on 4 different configurations presented are that the systems initially prefers to shift upwards and then once it reaches equilibrium, it tends to stay there. In other words, sum of the outcomes is largely attracted towards equilibrium line at $N/2$ and it oscillates there. It can be concluded that the equilibrium characterizes the configuration with the largest number of states in a particular game.

Conclusions form the observed configurations based on monitoring how the dice game evolved all the way from 2 Dice to 100 Dice , the fundamental reason why the system changes is that it stays at a higher probability state most of the time and spends less time at a lower probability state. This phenomenon is illustrated in Figure 6.

As observed in the configuration illustrated, when the value of N increases to degrees higher than 10^3 this change in the system behavior becomes certain, which is the essence of the second law

of thermodynamics. This helps to understand why in a spontaneous process, there is an increase in entropy involved (Frequent occurrence of more probable events).

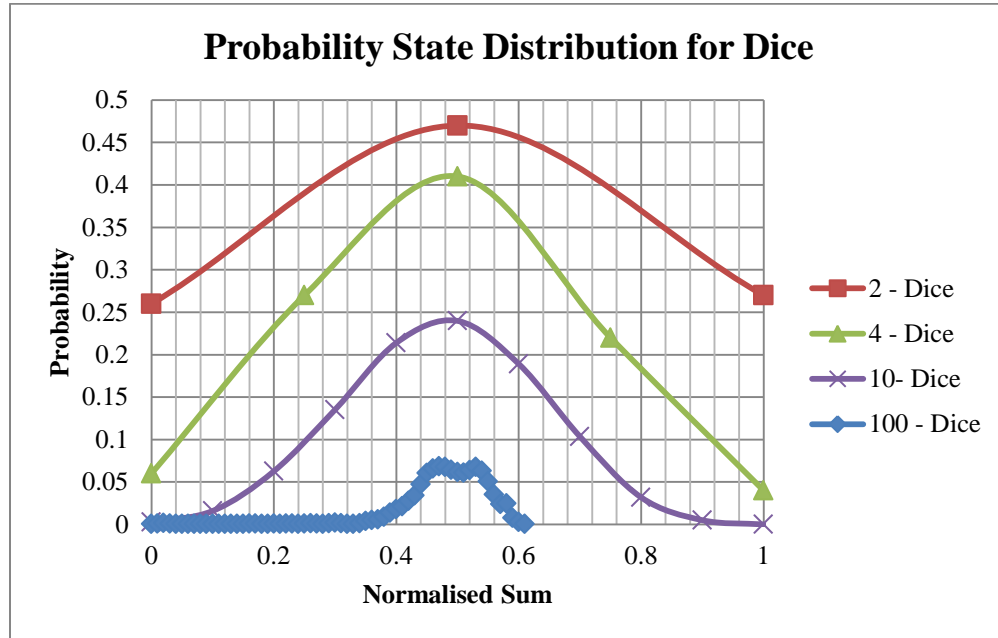


Figure 6: State Distributions for 2, 4, 10 and 100 Dice

This dissertation takes its foundation on Entropy from Information theory, given by Shannon. Information theory plays an important role in understanding the meaning of Entropy. This is because it is a better approach to consider information that can be defined both quantitatively and also subjectively. Also, information can be used as an anchor to understand what is changing in a spontaneous process. It is to be noted that Information theory helps to precisely establish a measure of information defined in terms of probabilities based on some given evidence [Ben-Naim, 2008].

Motivation

Complex Systems are defined as systems that are composed of many independent system elements playing a key role in the whole system's behavior. While the independent system elements follow their own logic and behavior, and interact among themselves, such elements define the dynamic behavior observed in a collectively Complex system. Based on the principles of Complexity Science, the complexity in any system can be characterized by understanding the relations, interactions and the behavior of the constituent system elements. According to Weaver [1948] systems can be formalized by two distinctions, Disorganized Complexity and Organized Complexity. In Weaver's view, disorganized complexity is a result of many constituent system elements. In such a case, the system elements interact among several others, contributing to random system behavior. To understand such a system, Statistical and Probabilistic methods can be used. On the other hand, organized complexity, in Weaver's view, is a non- random behavior of the system elements where the number of parts need not be large for the system to be knowledge emergent. Properties of such a system can be understood with the help of simulations and various modeling techniques. For example, we can observe organized complexity in Self-Sustainable cities and disorganized complexity in the behavior of planets and their orbital rotations.

The whole idea of complex system is still fuzzy based on the fact that it differs in definition, understanding and its idea from author to author according to their perspective [Baranger, 2000].

To provide a few definitions:

“Complex systems exhibit several defining characteristics, including feedback, strongly interdependent variables, extreme sensitivity to initial conditions, fractal geometry, and self-

organized criticality, multiple metastable states, and a non-Gaussian distribution of outputs”
[Kastens et al., 2009].

"Complex systems are those with many strongly interdependent variables. This excludes systems with only a few effective variables, the kind we meet in elementary dynamics. It also excludes systems with many independent variables; we learn how to deal with them in elementary statistical mechanics. Complexity appears where coupling is important, but doesn't freeze out most degrees of freedom"[Boccaro, 2004].

“Complex system: a system with numerous components and interconnections, interactions or interdependence that are difficult to describe, understand, predict, manage, design, and/or change” [Magee & Weck, 2004].

“Complex System can be defined as a system with large number of components, often called agents or constituent system elements, that interact, adapt and learn” [Holland, 2006].

The study increase in complex systems is currently observed, thereby establishing an increased need for the use of statistical methods to analyze complex systems [Baranger, 2000]. In regards to the increase in complexity of systems, as mentioned by Percivall [1994], “Complexity tends to increase as functions and modifications are added to a system to break through limitations, handle exceptional circumstances or adapt to a world itself more complex”. A parallel can be drawn between entropy that is previously discussed in this chapter and system complexity, where, as a system evolves or changes its state, the entropy will also change. This helps to identify entropy in a complex system as a function of the systems components, and their interactions. Supporting this relationship is the paper authored by John J. Johnson IV *et al.* on the theory of System of System Entropy and Emergence.

John J. Johnson IV *et al.* theorize that entropy is a special case applicable to system of systems based on combination of change in information and system disorder, where, the increase in information (energy) leads to an increase in disorder (entropy) thereby causing a system behavior related to an increase in macro and micro system relationship giving rise to emergence. The four main aspects that help to understand entropy in a system of system are Information, Semiotics, Variety, Dispersal and Decay [Johnson *et al.*, 2013].

- *Information*: Information is defined as a commodity that allows the agents to function with other sub systems of the system. It can also be looked at as the energy that enables the sub systems to interact among themselves. Information can take many forms such as social norms, attitudes, and outputs from media sources, feedback and response [Johnson *et al.*, 2013].
- *Semiotics*: For subsystems to interact, the property of information sharing must be existent. The information shared among the sub systems can be characterized by Semiotics i.e. Syntactic Symbols, their Semantic definition and the use of Symbols, allowing the sub systems to recognize and understand information [Johnson *et al.*, 2013].
- *Variety*: Sub systems of a system of system act as regulators to limit the information received and sent that include the desired and undesired outputs. This act of regulation by each sub system acts as a path forward for a system to achieve its objective [Johnson *et al.*, 2013].
- *Dispersal and Decay*: Semiotic aspect of sub systems helps a system of system to gain the capability of receiving and identifying the information. This recognition acts in providing the system of system an ability to increase or decrease the number of interactions [Johnson *et al.*, 2013].

From these four aspects of entropy according to John J. Johnson IV *et al.*, one can observe that the system components (in this case sub systems of a system of system) and their interactions take information as an anchor that drives a change in a system, thus supporting the previously mentioned statement “Entropy in a complex system is a function of the system components and their interactions”.

This illustrates that a realistic statistical approach to analyze and assess a complex system should incorporate a technique that:

- Shall be able to assess the relationships between sub systems/components of a complex system
- Shall be able to provide information to understand a system
- Shall be able to identify the ability of a component/sub system to store and assess the information in a system
- Shall be able to assess and identify the interaction patterns of components/sub systems in a system
- Shall be able to differentiate between the input and output information of a system.

Research Methodology

A systematical approach as illustrated in Figure 7 is used to develop this dissertation document. This helped to explore, address and define complex systems across different complexity profiles. Based on the identified application areas, initial study on understanding the theoretical concept was conducted, which provided insight on how a system works, and in identifying its interdependencies, sub-systems and interactions. Next step included identifying the research aspects that can be further explored, which included exploring areas related to measuring

complexity and identifying the knowledge base that addresses on how to measure complexity of a considered complex system.



Figure 7: Primary Research Approach

The main goal of literature review was to identify previous complexity measures defined for the system based on the concept of entropy. This helped to understand the benefits of previously defined complexity measures and their considered factors based on which the measures were developed.

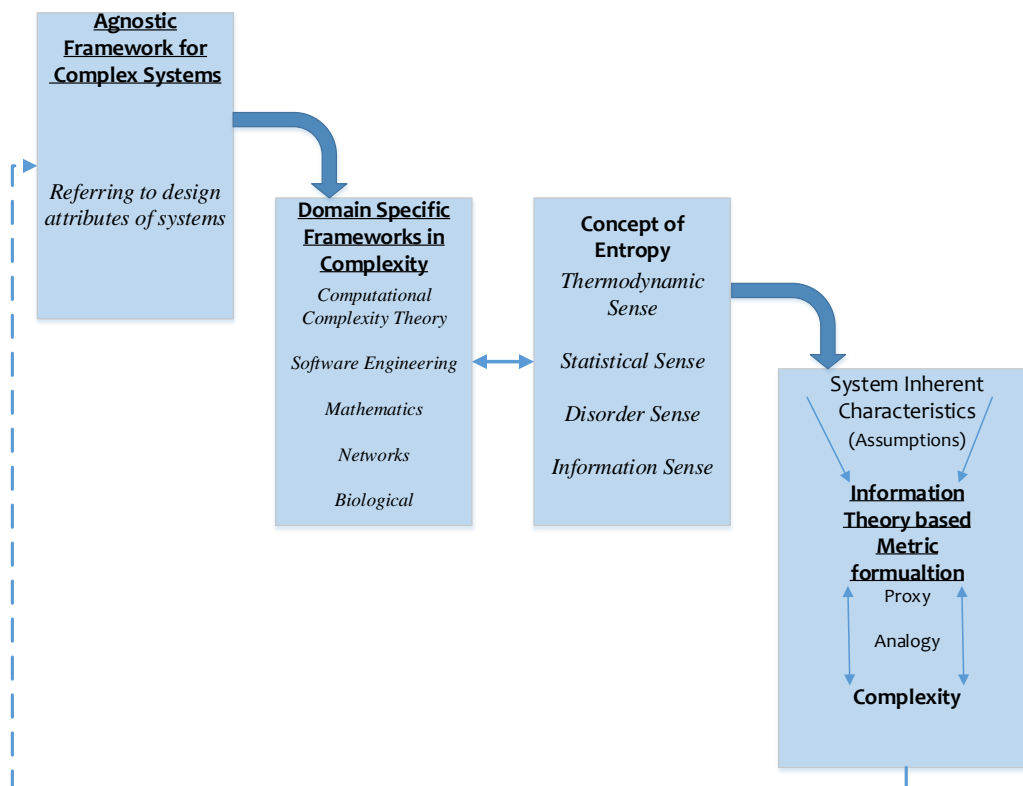


Figure 8: Framework identified to understand Complex systems

Identified in Figure 8 is the framework to enable an understanding of complex systems using the concepts of entropy and information theory. Towards the goal of providing an overarching agnostic framework, domain specific methodologies of what complexity is, were first explored. This involved understanding what complexity is, in specific domains. The case studies identified for this research are Simple Deterministic System (3-D printer) Architecture, Software Based Control Flow graphs, Academic Settings and Complex Networks. Mapped on to a complexity classification graph, the identified case studies are placed across Information and Rigidity. Figure 9 illustrates the case studies from different domains mapped on the complexity classification graph.

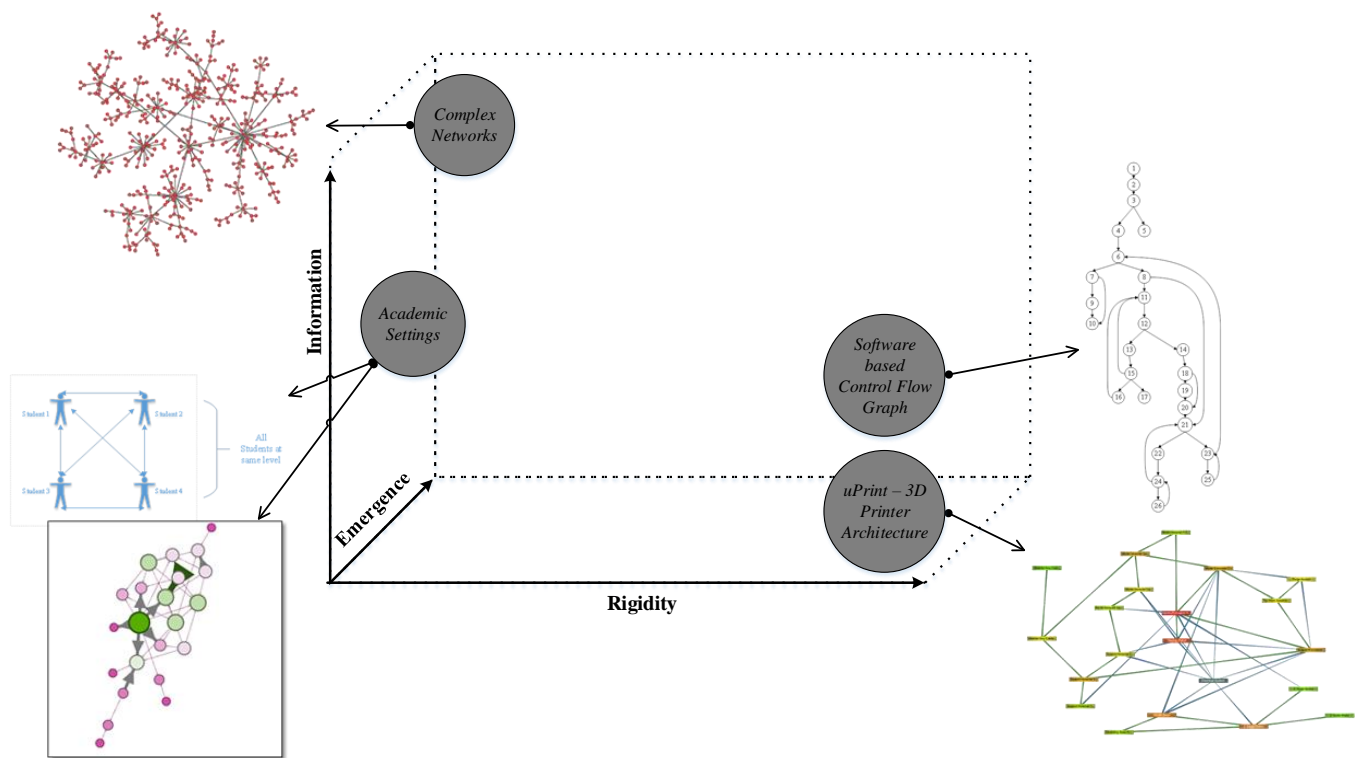


Figure 9: Complexity Classification of domains considered and respective Case Studies

Characteristics of the systems were explored based on their domain specificity that enabled in identifying system dependencies (intra-systems and inter-system). With a thorough

understanding of system characteristics the benefits and drawbacks of the entropy based complexity measures helped confirm the need for exploiting various characteristics of a system to establish new techniques to measure and understand the system. This followed engineering an entropy-based approach specific to the identified research areas of application that are presented in Chapter 2, Chapter 3, Chapter 4 and Chapter 5. The metrics developed using the frameworks are aimed at providing a proxy that reflects to an analogy of what complexity is in the specific system considered. Initial examples of simple case studies to illustrate the application of engineered entropy based measures are presented to explore the realistic and practical application of the proposed measures.

Document Organization

Chapter 2 starts with exploring the concept of Design Structure Matrix (DSM), its definition and its application in Academia and Industry. A simple example on how a design structure matrix is developed based on diagraphs is then presented. Concepts of Static DSM's and Time based DSM's are presented along with a brief review on how static design structure matrices are analyzed. An example of Uprint SE plus 3D printer (available in Industrial Manufacturing and Systems Engineering research lab) is illustrated to map its product architecture on a DSM based on its component dependencies and information exchanges. LOOMEQ, freely available software was used to take advantage of its efficient clustering algorithm in order to identify the clusters based on the product architecture. Around 20 iterations were used to verify the consistency of identified clusters by the software. A brief review on the concept of entropy and its application on DSM are also presented. A proposed methodology on application of entropy towards the clusters in DSM is then developed and illustrated with example.

Chapter 3 of this document introduces the concept of software engineering along with literature review on how complexity is measured in software codes. Entropy and its application towards measuring software complexity are explored. A complexity measure for software codes mapped onto control flow graphs, which considers logical decision-making, processes, software statement interactions patterns, based on the concept of entropy is then defined and illustrated. This proposed measure is validated using well-established McCabe's software complexity measure. An improved complexity measure is also suggested which builds upon the proposed metric by incorporating individual node execution times of each node in a software control flow graph. This metric is then evaluated against nine different axioms that a software complexity measure should satisfy, which were established by Weyuker.

Chapter 4 in this dissertation expands upon current research and techniques related to different classroom settings in academia. This illustrates on how the stakeholders of a classroom interact in traditional and flipped settings. A brief literature on different assessment techniques currently used to measure student learning is presented. Also, a brief review on the concept of entropy and its application in the field of educational assessment is explored. A proposed statistical measure based on the concept of entropy that enables in assessing a given classroom setting is defined and illustrated. Entropy based Classroom Assessment Framework that underlies a stakeholder centric probabilistic model for assessing the state and structure of a complex classroom structure is also developed.

Chapter 5 in this dissertation presents the application of information theory concepts to understand complex networks. It explores what complexity is, in complex networks. Social networks, a special case of complex networks are used to understand how information grows with complexity in networks. The structural importance of the nodes in information processing

of a complex network is also explored. Examples of network phenomenon such as preferential attachment and small world phenomenon are used to illustrate on how the concept of entropy can be used to understand their structural emergence. An example of geographically separated social network is used to illustrate the applicability of entropy measures. Furthermore, the concept of criticality in complex network is explored by comparing a network to a Bak-Tang-Wiesenfeld (BTW) sand pile model.

Chapter 6 presents the conclusions drawn from the case studies used in this research on using information theory to understand at a system and sub system level, the interdependencies among the considered system constituents. Finally, the contributions to the complex systems body of knowledge by this research are presented.

Chapter 2 – Application of Entropy towards Design Structure Matrix

Clusters

Design Structure Matrix (DSM)

The Design Structure Matrix (DSM) methodology is nowadays adapted widely in organizations for engineering complex systems, engineering management and academic research for modeling systems in various areas of applications. The early use of DSM can be traced back to the 1960's where it was used by professor Don Steward from California State University at Sacramento as a part of graph theory to depict the relationship among the nodes in a graph. Later on, this approach was extended for managing systems and characterizing system behavior [Eppinger and Browning, 2012].

DSM has many similarities to various other matrix-based and non-matrix based methods such as dependency maps, adjacency matrices, contribution matrices, reachability matrices and N^2 diagrams, architecture diagrams, directed graphs and dependency models along with many other models extending the science of networks [DSM Web.org, 2014].

DSM, by definition a “Design Structure Matrix,” or, also known as a dependency structure matrix, is the representation of various interactions in a system mapped onto a matrix. These interactions are mapped in a square (N rows x N columns) matrix among the different elements of a system that are being considered, which are listed as the titles of the rows and columns of the matrix, where the elements can take the form of systems, subsystems, physical components, people involved, design activities, parameters and many similar interfaces. The interactions among these components are usually energy flow, material flow, information flow,

communications, component interactions, and team interactions which can be classified as dimensions of energy, material, spatial or information [Ronnie Emile Thebeau, 2004].

According to authors Eppinger and Browning, in their book *Design Structure Matrix Methods and Applications*, DSM is defined as “a network modelling tool used to represent the elements comprising a system and their interactions, thereby highlighting the systems architecture (or a design structure)” (Eppinger and Browning, 2012, p.2).

The use of DSM spans a wide horizon in both academia and industries. DSMs have been applied in Automotive Industries [Pimmer *et al.*, 1994], [DSM Web.org, 2014], Electronics [Eun Suk Suh *et al.*, 2010], [DSM Web.org, 2014], Construction [Schmidt III *et al.*, 2011], Software Design [DSM Web.org, 2014], Aerospace [DSM Web.org, 2014], and Organizations [DSM Web.org, 2014].

A simple DSM example is shown below, based on a digraph form. A digraph is a representation of a set of nodes connected via edges along with the direction of flow associated with them. Figure 10 shows a sample digraph consisting of 5 nodes, 2 feedback and 6 feed forward loops. Also shown is a binary DSM representation of this digraph indicating the presence or absence of interactions, where, the nodes analogous to the diagram are labeled A through E, which are placed across rows and columns. This approach of node placement helps to identify the relations among them.

Reading across row E for example, we see that node E has inputs from node B and node D marked by an ‘X’ in the table. Similarly, reading down column D, we see that the output of node D acts as an input for node B. It can also be observed that, all the feed forward loops are placed below the diagonal and all the feedback loops are placed above the diagonal.

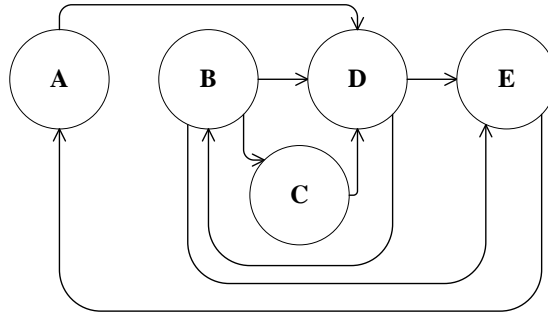


Figure 10: Example digraph form

	A	B	C	D	E
A					x
B				x	
C		x			
D	x	x	x		
E		x		x	

Table 4: Binary DSM representation of digraph in figure 8

DSM Categories

DSM's are mainly classified into 2 categories currently, with many sub-types, depending upon the model being analyzed. Figure 11 illustrates the types of DSM models.

Static DSM's fall under a first category representing all the system models and constituent elements that exist concurrently. Product components (physically interacting with each other) and the organizational group (people communication with each other) architectures are a few examples of static DSM's [Tyson R. Browning, 2001]. Some interaction types that can be captured through mapping a static DSM are Spatial, Energy, Information and Material. Spatial dimension captures the association of physical space between the elements. Energy captures the

need for exchange of energy between two elements, Information captures the exchange of data between two elements and Material signifies the need for material exchange between two elements [Ali A. Yassine, 2004].

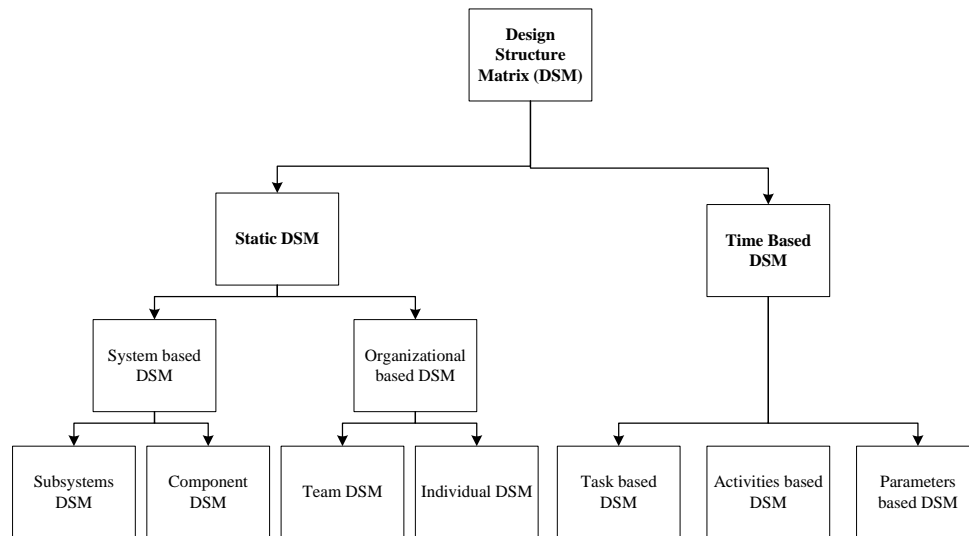


Figure 11: DSM Models

The second category is of Temporal based DSM's. These DSM's are ordered according to models with corresponding rows and columns indicating a flow through time. Types of processes, activities, interactions, and parameter based models along with few software models executing procedurally and any such processes that are actuated over a period of time are represented using this time-based model [DSM Web.org, 2014].

DSM Analysis

Analyzing a model, either time-based or static, after it is mapped onto a DSM helps to understand and characterize its behavior. The algorithms developed are mainly based on the type of DSM being characterized.

Static DSM's are analyzed using a clustering approach that is often geared towards finding optimal subsets/clusters of a model of DSM elements, which are minimally interacting. These clusters are formed in such a way that most of all the interactions of the cluster elements are absorbed within clusters, and the interactions among the clusters are minimized [Ali A. Yassine, 2004]. There are many techniques and algorithms for clustering a DSM that capture a best solution aimed towards minimizing intra-module dependencies and maximizing inter-module dependencies.

Applying clustering algorithms and techniques to Static DSM's of organizational teams helps to capture the highly significant or frequently occurring interactions. Obtaining such interaction clusters will be significant for a framework development to address team characterization focused mainly towards required organizational architectural layers and development. Analogous to organizational teams and people interactions, clustering when applied to product architectures and components will help to capture highly interactive components in a product by helping the product testers to focus on the significant interactions and flow patterns between and among the components, sub-systems and the system.

Eppinger and Pimmler from MIT created a DSM for the Climate Control Division (CCD) of Ford Motor Company to better understand the network of components and their interactions in the climate control system designed for Ford cars and trucks. They captured 16 components using system decomposition and then documented the list of interactions among these components in 4 dimensions (Spatial, Energy, Material and Information) on a scale of -2 (detrimental) to +2 (required). The values for all the 16 components and 4 different interaction types were captured in a composite DSM. A clustering algorithm based on weighed functions was used for all the dimensions of its interactions. Many significant observations were made

based on clustering. One such observation is that in highly integrative chunks, the interactions were observed to be of type spatial and information. Also, no cluster was related to flow of engine coolant through the radiator and the heater core that provides heat to passengers was observed. See Eppinger *et al.* for more information [Pimmler *et al.*, 1994].

Authors Schmidt, Deamer and Austin used DSM Clustering algorithms using Loomeo Software for a UK government initiative to create schools that accommodate the changing demands of the users. DSM clustering allowed them to guide designers and contractors in providing best practices rather than a pre-defined ontology on how a building could accommodate a change. They used brands taxonomy to decompose the buildings into six different layers (Space, Stuff, Space Plan, Services, Skin and Structure) as a guide to group the components. The DSM captured 90 different components decomposed into the 6 layers mentioned. They observed that the structure and space layers were tightly bound whereas the other modules were less bound. The cluster helped to better characterize the components thereby helping to identify the components that need further design alternatives and suggestions for design changes. See Schmidt *et al.* for more information [Schmidt III *et al.*, 2011]

Example- U-Print 3D Printer

As an example, consider the architecture presented below along with its corresponding DSM representing the interaction of various components in a 3D printer. This architecture has been developed based on the user manual of a U-Print[®] SE personal 3D printer manufactured by Stratasys.

The following steps cover the creation of a design structure matrix starting from the 3D printer component listing all the way to cluster identification. To create the DSM, this system was chosen based on the availability of its installation and instruction manuals at the UTEP

Intelligent Systems Laboratory. The DSM created is solely based on perceived interactions among the printer's visible components and based upon user experience. Due to the unavailability of the considered system inner component interaction map and communication architecture, the DSM entries are purely at a high level. If all the detailed functional and sub systems interactions were available, a lot of detailed interactions would be expected along with every individual components functional allocation.

The following steps were followed:

- System (U-Print® SE personal 3D printer) high-level component decomposition based on available installation and operational manual.
- System high-level component interaction identification and analysis.
- Dimensional categorization: Spatial, Energy, Information and Material flow mapping.
- Graph mapping based on component interactions.
- DSM extraction based on the graph created in the previous step.
- Cluster identification based on DSM.

High Level Component Decomposition

Due to the constraints, not all information was captured except for the system high-level components decomposition. U-Print® SE and U-Print® SE Plus personal 3D printers' introductory information manual and printer Assembly Instructions book were used to facilitate this task [U-Print® SE and U-Print® SE plus Personal 3D Printers Information Manual, 2011], [U-Print® SE and U-Print® SE plus Personal 3D Printers Assembly Instructions, 2011]. Figure 12 illustrates a high decomposition diagram.

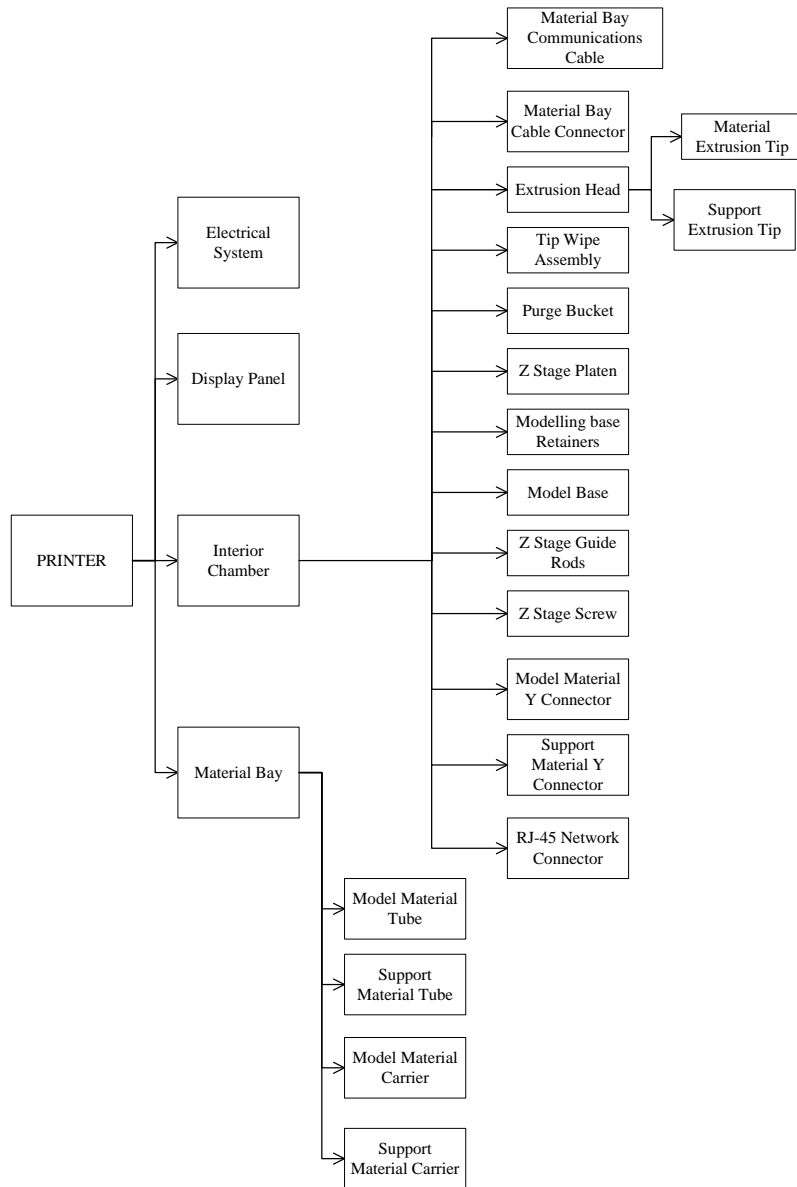


Figure 12: 3D Printer Decomposition

Dimensional Interaction Identification

System decomposition diagram acts as a verification and identification tool for capturing all the component interactions. The motivation behind capturing 4 dimensions of component interactions was based on observing the system operability. The 4 distinct types of flows captured were:

- Spatial Flow (i.e. Physical Interactions)
- Information Flow (i.e. Flow of commands, Information and Functions)
- Energy Flow (i.e. Electricity and Heat Transfer)
- Material Flow (i.e. Transfer of Material among Components)

Graph Mapping

A component interaction map was developed using the Graph functionality of Loomeo® software from TESEON. Figure below illustrates the component interactions mapped. Single interaction edges are colored in blue and double interaction edges are colored in green here. The color-coding of the nodes is based on their active degree. The degree of a node denotes the total number of associated interactions with all the other nodes. The higher the degree of a node, the more active it is perceived to be. In the figure 11 shown, nodes coded red have a higher degree and the nodes with green have the least degree.

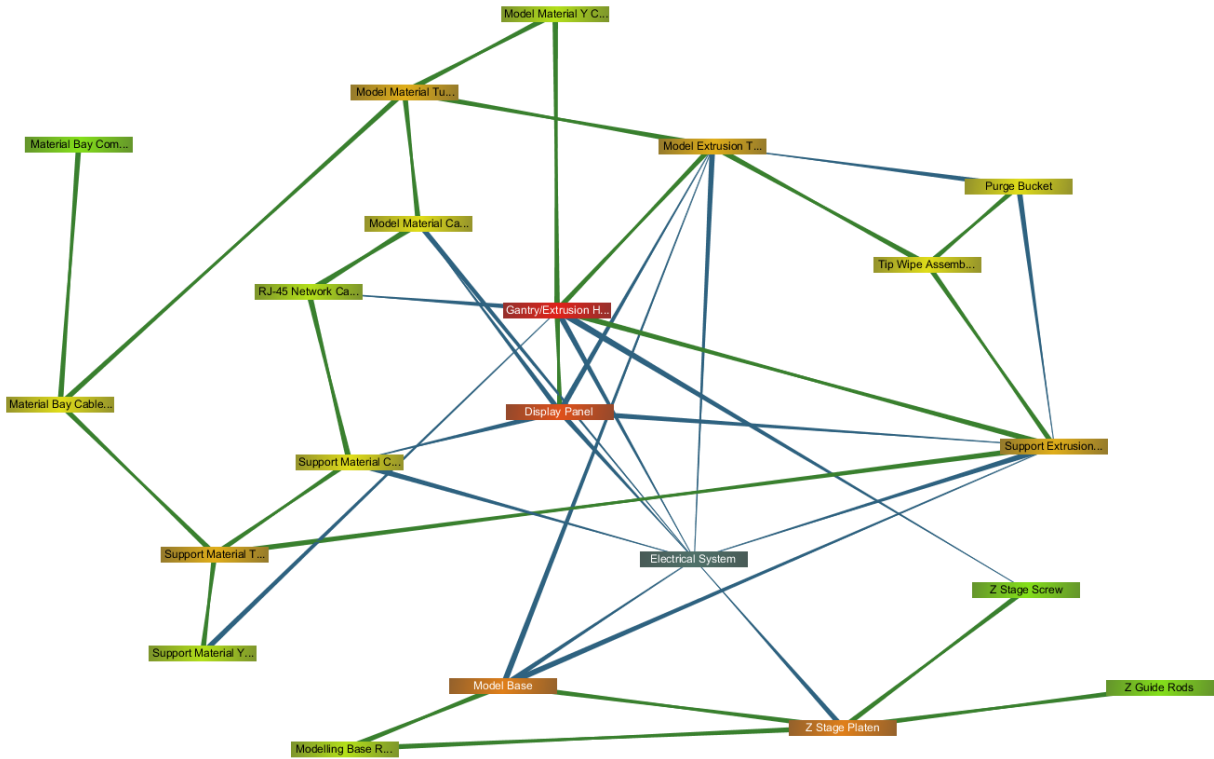


Figure 13: 3D Printer Component Interaction Map

DSM Extraction and Clustering

Representing information and the system component interactions, a DSM model was generated based on the captured network interaction map. The DSM model illustrates a network of 21 components associated with the 3D printer. The DSM model is developed using LOOMEIO. A color-coding scheme was assigned based on the type of interaction between the nodes and to easily track the shift of the nodes while clustering. Figure shows the DSM with the printer components on both the axes of matrix. “X” marks the entries wherever there is an interaction observed. These entries are then color-coded based on their type of interaction i.e. either spatial interaction, Information, energy or material flow.

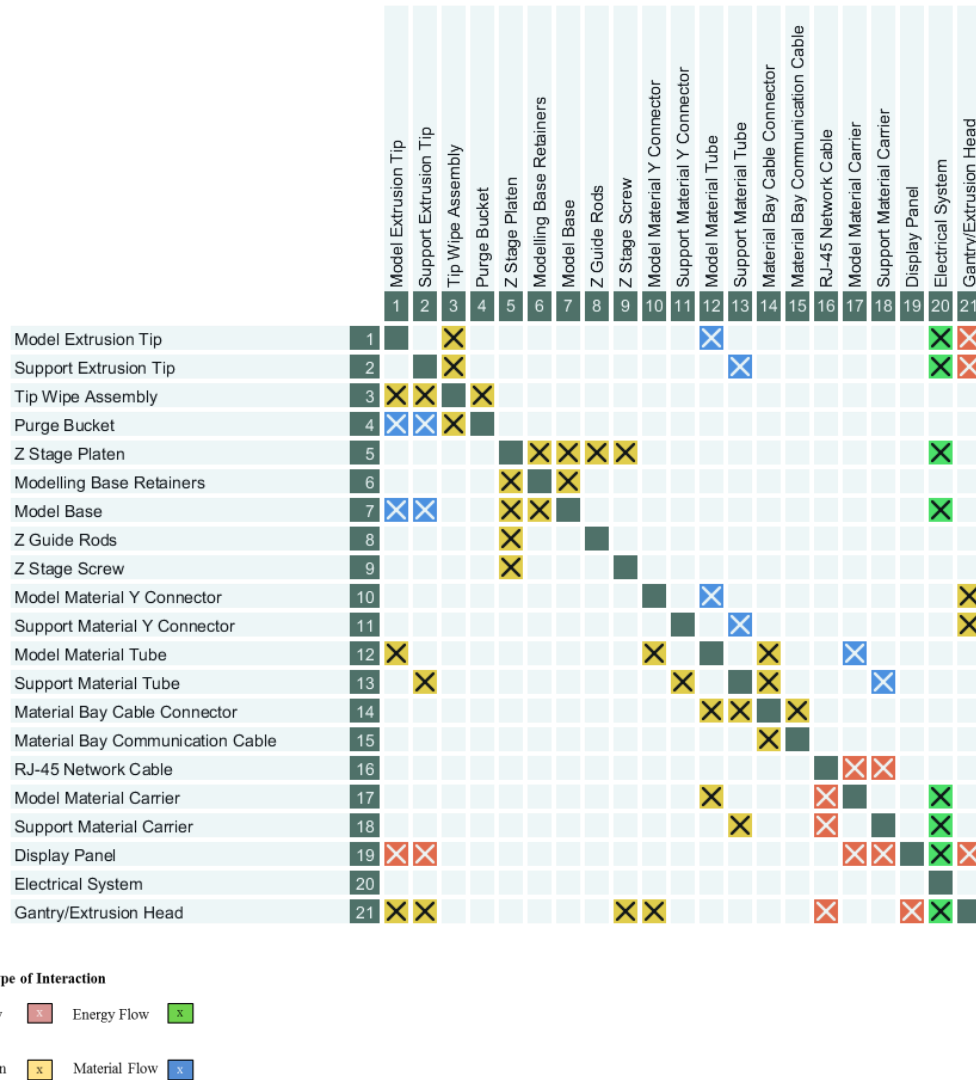


Figure 14: DSM Showing interactions across components of a 3D printer

This DSM was used in Loomeo software to take advantage of its clustering algorithm. Loomeo reorganized the matrix solely based upon the interdependency among the identified components and their interactions. The software for clustering was used to cluster the 21x21 component interactions. The clustering algorithm allows the user specify the required number of clusters the user would like the DSM to be divided into. The result was reiterated for around 20 times for

consistency and to properly identify the constituent clusters. Figure 15 below shows the clustered DSM matrix obtained as a result of the interdependent interactions.

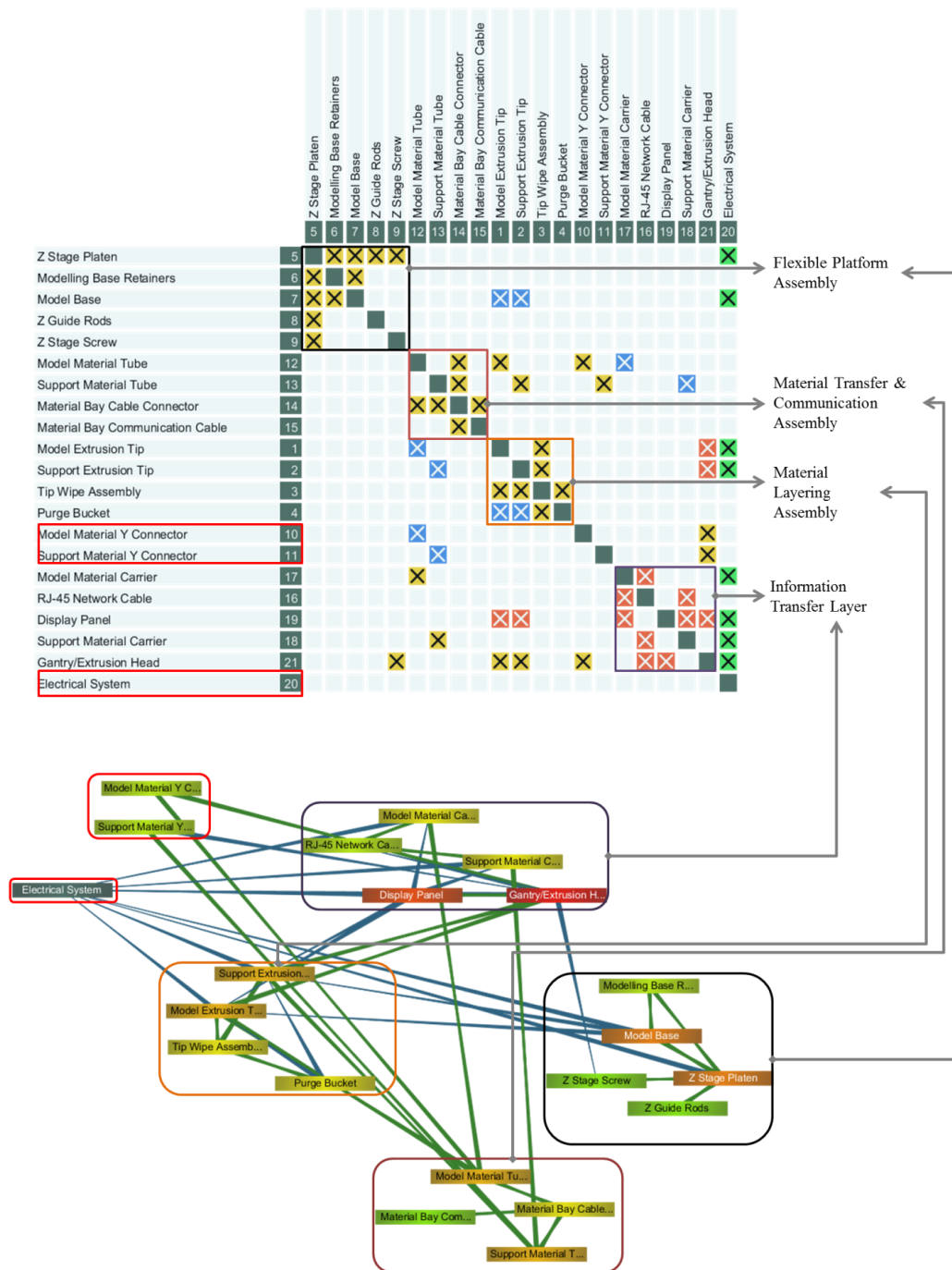


Figure 15: Clustered DSM with corresponding network map

Analysis of the clustered DSM for the most part reveals that identified clusters maintain their integrity in the system. Four tightly bounded modules, Flexible platform Assembly, Material transfer & Communication assembly, Material layering Assembly and Information transfer layers are identified. The highlighted cluster in violet indicates that a beneficial manipulation may be required to be implemented by combining Gantry /Extrusion head module to be combined with information layer. Clustering also identified:

- Components with high interdependencies
- Components that may not be required for system functionality, in this case *Model Material Y Connector* and *Support Material Y Connector*. This conclusion is validated by verifying the printer operation where in which these two components are only required to function in case if an optional material bay is connected to the 3D printer.
- Component with an overarching operation on all the other components. In this case, the *Electrical System* which facilitates the system with energy flow for its full functionality and availability.

This approach towards the exemplification of the system (3D-Printer) helps to verify and validate the clusters accurately representing the system modules and their functionality. This work helps to quickly identify the components thereby providing a framework for further system improvements along with high-level visualization to capture and reflect the underlying design processes.

Application of Entropy towards DSM Clusters

The structure of a clustered DSM includes different dimensions such as, the number of elements being considered for analysis, the number of clusters identified, the number of elements in a cluster, the number of interactions among the elements in a cluster and the interactions among the clusters.

For making rational decisions based on identified clusters in a given DSM, a scientific and statistical tool is necessary to identify the measure of inherent disorder. To address this need for measuring the inherent disorder of these sub systems called clusters of any system considered, entropy, a well-defined and preexisting measure and methodology is used.

Proposed Methodology for Calculating Clustering Entropy

Entropy here is extended to measure the disorder in a system based on the number of clusters identified and the number of inter-component interactions of individual clusters. Below are given the four main steps of the proposed methodology to measure the disorder of a clustered DSM:

1. Identify the system clusters captured using a given clustering algorithm.
2. Find the probabilities of occurrence of each and every cluster identified with respect to the total number of system components.
3. Once clustering probabilities are calculated, obtain the individual component interaction probabilities of a given cluster based on the number of components it is associated with.
4. Map the corresponding probabilities obtained from step 2 and step 3 on to a probability tree.

5. Using a weighed sum approach, which is projected onto a Markov chain, calculate the total Clustering Entropy based on the values obtained from previous steps.

Identifying System Clusters

There are many clustering algorithms being used in industry and academia for clustering static DSM's. The clusters identified using such algorithms are captured in such a way that all the intra-element interactions of a cluster are absorbed in the cluster. These clusters are mainly aimed towards reducing inter cluster dependencies.

The first for identifying system clusters would be map system component interactions in a DSM. This would help to capture the system elements with high interdependencies and also the elements that act as intermediate carriers for interactions among clusters, if any. This snapshot of system component behavior would help in understanding their influence on system functionality thereby enabling us to characterize the disorder.

Probabilities of Cluster Occurrence

This step involves calculating the probability of occurrence of each and every cluster identified in the first step with respect to the number of system elements/components identified for mapping DSM.

Let ' K ' be the total number of components/system elements considered for capturing the DSM.

Let the number of clusters identified by a clustering algorithm be ' n ', where, $N=1,2,3\dots n$ and let C_1 represent cluster 1, C_2 represent cluster 2 and so on let C_n represent cluster n .

Now, let ' r_n ' be the number of elements captured in cluster ' C_n '.

The probability of occurrence ' P_{C_n} ' of a given cluster can be calculated using the following formulation, assuming, if and only if the clusters are considered to occur sequentially, i.e. a random cluster is assumed to be chosen first, then the second and so on.

$$P_{C_n} = \frac{1}{\binom{k-r_{n-1}-r_{n-2}-\dots}{r_n}}$$

Where,

$\binom{k-r_{n-1}-r_{n-2}-\dots}{r_n}$ determines all the possible number of ways the combinations for obtaining a cluster of r_n elements from a larger set of $k - r_{n-1} - r_{n-2} - \dots$ distinguishable elements.

Probabilities of Individual Component Interaction Occurrence

Here the probabilities of interactions for each and every identified element in DSM are calculated with regards to the cluster they are placed in and the total number of interactions captured by that respective cluster.

Considering ' K ' be a vector of k system elements or system components identified for creating a DSM from the previous step, the probabilities can be calculated as follows:

Let ' j_k ' be the total number of interactions across a row captured in DSM by an element in its corresponding cluster ' C_n '.

Let ' I_n ' be the total number of interactions across a row captured by cluster ' C_n ' on a whole.

The probability of interaction occurrence ' PI ' for a given element is obtained by the relative frequency formula:

$$PI = \frac{j_k}{I_n}$$

Calculation of Clustering Entropy

Once both the probability of occurrences for individual clusters and element interactions in each respective cluster are obtained, a weighed sum approach is used as each probability of interaction of an element in a cluster is always dependent upon the occurrence of that particular cluster. This weighed sum approach is project onto a Markov chain that helps to visualize the clusters and their corresponding elements. Figure 16 illustrates the above.

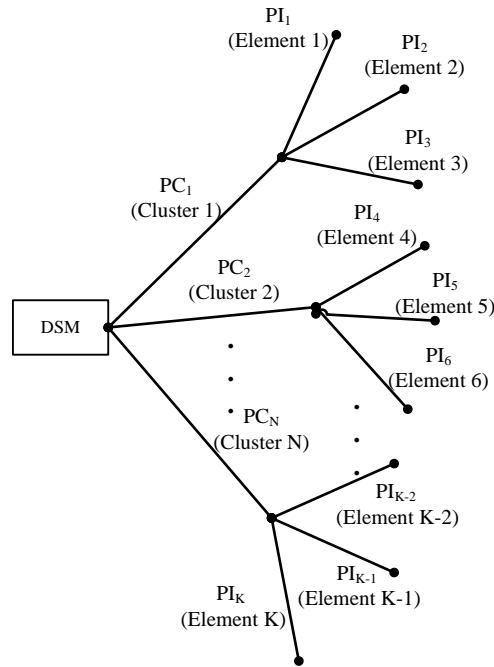


Figure 16: Decomposition of Probabilities

Once such a figure is developed from a given clustered DSM, the Clustering Entropy is calculated using the formulation below.

Assuming E to be the clustering Entropy of a clustered DSM, it is formulated as:

$$E = PC_1 * H(PI_1, PI_2, PI_3) + PC_2 * H(PI_4, PI_5, PI_6) + \dots + PC_N * H(PI_{k-1}, PI_{k-2}, \dots, PI_k)$$

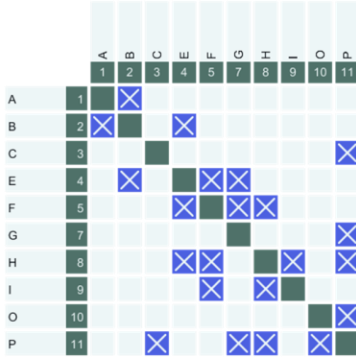
Where, $H(PI_1, PI_2 \dots PI_k) = -\sum_{i=1}^k PI_i \log_2 PI_i$

The resulting output would be a positive real number. It is to be noted that higher the value of E, the more disordered or uncertain the system clustering is.

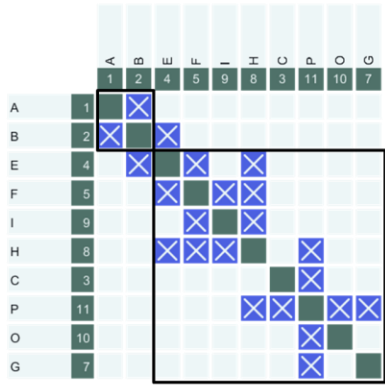
Note: In this case, the interactions outside the clusters are not considered for the analysis.

Example A

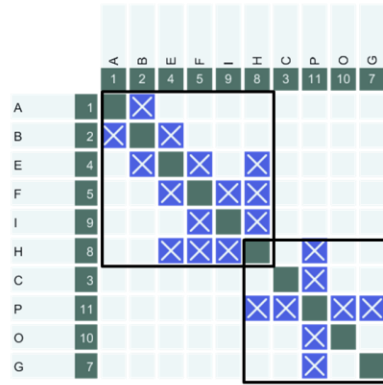
To illustrate the application of the above formulation, a clustering analysis example based on objective function capable of taking input parameters such as cluster size and interactions inside a cluster to minimize the cluster size and the interactions outside the cluster from Eppinger *et al.* is considered. This objective function considered provided flexibility to the authors for observing and characterizing the system behavior. Figure below illustrates an un-clustered DSM with four possible clustering solutions resulting, Two Non-Overlapping clusters, Three Non-Overlapping clusters, Two Overlapping clusters and three Overlapping clusters. See Eppinger *et al. Design Structure Matrix Methods and Applications*, page 26, figure 2.6 for more information [Eppinger and Browning, 2012].



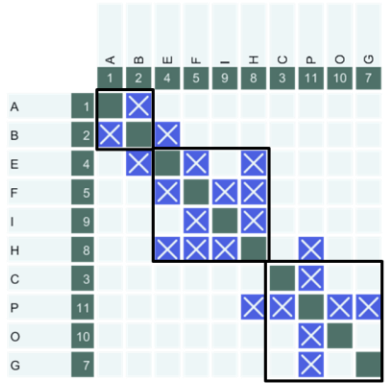
A. Un-clustered DSM



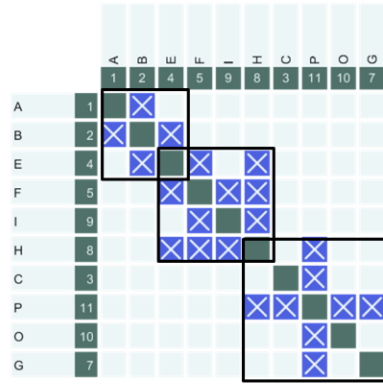
B. Two Non overlapping Clusters



C. Two Overlapping Clusters



D. Three Non-Overlapping Clusters



E. Three Overlapping Clusters

Figure 17: Cluster Analysis (adapted from Eppinger et al., pg.26, fig 2.6)

It can be observed from the figure17 that the overlapping clusters have one element in-common, or, in other words, have membership in two clusters. In the figure, under sub caption 'C', we see that element 'H' had membership in both the clusters the matrix is cluster into. Similarly, in sub

caption ‘E’, we can see that element ‘E’ has membership in clusters 1 & 2 and element ‘H’ has membership in clusters 2& 3. The identification of such linking elements provides an insight on how the system works. Contrary to this, in the case of non-overlapping clusters, the system elements identified for clustering have membership only to the cluster it is assigned to.

Clustering Entropy Calculation

Case 1: 2- Non-overlapping Clusters (sub-caption ‘B’ in figure)

Assuming cluster 1 (2 elements: A, B) is chosen first and then cluster 2 (8 elements: E, F, I, H, C, P, O, G) the following procedure is followed along the guidelines explained previously.

- Out of all the 10 elements used to map a DSM, the probability of cluster 1 occurring would be dependent upon all the possible combination of 2 elements together out of all the 10 elements available:

$$\text{i.e. } PC_1 = \frac{1}{\# \text{ of possible combinations of 2 different elements out of 10 elements available}}$$

We see that the numerator is 1 because, there is only one way where in which A& B elements occur together out of all the possible combinations of 2 elements out of 10.

$$\text{Therefore, } PC_1 = \frac{1}{\binom{n}{k}} = \frac{1}{\frac{n!}{(n-k)!k!}} = \frac{1}{\binom{10}{2}} = \frac{1}{\frac{10!}{(10-2)!2!}} = \frac{1}{45} = 0.0222$$

- According to probability measure of likeliness of an event to occur, the numerical measure of probability always ranges from 0 to 1, where, 0 indicates uncertainty and 1 indicates certainty.
- Also, according to widely accepted probability axioms, A probability measure $P[*]$ is a function that maps events (parts of sample space with one or more

outcomes) in a sample space to real numbers such that [Yates and Goodman, 1999]:

- Axiom 1: For any event A, in a sample space, $P [A] \geq 0$.
- Axiom 2: Considering the whole sample space, $P [S] = 1$.
- Hence, basing upon the above axioms, the probability of occurrence of cluster 2 with elements E, F, I, H, C, P, O, G in it will be

$$PC_2 = (1 - PC_1)$$

$$i.e. PC_2 = (1 - 0.0222) = 0.977$$

Now that the probability of cluster occurrences is known, the next step would be to calculate the probabilities of interactions occurrence of the individual elements.

- Considering element A in cluster 1, we can see that there are 2 interactions with respect to this element which are, the output of B as an input to A and the output of A as an input to B. Similarly in the case of element B, there are 2 associated interactions. Therefore, the total numbers of interactions in cluster 1 are 4.
- The probability of interaction occurrence of individual element pertaining to their assigned clusters can be given as

$$PI_N$$

$$= \frac{\text{\# of interactions of the considered element captured in its assigned cluster}}{\text{total \# of interactions of all the elements captured in the cluster}}$$

- In this case, probability of interaction occurrence with respect to element A is

$$PI_A = \frac{\text{\# of interactions of element A captured in cluster 1}}{\text{total \# of interactions captured by cluster 1}}$$

$$PI_A = \frac{\# \text{ of interactions of element A captured in cluster 1}}{(\# \text{ of interactions by element A}) + (\# \text{ of interactions by element B})}$$

$$PI_A = \frac{2}{2+2} = \frac{1}{2}$$

- Similarly, $PI_B = \frac{2}{2+2} = \frac{1}{2}$
- Now, for interactions in cluster 2,

$$\{PI_E, PI_F, PI_I, PI_H, PI_C, PI_P, PI_O, PI_G\} = \{\frac{4}{36}, \frac{6}{36}, \frac{4}{36}, \frac{8}{36}, \frac{2}{36}, \frac{8}{36}, \frac{2}{36}, \frac{2}{36}\}$$

With both the cluster and individual element interaction probabilities calculate, these values are projected onto a Markov chain.

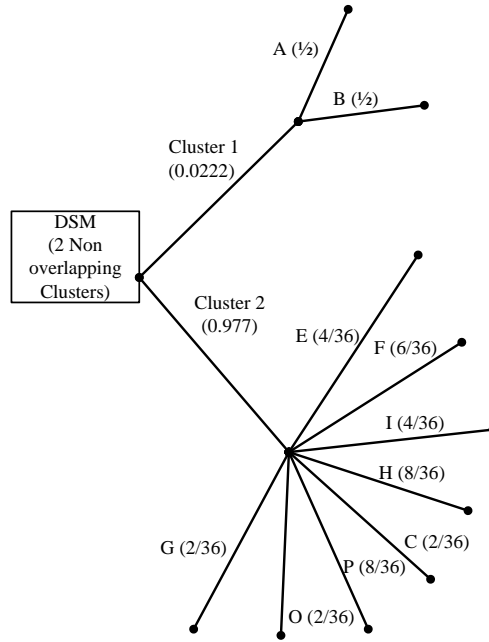


Figure 18: Probability Decomposition for 2 Non-Overlapping clusters in Figure 17

The clustering entropy E is given as

$$E = PC_1 * H(PI_A, PI_B) + PC_2 * H(PI_E, PI_F, PI_I, PI_H, PI_C, PI_P, PI_O, PI_G)$$

$$= 0.022 * H\left(\frac{1}{2}, \frac{1}{2}\right) + 0.977 * H\left(\frac{4}{36}, \frac{6}{36}, \frac{4}{36}, \frac{8}{36}, \frac{2}{36}, \frac{8}{36}, \frac{2}{36}, \frac{2}{36}\right)$$

$$E = 2.754$$

Case 2: 2- Overlapping Clusters (sub-caption 'C' in figure)

To calculate the entropy here, the procedure is same as the followed in the case of 2 non-overlapping clusters, except for, the element H which has membership in two clusters; the interactions are considered once for each cluster when calculating the interaction occurrence probability. Figure below shows a Markov chain constructed to calculate the entropy. It is seen that element H is represented twice as H1 for Cluster 1 and H2 from Cluster 2. The interactions with respect to H being captured in the analysis are with respect to the number of interaction element h contributes to the two different clusters individually.

Assuming cluster 1 is chosen first and the cluster 2 the entropy is calculated as

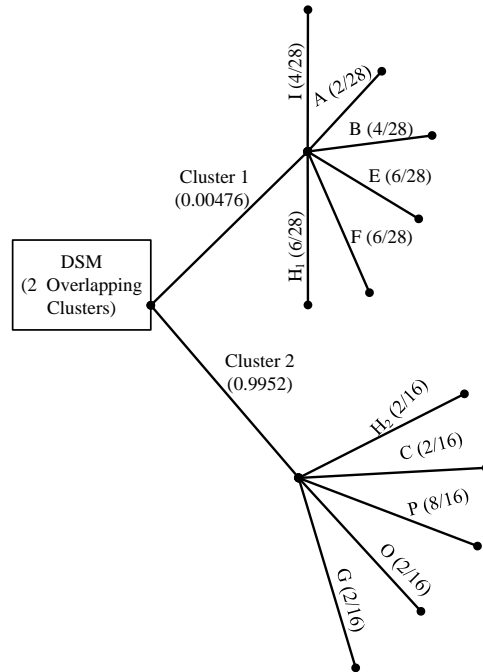


Figure 19: Probability Decomposition for 2 Overlapping clusters in Figure 17

$$\begin{aligned}
E &= PC_1 * H(PI_A, PI_B, PI_E, PI_F, PI_I, PI_{H1}) + PC_2 * H(PI_{H2}, PI_C, PI_P, PI_O, PI_G) \\
&= 0.00476 * H\left(\frac{2}{28}, \frac{4}{28}, \frac{6}{28}, \frac{6}{28}, \frac{4}{28}, \frac{6}{28}\right) + 0.977 * H\left(\frac{2}{16}, \frac{2}{16}, \frac{8}{16}, \frac{2}{16}, \frac{2}{16}\right) \\
E &= 2.00230
\end{aligned}$$

The Entropy calculations for the considered example of two Non-Overlapping clusters, three Non-Overlapping clusters, two Overlapping clusters and three Overlapping clusters are all summarized in the tables below.

Table 5: Cluster entropy for Overlapping clusters in figure 8

<i>Overlapping Clusters</i>	
No: of Clusters	Entropy
2	2.0023
3	1.9952

Table 6: Cluster entropy for Non-Overlapping clusters in figure 8

<i>Non-Overlapping Clusters</i>	
No: of Clusters	Entropy
2	2.754
3	1.777

Example B

This example illustrates the interactions identified in the 3D printer based on its high-level component decomposition. Please refer to the previous explanation on how a DSM for 3D printer is captured and reflected on to LOOME software to identify its clusters. Figure 20 illustrates the DSM clustering analysis using LOOME software, performed on U-Print® SE personal 3D printer.

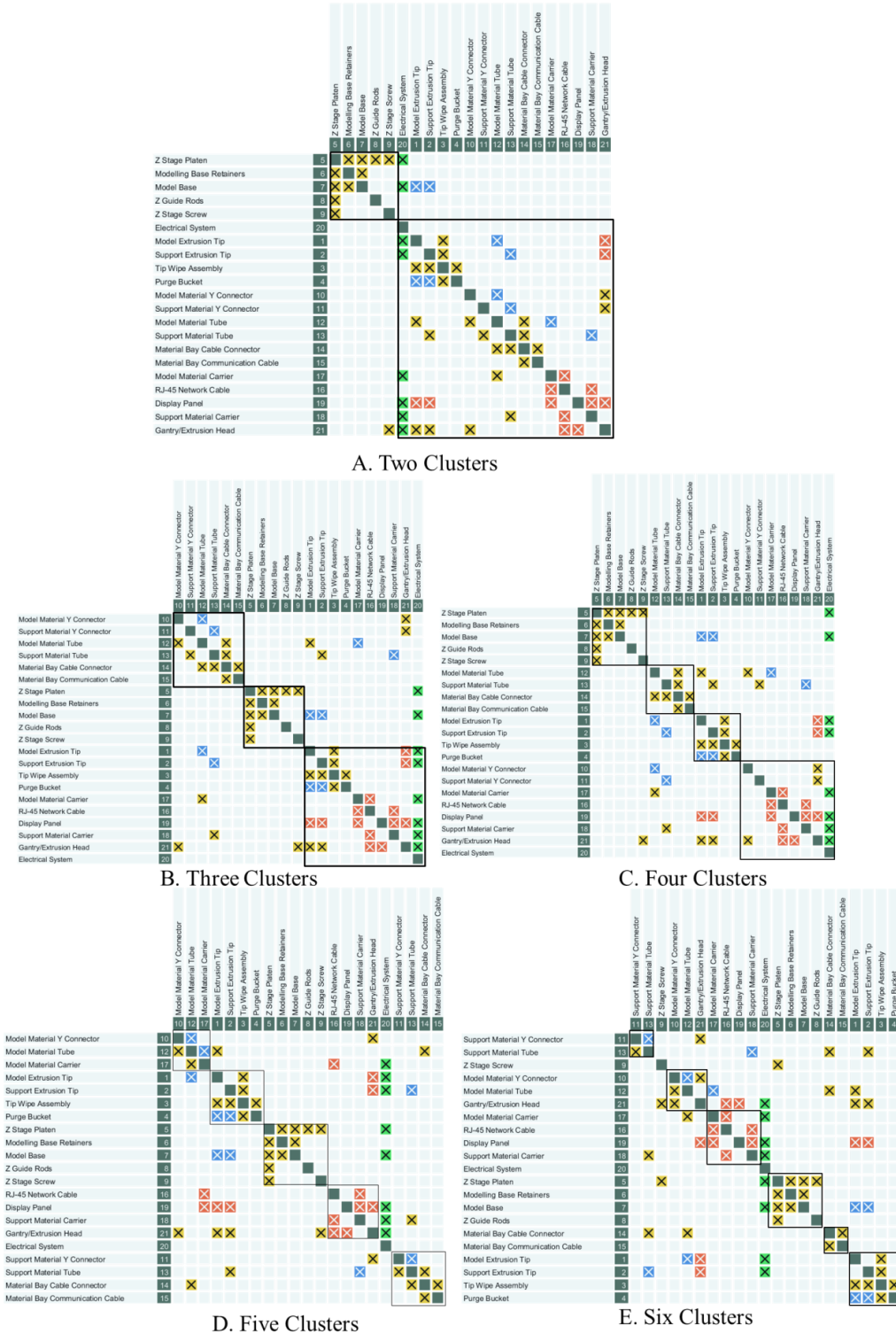


Figure 20: U-Print® SE personal 3D printer clustering analysis based on its un-clustered DSM

To identify the effect of Entropy calculated based on its system clusters for this example, various clusters (2, 3, 4, 5, and 6) were identified as shown in the figure based on the software's embedded clustering algorithm. Around 15 iterations were run to maintain the consistency of results while identifying the clusters. Clustering entropy results for these clusters are summarized individually in table 7.

Table 7: Cluster entropy for U-Print® SE personal 3D printer clusters illustrated in figure 18

<i>U-Print® SE personal 3D printer Clusters</i>	
No: of Clusters	Entropy
2	4.375
3	3.054
4	2.628
5	1.914
6	1.751

The results obtained & illustrated in the table show that the value of clustering entropy decreases with the increase in the number of clusters the DSM is divided to.

The most important aspect of any complex system, in this case considering U-Print® SE personal 3D printer as an example, is the integration strategy followed. Integration of the system may be bottom-up, piece wise or by sub-system functionality reflecting the system as a whole. Integration strategies followed incorporate testing of individual components that would better help to identify defects, if any, or, possibly following structured integration along each phase toward the complete system.

System clusters identification will act as boundaries that help to understand the functionality, system elements behavior, and sub systems, along with understanding their logical and structural relations. Understanding and identifying these boundaries based on the obtained clusters from

the static DSM developed, helped us to differentiate the level of interactions among the system elements and their inter- dependency.

Clustering Entropy, based on the two examples provided, portrays a specific trend in its behavior. It gradually decreases in value when there is an increase in the number of clusters in the DSM. Comparatively speaking, there is more uncertainty present when a system is divided into fewer clusters than when it has more clusters. This conclusion can be verified based on a simple understanding that:

- When a system with many interdependent and interacting components is divided into only two clusters, a faulty element or a faulty interaction in either of the clusters will propagate quickly throughout the whole system. To mitigate or to rectify such an error requires more effort and time to zero in on the error, thereby reflecting the uncertainty in the system.
- On the contrary, if a system has more clusters, an error or a faulty interaction in any one of the clusters, though it will affect how the system works, will take less effort and time to identify and rectify the error, comparatively, resulting in rapid decrease of the associated system uncertainty.

In addition, the clustering entropy for each identified cluster will help to identify the individual clusters having more or less uncertainty compared with the other clusters in the system. Clustering entropy, when applied to organizational architectures with relationships tracing throughout the whole system hierarchy, (such as in Departments, Individuals and Business units), and to team and people based DSM's, helps to identify and address the key weakness of the system based on cluster analysis, thereby enabling continual performance improvement.

Chapter 3 – Entropy Based Complexity Measure For Software Based Control

Flow Graphs

Software Engineering

Exploring the literature available for understanding and capturing the principles behind software engineering, one often comes across a few well-defined and widely followed definitions. One of the earliest definitions for software engineering was laid down in the first NATO conference in 1968 [Hans van Vilet, 1993]:

“Software engineering is the establishment and use of sound engineering principles in order to obtain economically, software that is reliable and works efficiently on real machines.”

The definition given by Mills in his paper titled *“The management of software engineering, Part 1: Principles of software engineering,”* published in the IBM Systems Journal in 1980, emphasizes the systematic design and development approach of software products [Mills, 1980].

“Software Engineering may be defined as the systematic design and development of software products and the management of the software process. Software engineering has as one of its primary objectives the production of programs that meet specifications, and are demonstrably accurate, produced on time and within budget.”

The two main attributes to be focused on for engineering software, according to the first definition, are *Reliable* and *Efficiency*. *Reliable* means that the software will consistently perform based on recognized industry standards (few industry standards geared towards achieving software reliability are: IEEE Std-982.1-1988, IEEE Std-982.2-1988, IEEE Std-1413-1998, ISO/IEC 15504:1998); *Reliable* also has implications in cyber security and trust. *Efficient* means that the software will achieve maximum productivity with minimum effort. With many

such attributes involved in the development of software in industry, a systematic approach is used for software development.

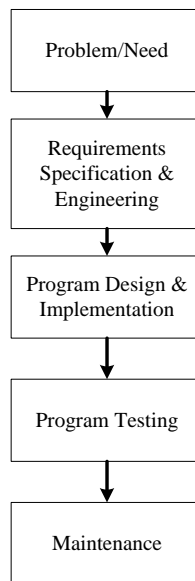


Figure21: Software Engineering process & its associated phases

As Figure 21 illustrates, engineering any software incorporates many phases. Phase 1: Problem/Need identification, and Phase 2: Requirements Specification & Engineering, involve problem identification, functions identification to develop the software, gathering the required documentation, establishing performance requirements and problem decomposition into flexible components to be encoded in a specific programming language. Phase3: Program Design and Implementation that involves design and implementation, is emphasized here. The Program Design and Implementation phase concentrates on developing individual program modules based on pre-defined specifications. This phase helps to develop an executable modular code rather than a single complex executable code that often tends to be too large. The goal of a programmer in Phase 3 will be geared towards developing a well-documented, reliable, efficient, flexible, and

understandable code. Phase 4 and Phase 5 involve verification and validation of the developed code, along with regular maintenance and improvement if required.

Software Quality and Complexity Measures

Measurement is a process of ascertaining the degree of a given attribute, or, in other words, it is a process of assigning a quantitative value to a functional unit, e.g. Number of variables, Lines of code, etc.

Software quality measures are defined as metrics primarily measured against the degree to which user requirements are met. These requirements could be, for example: Correctness, reliability, efficiency, integrity, usability, maintainability, etc. Table 8 illustrates the quality factors of software that can be measured either directly or indirectly [Hans van Vilet, 1993]. For more details on measuring these attributes, refer to McCall *et al.* [1977].

Table 8: Software Quality Metrics [McCall et al., 1977]

<i>Quality Attributes</i>	<i>Definition</i>
Correctness	Measure where the program satisfies the requirement to an extent
Reliability	Measuring the performance of the program functions to its expected function and precision
Efficiency	Computing and Code resource measure for a program to perform a function
Integrity	Measure to which the access to software is controlled
Usability	Effort measure required to capture and reflect the functionality of the program
Maintainability	Effort required to identify and fix program errors if any
Testability	Effort required to test a program
Flexibility	Measure of flexibility of the program for it to be adaptable of multiple platforms
Reusability	Measure where either a part of the program or the whole program can be reused for different applications
Interoperability	Interface capability measure

Software code complexity measures are mainly used or adapted in the design and implementation phase of a code. They are used to measure the individual inherent complexity of code modules, individual components (a software component is an element composed based on a set of pre-defined standards that conform to a specific behavior) and procedures. Modules, procedures and components of a code, irrespective of the level at which they are developed, are inter-dependent. The structural and information architecture of code has a significant impact on both complexity measures and quality measures.

Complexity in software codes can be defined as the attribute associated to a code that effects the effort required to either develop, change or debug a piece of software. Many different methods have been suggested throughout the literature in this field for the quantitative characterization of the complexity inherent in software. These metrics, when captured quantitatively, work as anchors for software design development and re-engineering efforts. The complexity metrics can be broadly characterized into Information based complexity metrics, and Structural based complexity metrics.

Information based complexity metrics generally relate to the information content used while developing a piece of software, and include complexity measures calculated based on size of a software, lines of code, number of operators, number of operator occurrences, number of string occurrences, vocabulary size, information content, information shared, number of variables, and so on.

Metrics calculated based on either flow graph notation, information flow pattern, number of connections among the modules, module hierarchies, fan in and fan out of modules, and modules or chunks with inter-modular dependencies, fall under Structural based complexity metrics.

Table 9 lists contributions in the development of software code complexity metrics.

Table 9: Contributions to Software Complexity Metrics

Category	Contributing Authors	Complexity metric
Information based	G.M. Weinberg	Complexity measure based on number of lines of code [Hans van Villet,1993]
	Maurice Howard Halstead	Based on number of operators and operands [Hans van Villet,1993],[Hamer <i>et al.</i> , 1982]
	Scott W. Woodfield	Based on Conceptually unique operands [Scott N. Woodfield, 1979]
	Eli Berlinger	Information based Complexity Measure [Eli Berlinger, 1980]
	Maurice Howard Halstead	Program Effort and Difficulty measure based on program vocabulary, number of distinct operators, total number of operators, total number of operands and volume.[Elaine J. Weyuker, 1988],[Hamer <i>et al.</i> , 1982]
	Fitzsimmons & Love	Number of delivered errors based on Halstead's Effort metric[Fitzsimmons and love, 1978]
Size & Structure based	El Oviedo	Complexity measure based on control and data flow using number of available definitions of variables in blocks of program body.[Elaine J. Weyuker, 1988],[Enrique I. Oviedo, 1980]
	Thomas J. McCabe	Cyclomatic Complexity Metric. [Watson <i>et al.</i> , 1996]
	Sallie Henry & Dennis Kafura	Metrics based on Global, local and Indirect flow relations.[Hans van Villet,1993], [Henry and Kafura, 1981]
	Shepperds	Metric based on Fan in and Fan out measures. [Hans van Villet,1993]

Concept of Entropy on Software Metrics

Entropy in thermodynamics represents the inherent disorder in a system over a period of time as the system heads towards thermodynamic equilibrium. In information theory, according to Shannon, Entropy helps to quantify the information [Shannon and Weaver, 1949]. Quantifying information implies analyzing the information present and measuring its associated uncertainty.

Higher values of entropy signify lesser order in a system and lower the values of entropy signify a more ordered system. Entropy has found broad application in many fields and can also be applied in software engineering for quantifying the uncertainty associated with a software code. Entropy can also be used to develop improved complexity measures.

Entropy, H , according to statistical mechanics:

H of a system is defined as:

$$H = -K \sum P_i \log P_i$$

Where, P_i is the probability of a particular state and K is Boltzmann's constant.

Shannon Entropy H is given as:

$$H = \sum P_i \log_2 P_i$$

Where P_i is the probability of a symbol showing up in a given stream of symbols and the use of the logarithm base two corresponds to expressing information entropy in terms of bits.

There are a number of studies where entropy is used as basic foundation for software complexity measures. Here we introduce a few entropy based complexity measures. Several measures have been proposed and defined which are sensitive to the probability values calculated based on the frequency of usage of symbols, set of software inputs, set of outputs, set of links of nodes, frequency of string occurrence in a code, frequency of names occurring in a code, frequency of operator occurrence, number of attributes, reuse ratio, frequently occurring operators, number of leaf nodes and also a few object oriented design metrics [Eli Berlinger, 1980], [Woo-Sung Jung *et al.*, 2011], [Selvarani *et al.*, 2010], [Jose Luis Roca, 1996], [Chaturvedi *et al.*, 2014], [Greg Snider, 2001], [Warren Harrison, 1992], [Bansiya *et al.*, 1999].

Greg Snider [2001] provided a complexity metric using structural graphs. This model proposed that a structure graph consists of two types of nodes and two different edges. The two types of

nodes, leaf nodes and interior nodes, along with two different edges, structural and dependency edges are used to calculate the complexity. Leaf nodes relate to global symbols, function names, global variables, global structure names or a set of global symbols in a piece of code, while an Interior node is an aggregation of leaf nodes. For the edges, a Structural edge connects interior and leaf nodes, whereas Dependency edges connect only leaf nodes. Here, the entropy based complexity metric for measuring the entropy of large software systems is based on number of leaf nodes, number of dependency edges, and the distance between two leaf nodes (minimum number of interior nodes traversed) [Greg Snider, 2001]. Refer to Greg Snider [2001] for more information.

Warren Harrison provided an entropy-based measure of software complexity on the basis of information theory. This metric is developed based on the hypothesis that a program with high information content on average should on whole be less complex compared to a program with an average of less information content. Harrison calls the complexity metric an Average Information Content Classification (AICC) measure, which is dependent upon the total number of operators used in the program and the frequency by which a considered operator appears in the code [Warren Harrison, 1992]. Refer to Warren Harrison [1992] for more information on this metric.

According to Bansiya *et al.*, an entropy based complexity metric for object oriented designs can be applied in the early stages of development to ensure that a developer analyzes and reiterates the internal characteristics that lead to a quality oriented design. The entropy measure developed is solely a measure of class complexity to measure the information content, which is a function of number of strings in a class and on how frequently a string repeats within class definitions, irrespective of the language being used. The Class Definition Entropy (CDE) is developed on the

basis of Shannon entropy, where CDE is characterized by the probabilities of most frequently occurring strings [Bansiya *et al.*, 1999]. Refer to Bansiya *et al.* [1999] for more information on this metric.

Eli Berlinger provided an information theory based complexity measure based on entropy theory. The defined complexity measure is sensitive to the frequency of occurrence of all the tokens in a program. Tokens such as operators and operands here refer to elements of the programming language being used. According to Berlinger, there are several possible interpretations of this measure, either an information point of view where it represents the total information contained in the code or an ideal coding scheme representing the total length required to develop the program. To add, Berlinger suggests that irrespective of the interpretation used, this measure is sensitive to the frequency of the symbols' occurrence and the proportion of the number of times the symbol occurs in the past [Eli Berlinger, 1980]. Refer to Eli Berlinger [1980] for a detailed explanation of this metric.

Although there are a lot of contributions and studies observed in this field of software complexity, it is observed that not many authors consider structural and logical flows of input and output variables among the developed software modules for calculating software complexity, which either directly or indirectly relates to software quality. A logical flow here is defined as a representation of decision-making processes coded into software modules and a structural flow to be a representation of the interaction patterns among the statements in a software code. Though the topological interactions and logical characteristics have been previously considered individually, characterizing complexity based on structural and logical flows along with time is not observed. Therefore we develop a complexity measure for software which considers logical

decision making processes, software statement interactions patterns and time, utilized together to create an improved software complexity measure based on the concept of entropy.

Proposed Metric Definition

The proposed software complexity metric will be developed based upon Shannon's entropy form. Shannon's entropy, based in terms of symbols output from a source, utilizes the number and type of symbols historically output, and the frequency of occurrence of the symbols, to represent the average information content of the message source [Shannon and Weaver, 1949].

The modules of software code, when similarly contemplated in terms of symbols, will have input and output flows that provide for information transfer from one module to another. Assuming that the modules are fully functional without any uncertainties associated with them, the entropy metric considers the data flow relationships of a module. A module represents a decision control structure, loop control structure, case control structure, subroutine, or a function.

The entropy associated with a module is dependent upon the output data flows from a module and input data flows to a module and the time of execution at the modules. Depending upon the behavior of the module considered, it has corresponding paths for input and output flows. This behavior of a module, embedded in software, when mapped graphically, forms the foundation of the metric developed here.

We now define a new entropy based complexity metric. Let a given piece of code be characterized using a control flow graph wherein the nodes represent various modules associated with the code and the edges represent the input and output flows associated with them. Each edge originating at a node has a time factor associated with it based on the time taken to successfully complete the task it is coded for. The more the number of the inputs and output

flows associated to a module, depending upon its characterization, the more the associated uncertainty. The complexity measure is thus defined as:

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji} \quad (1)$$

Where

n is the number of nodes characterizing the software

k is the number of outputs associated to a node. Number of outputs here are different outgoing edges representing all possible distinct outputs leading to different nodes.

l_j is the likelihood of occurrence associated node to j based on number of arcs incoming to the node

P_{ji} is probability distribution of the output i associated to node j

When a software code when transformed to a control flow graph, the value of n is obtained based on the number of nodes in the graph. It is assumed here that all the edges in CFG have a travel time of one unit each associated with them. Likelihood of occurrence l_j , associated to a node depends upon the number of inputs (number of arcs incoming to a node) of node j . This is based upon the assumption that more the number of inputs of a node, it is more likely to occur in a graph which may be due to the fact that it either has more feedback loops, controls from predecessor nodes, loops entering a node, loops ending at a node, or multiple possible executions. Probability distribution P_{ji} , of output i associated to node j , is based on its output according to the Principle of Indifference. According to the principle of indifference, suppose there are k indistinguishable possible outputs coming out of a node represented on a control flow graph, then each outcome will be assigned a probability $1/k$. For a given node, each outgoing

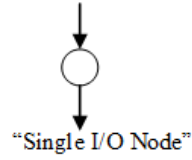
edge represents an outgoing flow of control after some part of node execution. The probabilities are based on the outgoing flows.

Proposed Metric Characterization

The proposed entropy measure is additive; that is, the amount of total entropy in a piece of code characterized by a control flow graph is the sum of the individual entropies of all the associated nodes. This metric can be characterized in different ways to support its application in software code systems. As discussed previously, nodes are associated with their individual input and output flows, which influence their software behavior. Since the nodes of a control flow graph at the structural level require representing decision logic and flows, the following primitive formulations can be established, as shown in Figure 22. The value of $H(1)$ in Figure 22 implies that the particular node with the value of $H(1)$ has a single outflow that occurs with a probability of 1.

According to the principle of Shannon entropy, considering an example of a coin toss, there is an equal chance of heads or tails and the outcome of this experiment has an entropy or information content of one bit. Similarly, if there are two outputs from a node, the entropy associated, based on the principle of indifference, will be equal to one.

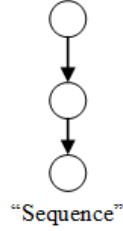
$$\text{i.e. } E = H\left(\frac{1}{2}, \frac{1}{2}\right) = \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{2} \log_2 \frac{1}{2} = 1$$



$$E = H(1)$$

$$H(1) = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

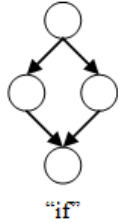
$$= (1) \times (1) \log_2 (1) = 0$$



$$E = H(1) + H(1)$$

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

$$H(1) + H(1) = (1) \times (1) \log_2 (1) + (1) \times (1) \log_2 (1) = 0$$



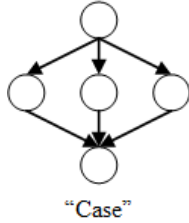
$$E = H(1/2, 1/2) + H(1) + H(1)$$

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

$$H(1/2, 1/2) + H(1) + H(1)$$

$$= (1) \times [(1/2) \log_2 (1/2) + (1/2) \log_2 (1/2)] + 0 + 0$$

$$= 0.5 + 0.5 = 1$$

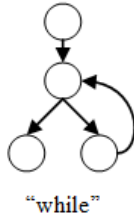


$$E = H(1/3, 1/3, 1/3) + H(1) + H(1) + H(1)$$

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

$$H(1/3, 1/3, 1/3) + H(1) + H(1) + H(1)$$

$$= (1) [(1/3) \log_2 (1/3) + (1/3) \log_2 (1/3) + (1/3) \log_2 (1/3)] + 0 + 0 + 0 = 1.58$$

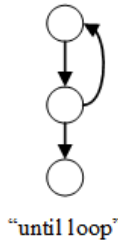


$$E = H(1) + 2 \times H(1/2, 1/2) + H(1)$$

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

$$H(1) + 2 \times H(1/2, 1/2) + H(1)$$

$$= 0 + (2) \times [(1/2) \log_2 (1/2) + (1/2) \log_2 (1/2)] + 0 = 2$$



$$E = H(1) + H(1/2, 1/2)$$

$$H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$$

$$H(1/2, 1/2) + H(1)$$

$$= (1) \times [(1/2) \log_2 (1/2) + (1/2) \log_2 (1/2)] + 0 = 1$$

Figure 22: Basic primitive definitions (Note: All the starting nodes are assumed to have inflow of 1)

Figure 23 shows a sample Control Flow Graph (CFG) for the application of this complexity metric. The metric is computed for all the nodes of the graph based on its input and output flows.

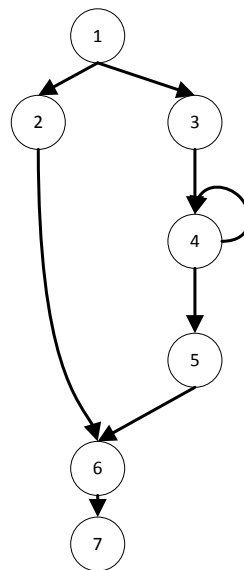


Figure 23: CFG for Metric Application

The computation of the metric is based on the Entropy value calculated at each and every node identified using a Control Flow Graph (CFG). The Entropy based software complexity is obtained by the summation of the entropy values at each node of CFG. As mentioned previously, the metric is solely based on the output distribution and likelihood of occurrence of the nodes. All the edges associated to the CFG are assumed to be of one time unit each. Table 10 illustrates the metric calculation for the considered CFG. The CFG Complexity metric equals to the sum of the values in column 4 of the Table 10 $E = H_{total} = 3$. The complexity metric computes a number to represent the complexity. The higher the value of the entropy, the more complex is the considered piece of code, and vice versa. Also, information can be drawn at each node to identify the possible complexity associated with that particular node by measuring the entropy value based on the number of outputs and the likelihood of occurrence of each output.

Table 10: Entropy based metric value for CFG

<i>Node</i>	<i>Likelihood of occurrence (Incoming edges)</i>	<i>No: of outputs (data outputs)</i>	<i>Entropy</i> $H = - \sum_{j=1}^n \sum_{i=1}^k l_j P_{ji} \log_2 P_{ji}$
1	1	2	$1 * H(\frac{1}{2}, \frac{1}{2}) = 1$
2	1	1	$1 * H(1) = 0$
3	1	1	$1 * H(1) = 0$
4	2	2	$2 * H(\frac{1}{2}, \frac{1}{2}) = 2$
5	1	1	$1 * H(1) = 0$
6	1	1	$1 * H(1) = 0$
7	1	1	$1 * H(1) = 0$
SUM			H total = 3

Therefore, based on the illustrated computations of the metric values, node 4 ($H = 2$) is more complex than node 1 ($H = 1$), and node 1 ($H = 1$) is complex comparative to all the other associated nodes (with $H = 0$) in the CFG in Figure 3. This metric is not suggested to be used on known modules (nodes) with same characteristics used in different code, as the likelihood of occurrence at each module is dependent upon the type of code in which it is being used.

Before validating the metric, Figure 24 illustrates how the proposed entropy based complexity metric ties into the core concepts and fundamentals of software complexity theory. Based on the proposed metric and its sensitivity for the number of outputs from each node and the likelihood characterized by the inputs at each node, it falls under the umbrella of software complexity attributes. Referring back to different categories of software complexity metrics, Information based complexity measures are based on vocabulary, number of distinct operators, total number of operators, total number of operands, and lines of code, while Size and Structure based complexity metrics are based on control flow, global and local input and output flows, fan-in and fan-out are dependent upon input and output factors of the associated nodes. Also, at a system

structural level, complexity is dependent upon the inter-modular interactions characterized by nodes, edges and their relations. Thus, it can be seen that the proposed complexity metric which is highly dependent upon the control flow structure of a given piece of code, the number of nodes associated, the input and output at each node, and their relations, can be undoubtedly traced back to the fundamentals of software complexity measures.

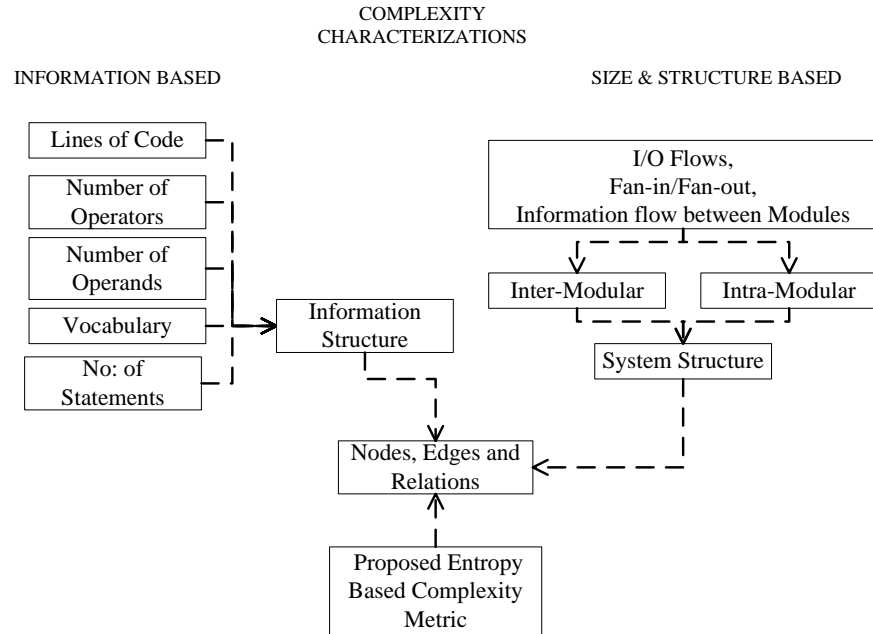


Figure 24: Architectural relation of proposed metric to software complexity characterization (dashed arrow indicates dependency relationship)

Proposed Metric Validation

In order to verify and validate the proposed metric and its functionality of determining a complexity measure based CFG of a given piece of code, it is here correlated with a well-known and frequently cited, Thomas J. McCabe's Cyclomatic Complexity measure. This validation is based on calculating complexity of 12 different control flows using the proposed metric and using McCabe's complexity measure. The Cyclomatic Complexity measure is based on the number of edges, number of connected components, and the number of vertices in a CFG. The

12 different control flow graphs shown in the Appendix A are adapted from McCabe's paper which establishes a Cyclomatic Complexity measure for a given program based on its characterization as a control flow graphs [Thomas J. McCabe, 1976]. McCabe's Cyclomatic Complexity number along with the complexity values obtained using the proposed metric are tabulated in Table 11.

Table 11: Complexity Metric Values of CFG in Appendix A

<i>Control Flow Graph</i>	<i>Number of Nodes</i>	<i>McCabe's Cyclomatic Complexity Measure</i>	<i>Measure using proposed Complexity Metric</i>
1	3	0	0.00
2	7	3	3.00
3	10	5	6.00
4	12	6	7.00
5	12	8	7.58
6	13	8	8.16
7	19	9	12.16
8	20	10	13.00
9	23	10	11.58
10	25	11	16.58
11	18	10	16.00
12	36	19	22.16

Spearman's rank correlation method was used to measure the correlation between the two different variables of complexity measures obtained in Table 11. According to Spearman's rank correlation method, for given two sets of variables s_1 and s_2 , the strength of the correlation is a monotonic relationship. The variables are first ranked in such an order that, the variable with an higher value will be assigned the value of n , and the variable with the least value will be assigned rank 1, where n is the number of variables considered. The formula used for Spearman's correlation coefficient is [Bansiya *et al.*, 1999],

$$r_s = 1 - \frac{6 \sum d^2}{n(n^2 - 1)} \quad (2)$$

Where

r_s is Spearman's correlation value,

n is the number of variables considered for analysis, and

$\sum d^2$ is squared summation of difference of the variable ranks

A correlation value in the range of 0.00 to 0.19 represents a “very weak correlation,” a value in the range of 0.20 to 0.39 represents a “weak correlation,” a value in range of 0.40 to 0.59 signifies a “moderate correlation,” a value between 0.60 to 0.79 signifies a “strong correlation,” and a value between 0.80 and 1.0 represents a “very strong correlation” [21].

Table 12 shows the relative ranking of the complexity metrics.

Table 12: Relative Rankings of Complexity Metric Measures in Table 4

<i>McCabe's Complexity Measure (x_i)</i>	<i>Complexity Measure using proposed metric (y_i)</i>	<i>Ranks of (x_i)</i>	<i>Ranks of (y_i)</i>
0	0.00	1.0	1.0
3	3.00	2.0	2.0
5	6.00	3.0	3.0
6	7.00	4.0	4.0
8	7.58	5.5	5.0
8	8.16	5.5	6.0
9	12.16	7.0	8.0
10	13.00	9.0	9.0
10	11.58	9.0	7.0
11	16.58	11.0	11.0
10	16.00	9.0	10.0
19	22.16	12.0	12.0

A value of $r_s = 0.9771$ is obtained using Spearman's correlation coefficient formulation for a sample size of 12 variables implying that the proposed Complexity metric and McCabe's complexity metric are very strongly correlated.

To further validate the metric, sets of seven different Matlab codes are converted to Control Flow Graphs. These codes are adapted and randomly chosen from freely available online database provided by Massachusetts Institute of Technology [Matlab Teaching Codes, MIT]. These Matlab codes are programmed to perform basic linear algebraic computations. Please see Appendix B for the Matlab codes and their corresponding Control Flow Graphs.

Validation here is based on the same guidelines followed previously, where, both the Cyclomatic Complexity and proposed entropy based complexity measures are each calculated for the control flow graphs. Spearman's rank correlation method is used to measure correlation between the two complexity measures. Table 13 illustrates the complexity metric measures calculated based on the control flow graphs developed from the Matlab codes and the relative rankings of the obtained measures are tabulated in Table 14.

Table 13: Complexity Metric Values of CFG in Appendix B

<i>Control Graph</i>	<i>Flow</i>	<i>Number of Nodes</i>	<i>M McCabe's Cyclomatic Complexity Measure</i>	<i>Measure using proposed Complexity Metric</i>
1		15	5	5
2		26	11	19
3		28	13	23
4		11	4	7
5		6	2	2
6		10	2	4
7		9	3	2

Table 14: Relative Rankings of Complexity Metric Measures in Table 13

<i>McCabe's Complexity Measure (x_i)</i>	<i>Complexity Measure using proposed metric (y_i)</i>	<i>Ranks of (x_i)</i>	<i>Ranks of (y_i)</i>
5	5	5	4
11	19	6	6
13	23	7	7
4	7	4	5
2	2	1.5	1.5
2	4	1.5	3
3	2	3	1.5

A value of $r_s = 0.8818$ is obtained using Spearman's correlation coefficient [Bansiya *et al.*, 1999] implying that the proposed Complexity metric and McCabe's complexity metric are further strongly correlated. Based on the set of considered graphs, it can be seen that the proposed metric can further be very strongly correlated with well-established McCabe's Cyclomatic Complexity measure.

This metric is now modified to include the time of execution at each and every module for incorporating execution time into the complexity analysis. Execution time of a program can be defined as the time taken by the program to process its inputs [reference]. A software code module's execution time depends upon several factors such as the instruction set used, type of compiler, processor speed and several other similar factors [Victor S. Adamchik, 2009]. This implies that the time of execution associated to a module depends upon its implementation. Complexity of a program based on Time is defined as a measurement of how fast the time taken by the program grows with an increase in the input size, implying that for a given input vector $n = \{n1, n2, n3...\}$, the execution time t taken will be proportional to n , which can be represented as $t \propto n$.

It is here assumed that a software code when represented by a control flow graph has exponentially many paths for its execution. The execution time of a module (represented as a node in control flow graph) remains same and the total execution time will be based on the count of the individual node occurrence. Therefore the run time of the program will be equal to the summation of the total execution times at each node.

The improved complexity measure, which also incorporates individual node execution times, is now represented as shown below based on the following assumptions

Let 'n' be the number of nodes in the CFG

Let 'm' be the number of outputs originating from a node 'n'

Let 'r' be the number of inputs converging to a node 'n'

$$H = - \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^r T(i) \text{cnt}(i_k) P_{ij} \log_2 P_{ij} \quad (3)$$

Where,

$T(i) = \{t(i_1), t(i_2), t(i_3) \dots t(i_k)\}$, a vector of execution times for each node associated to CFG.

$\text{cnt}(i_k)$ Is the count of the number of inputs to a node n .

P_{ij} Is the probability distribution of output j associated to node n .

From the perspective of a programmer, the execution time of a node while either active (while executing the functions) or inactive (while waiting for an input) depends upon the computational algorithms and processes of the node, thereby units ranging in the order of nanoseconds, milliseconds or seconds. To overcome the effect of the units in complexity analysis we suggest normalizing these values on a scale of 0 to 1. Normalizing the execution times though maps them to a range of values among 0 to 1; the effect of execution time on complexity analysis still

remains the same. For convenience, the formulated metric shown in equation 3 can be represented in the following terms

$$H = - \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^r C(i_k) * (P_{ij} \log_2 P_{ij}) \quad (4)$$

Where,

$$C(i_k) = \left(\frac{t(i_k) - \min(T(i))}{\max(T(i)) - \min(T(i))} \right) * cnt(i_k)$$

The data and procedure used for validation of the initially suggested metric in equation 1 holds credible for the improved version of the metric illustrated in equation 4 based upon the assumption of a unit execution time taken by each node on the fact of unavailable data on specific node execution times.

Proposed Metric Evaluation

In order to evaluate the proposed entropy based metric, we use a set of eight axioms formulated and proposed by Elaine J. Weyuker in the year 1988. Weyuker suggests that these axioms are a set of conclusive evaluation measures to be satisfied by any syntactic complexity measure. The four well known software complexity measures, Cyclomatic Complexity Number, Halstead's Programming Effort, Statement Count and Oviedo's Data flow Complexity were evaluated by Weyuker against the proposed axioms and it was found that none of the measures satisfy all the properties [Weyuker, 1988].

Here, these measures are used to intuitively understand the properties of the proposed metric and to identify possible scenarios where it can be applied, along with identifying the weaknesses, which helps to decide whether the proposed metric is useful in a given scenario or not. This

section introduces each axiom that Weyuker [1988] deemed necessary for any complexity measure and thereby shows whether the proposed metric satisfies the properties.

Notations:

- $\exists P$ represents a program body P
- $\exists Q$ represents program body Q
- $\exists R$ represents program body R
- $|P|$ represents the complexity measure of P
- $|Q|$ represents the complexity measure of Q
- $P \equiv Q$ represents that P and Q have same functionality
- $|P; Q|$ represents the complexity measure obtained by concatenation of P and Q
- $|P; R|$ represents the complexity measure obtained by concatenation of P and R
- $|R; P|$ represents the complexity measure obtained by concatenation of R and P
- $|R; Q|$ represents the complexity measure obtained by concatenation of R and Q
- $|Q; R|$ represents the complexity measure obtained by concatenation of Q and R

Property 1: $(\exists P) (\exists Q) (|P| \neq |Q|)$ i.e. there exists a program body P and a program body Q where, a given complexity measures should not rank them as equally complex.

This property requires the metric to uniquely measure the complexity of each program, thereby ensuring that not all the programs are calculated to be equally complex. From Equation 4 it can be seen that the proposed metric depends on: the type of control flows in the program, the number of inputs to each node, the output probability distribution of the nodes and the execution times which are unique for a given program, thereby satisfying this property.

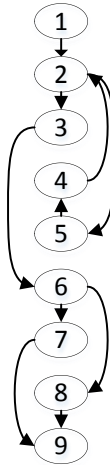
Property 2: $(\exists P) (\exists Q) (P \equiv Q \ \& \ |P| \neq |Q|)$ i.e. even though there may exist 2 programs that have same functionality, but the complexity of each program depends upon its implementation.

This property places emphasis on the effect of different implementations of a program on its complexity. We consider two functionally equivalent programs uniquely different based on their implementation procedures, which are converted to CFG's for calculating the complexity using the proposed metric. Figure 22 illustrates 2 control flow graphs developed from a C program and its optimized version. These programs are adapted from [Venkatachalam *et al.*, 2012], where the authors optimize a C program using graph-mining techniques.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int counter,a,b,n,c;
n = 10;
a = 1; b = 2;
for(counter=0;counter<n;counter+
+){
c = 20;
a=b+1;
b=a+b; }
c = c+a;
if(a<b)
a = 100
else
a = 1000
printf("%d",a);
getch();}
```

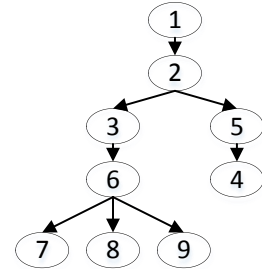
1. Program body P

CFG



```
BLOCK 1
n = 10
a = 1
b = 2
counter = 0
BLOCK 2
L1:
t1 = counter<n
If t1 goto L2
BLOCK 3
goto L3
BLOCK 4
L4:
counter++
goto L1
BLOCK 5
L2:
c = 20
t2 = b+1
a = t2
t3 = a+b
b = t3
goto L4
BLOCK 6
L3:
t4 = c+a
c = t4
t5 = a<b
If !t5 goto L5
BLOCK 7
a = 100
goto L6
BLOCK 8
L5:
a = 1000
BLOCK 9
L6:
Print a
```

CFG



2. Program body Q

Figure 25: Two programs with same functionality and different implementations along with their control flow graphs [Venkatachalam et al., 2012].

Calculating the complexity measures for program bodies P and Q in Figure 25 using the proposed metric, $|P| = 3$ and $|Q| = 2.58$. Therefore, for $P \equiv Q$: $|P| \neq |Q|$ showing that the proposed metric satisfies this property.

Property 3: For any non-negative number c, there are only finite programs with complexity c.

This property is further build on Property 1, addressing a complexity metric's ability to distinguish between the programs with the same decision structure that perform few computations and those which perform many computations. The proposed metric, which considers the execution time at nodes, can distinguish the complexity of a computation, based on the fact that nodes that perform few computations take less time, compared to the nodes that perform many computations. Therefore, this property is satisfied.

$$(\forall P), |P| \geq 0$$

Property 4: $(\forall P)(\forall Q)(|P| \leq |P; Q| \text{ and } |Q| \leq |P; Q|)$ i.e. the individual complexities of a given program body should always be less than or equal to the complexity when they are concatenated.

The emphasis here is on the increase in complexity when a program body is composed by combining two programs (children programs) and that the individual complexities of the two programs are always less than or equal to their parent. To illustrate this, we consider three different control flow graphs where in which the individual complexities of CFG's along with the complexity when they are combined are calculated.

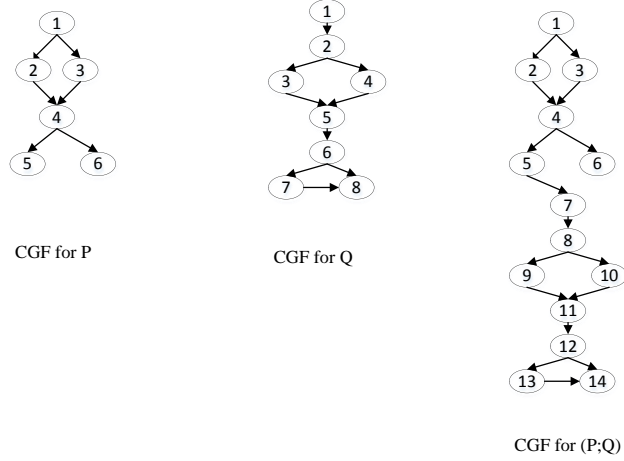


Figure 26: Illustrations of CFG's for (P), (Q) and (P;Q)

From figure 26, the complexities when calculated using the proposed metric $|P| = 3$, $|Q| = 3.58$ and $|P; Q| = 6.58$, which illustrates that this property is satisfied. Hence, whenever two different control flows (extracted from a program) are concatenated, an increase in total number of inputs, outputs and decision structure are observed. This increase in the number of inputs, outputs and decision structures results in an increased complexity. Therefore $(\forall P) (\forall Q) (|P| \leq |P; Q| \text{ and } |Q| \leq |P; Q|)$.

Property 5.1: $(\exists P) (\exists Q) (\exists R) (|P| = |Q| \& |P; R| \neq |Q; R|)$

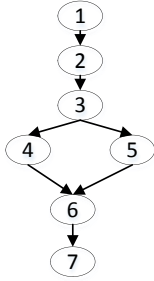
5.2: $(\exists P) (\exists Q) (\exists R) (|P| = |Q| \& |R; P| \neq |R; Q|)$ i.e. If there exist two program bodies P and Q with same complexity, when a new program body R is concatenated with P and Q the complexities will differ.

This property places emphasis on identifying the interactions, which may significantly impact the complexity of a program when concatenated with an external program body. To illustrate this, we consider three different program bodies P, Q and R where, P is a C code to identify if a given number is even or odd, Q identifies if a given number is greater or less than numerical 10

and R identifies if a given number is prime or not. Figure 27 illustrates program bodies P, Q & R, and their respective CFG's.

```
#include <stdio.h>
Int main()
{
    int num;
    Printf(" Enter a number to be checked: ");
    Scanf("%d", &num);
    If ((num%2)==0)
    printf("%d is even", num);
    else
    Printf("%d is odd",num);
    Return 0;
}
```

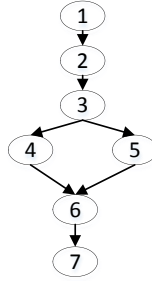
A. Program Body P (To check if a number is even or odd)



D. CGF of Program body P

```
#include <stdio.h>
Int main()
{
    int x;
    Printf(" Enter a number to be checked: ");
    Scanf("%d", &x);
    If ((x)>10)
    printf("%d is greater", x);
    else
    Printf("%d is lesser to 10",x);
    Return 0;
}
```

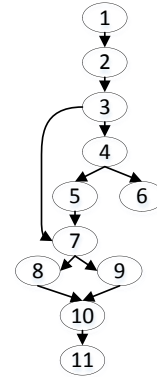
B. Program Body Q (To check if a number is greater or lesser to 10)



E. CGF of Program body Q

```
#include<stdio.h>
int main()
{
    int num,i,count=0;
    printf("Enter a number: ");
    scanf("%d",&num);
    for(i=2;i<=num/2;i++){
        if(num%i==0){
            count++;
            break;
        }
    } if(count==0 && num!= 1)
    printf("%d is a prime number",num);
    else
    printf("%d is not a prime number",num);
    return 0;
}
```

C. Program Body R (To check if a number is Prime or not)



F. CGF of Program body R

Figure 27: Program bodies P, Q, R and their respective Control Flow Graphs

In order to check if the proposed complexity measure holds this property, we concatenate program body R to program body P and to program body Q. Figure 28 illustrates program bodies (P;R) and (Q;R) and their CFG's.

From Figure 24, $|P| = |Q| = 1$ and $|R| = 5$. When program body R is concatenated to P and Q, based on the control flow graphs illustrated in Figure 28, it is observed that $|P; R| = |Q; R| = 6$. Although complexity when calculated using control flow graphs from Figure 25 is same

for both program bodies, it is to be noticed that in program body (Q; R) there is an additional assignment in line 14 of the number being considered to a variable x ('num == x') at node 6. This assignment increases the execution time at this node when compared to its execution time in (P; R). Therefore $|P; R| \neq |Q; R|$, which implies that this property holds for the proposed metric.

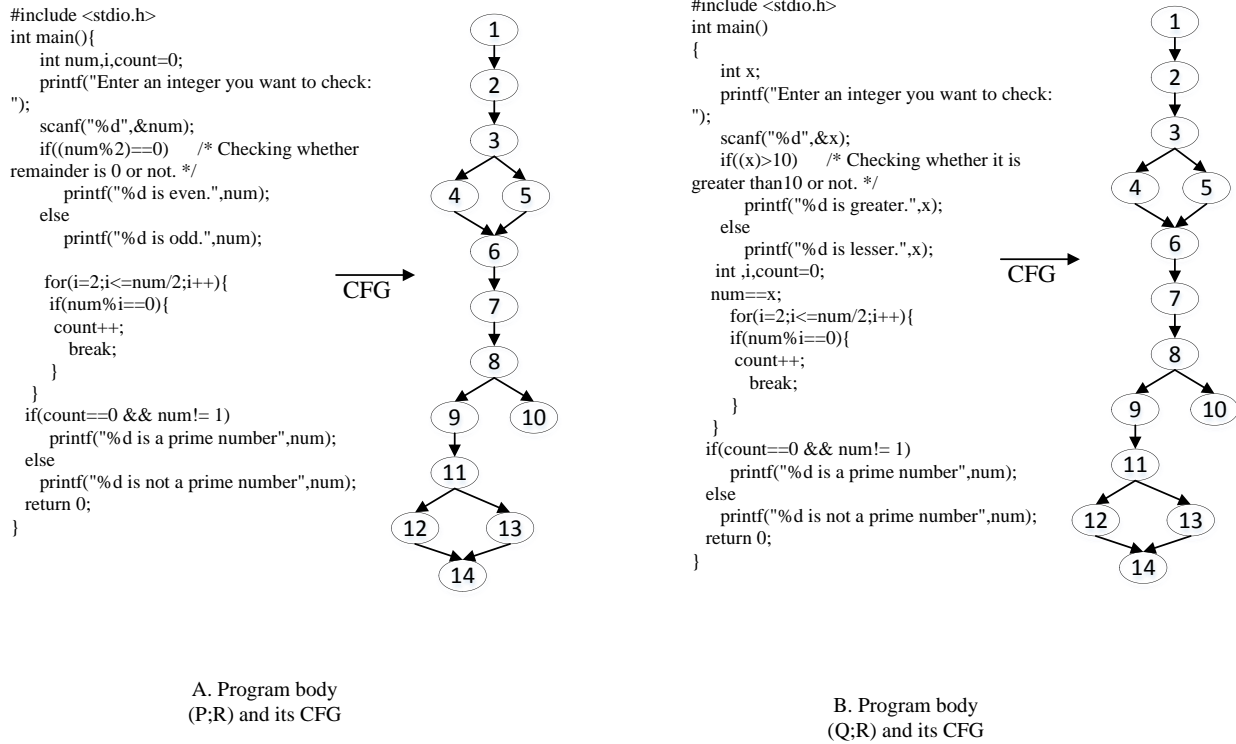


Figure 28: Program bodies (P; R) and (Q; R) and their respective Control Flow Graphs

Property 6: Two program bodies P and Q exist such that, Q is formed by permuting the order of statements of P and $|P| \neq |Q|$.

This property signifies the importance of permuting program statements, with the effect to be considered while quantifying a programs complexity. This property doesn't hold valid for the proposed metric as the nodes in the control flow graphs are independent of the program

statement's placement. Also, the execution time remains the same when given sets of statements are reordered.

Property 7: If program bodies P and Q are almost identical, then $|P| = |Q|$.

This property clearly hold valid for the metric. This is because, if the names chosen for identifiers (different mnemonics) are indeed different, the interaction & control flow patterns along with time of execution still remain the same. This phenomenon also holds valid if there is a change observed in the operators or the constants used in two identical program bodies while all the other factors remain same.

Property 8: $(\forall P) (\forall Q) (|P| + |Q| \leq |P; Q|)$ i.e. interaction of any two-program bodies always increases complexity.

This can be clearly observed from Figure 6 where, $|P| = 3$, $|Q| = 3.58$ and $|P; Q| = 6.58$ that this property holds valid for the proposed complexity measure.

Table 15: Metric comparison to other complexity measures using Weyukers criteria ('YES' indicates property is satisfies and 'NO' indicates property is not satisfied)

<i>Weyukers Property Number</i>	<i>Statement Count</i>	<i>Cyclomatic Number</i>	<i>Effort Measure</i>	<i>Data Flow Complexity</i>	<i>Proposed Metric</i>
1	YES	YES	YES	YES	YES
2	YES	YES	YES	YES	YES
3	YES	NO	YES	NO	YES
4	YES	YES	NO	NO	YES
5	NO	NO	YES	YES	YES
6	NO	NO	NO	YES	NO
7	YES	YES	NO	YES	YES
8	YES	YES	NO	NO	YES

As observed the proposed metric is mostly compliant with Weyukers Criteria. Illustrated in Table 15 is the complexity metric evaluation according to Weyuker's criteria for measures based

on Statement Count, Cyclomatic Number, Effort Number, Data Flow Complexity [please refer to Weyuker, 1988 for a detailed analysis] and the proposed metric.

When closely examined, this evaluation helped to identify the key properties of the proposed metric, where:

- It is sensitive to how components interact based on control and data flow,
- It will not rank all the programs to be equally complex,
- It divides programs into various classes of complexity,
- It is sensitive to program syntax,
- The complexity measure increases as a program grows.

This also identifies one key weakness of the measure. It is unable to distinguish the pattern in which the statements of a program appear.

To summarize, the proposed entropy based measure is defined considering the program component (node) interactions (i.e. their control and data flows), likelihood of component occurrence, and the time of execution at each node, to calculate software code complexity. Evaluation against Weyuker's criteria helped to support metric validity for practical use.

Chapter 4 – Application of Entropy in Engineering Education

Introduction to Engineering Education

“Engineering Education” includes the activities of imparting knowledge based on principles and practices related to the field of engineering in a structured educational environment. Classrooms simulate broader social systems [Cohen, 1972], with meaningful interactions among students and teachers providing structured, productive educational environments where activities, tasks, and expectations are defined and achieved based on students’ interests. Various educational factors of a classroom, such as training, curriculum development, testing methodologies, student classroom structure, student prior experience, and so forth, are often taken into account separately. These individual factors can be utilized for assessing the outcome of classroom effectiveness, and together the factors determine whether the whole educational effort has made a classroom synergistic [Brown, 1992]. Figure 29 illustrates the contributions of the mentioned factors in a classroom.

Elizabeth Cohen [1972], in her paper on setting the conditions for teacher-student interaction, suggests that there is a need for research to be oriented towards teaching effectiveness concerning classroom content and the quality of interactions among teachers and students. She raises the problem of extrapolating a set of ideas and procedures, which are useful in a tutorial setting, towards a classroom setting wherein the interaction patterns among the students and teachers may vary significantly.

“Sociological studies of classroom interaction raise a serious question for the research on teaching effectiveness. Teacher effectiveness studies typically concentrate on changing what the teacher says and the quality of teacher-student interaction. But there is a problem in assuming

that the learning of thirty students in a classroom can be understood with the same set of ideas useful for understanding learning in a two person tutorial situation. If I am a student and if I have a teacher who explains things very well, who asks me questions broadly, who makes students extend answers to questions and who frequently reinforces, it is thought that I will learn. But what if I never raise my hand, sit in the back of the room, often fail to listen and rarely engage in question-answer interchange with my teacher? Will I receive the same benefits as the eager student who sits up in front and has all the direct interaction with the teacher?”[Cohen, 1972, pg.443]

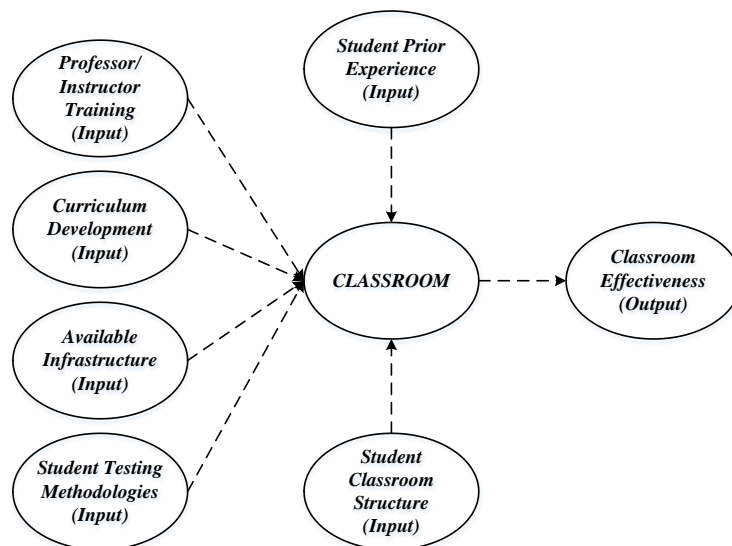


Figure 29: A Complex Synergistic Classroom

Interactions are one of the most important factors in a classroom. In a class, interactions can be student-initiated or teacher-initiated, and either between student-and-student or teacher-and-student(s). Fundamental learning activities in any classroom are comprised of class discussions, student debates, course disclosure, and sharing of opinions, which often lead to the creation of a scenarios of interactions which go towards identifying new ideas, thereby meeting classroom

learning objectives [Anna Ya, 2013]. Student-to-student and student-to-instructor interactions in a classroom create a sense of community and a sense of trust, which helps to achieve shared goals [Davies & Graff, 2005].

Interactions patterns (different, considered interaction patterns are briefly explained in the later sections) can be stimulated and created in a classroom totally at the discretion of the instructor, but students can also self-organize into these patterns. When interaction patterns are formally quantified, and analyzed together with the measured achievement of students, classroom interaction patterns can be helpful in predicting the effectiveness and efficiency of classroom teaching. In this dissertation, Entropy, as defined in information theory, is used to formulate an assessment method for interaction patterns among students and instructors based on given classroom structural settings.

Interactions and Classroom Structures

The word “interaction” is a *phenomenon where two or more objects have an effect upon each other*. Combinations of such simple interactions create webs of interconnected activities, and may give rise to emergent phenomenon. The word interaction in educational settings refers to person-to-person communication, and thus exists in in-class discussions, in-class debates, students raising doubts while in lecture, student–student discussions, student-professor discussions, and so forth. Interaction Hypothesis, proposed by Second Language Acquisition (SLA) expert Michael Long (as cited in Rod, 1991) elaborates on a way in which ESOL (English for Speakers of Other Languages) students can learn a target language by conversation. This implies that interactions help in developing language proficiency in ESOL students. Similarly, in educational settings, active academic collaboration among professors, peers and students is

gaining more importance and attention [Kang, 2007]. Educational settings that emerge from such collaborations give rise to a network of interactions among the stakeholders (stakeholders here refers to professors, lecturers, peers and students) involved. Figure 30 illustrates a model of emergence starting from a motivated individual initiating person -to- person communication giving rise to a web of communications that lead to emergent patterns. The effect of such interactions gives rise to a classroom structure, a structurally defined implication for organizing the notion of a classroom. Please refer to Table 16 for different classroom structures and their associated interaction patterns.

Table16: Classroom Structures and their Associated Interaction Patterns

<i>Classroom Structural Setting</i>	<i>Associated Interaction Pattern</i>	<i>Network Topology Observed</i>
Traditional Lecture	One-to-All	One-to-All
Peer-Based Team Learning	Hierarchy	Star
Flipped Classroom	Self-Organizing – fully linked	Fully-connected Ring, Mesh

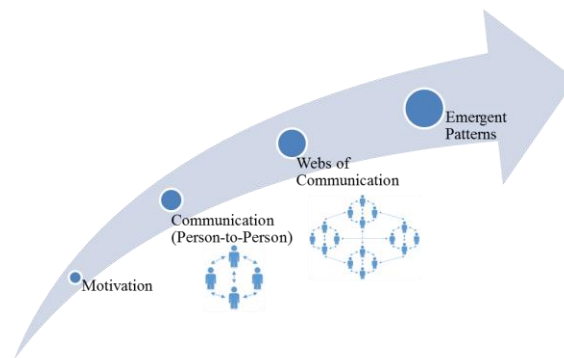


Figure 30: Emergent Model

A brief introduction on Traditional classroom setting, Flipped Classroom setting and Peer based learning are given below.

According to Hannum & Briggs (as cited in Relan & Gillani, 1977), a ***Traditional classroom setting*** can be defined as an instructional environment that encourages passive learning [Hannum & Briggs, 1982][Relan & Gillani, 1997] where a lecturer/professor gives lecture and students listen and take notes. *Traditional classroom* structure is usually comprised of a one-to-many teaching structure as illustrated in Figure 31, where the lecturer plays a crucial role in the course disclosure. It is a structured 2-step process where lecturer transmits information and students receive and process the information. The classrooms tend to be space bound, implying that learning takes place inside a physical boundary such as in classrooms, schools, universities and several similar locations [Relan & Gillani, 1997]. In regards to student learning in traditional settings, Felder & Silverman point out that students sometimes tend to selectively receive and process certain information and ignore the rest. This sometimes results in a non-uniform distribution of student learning [Felder & Silverman, 1988].

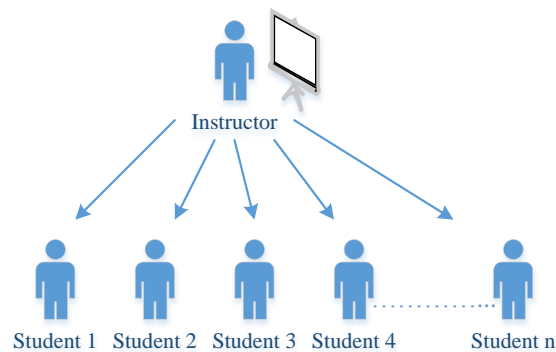


Figure 31: Traditional Classroom Setting

On the contrary, many educational institutions are now embracing the technique of *flipping a classroom (Flipped Classroom Setting)*. Flipping a classroom extends the boundary of learning with many forms such as web-based learning, video-based learning, interactive laboratories, interactive classrooms, peer based learning, and so on, inverting upon traditional classroom

structures [Berrett, 2012]. According to Harrison Keller, Vice Provost for higher-education policy at University of Texas at Austin, Flipped classrooms in educational & research institutions which have big classes, allow students to be more productive [Berrett, 2012]. A survey of literature based on several case studies identified in [Bishop and Verleger, 2013] suggests that there is a significant increase observed in the learning curve of the students in flipped classrooms, along with an increase in their participation and inter-activeness. According to Lage *et al.*, the inverted (flipped) classroom can be defined as a setting where the events (such as preparation for discussion) that usually take place inside a classroom, in a traditional setting, now take place outside the classroom, before class sessions. An example the authors use here to describe the inverted classroom involves the use of the World Wide Web and multimedia communications as aids that give the opportunity for students to view lectures before class [Lage *et al.*, 2000].

Bishop *et al.*, define a flipped classroom as an educational setting that promotes: (a) Computer based individual instruction outside the classroom, and (b) Interactive group learning activities inside the classroom. The term *flipped classroom* is often assigned to courses or activities that use asynchronous web based lectures and closed problems or quizzes, thereby expanding the curriculum in contrast to traditional course settings [Bishop and Verleger, 2013], where lack of student pre-preparation leads to default attendance of lectures. A simple illustration of a flipped classroom is shown in Figure 32. Bishop and Verleger [2013] suggest that though there are many studies based on flipped classrooms, there is no much evidence supporting the influence of flipped classroom in student learning improvement; this is because, there is no sufficient evidence to generalize the results as they are all very situation specific, creating a need towards additional research for examining the influence of flipped classroom instruction on

student learning outcomes [Bishop and Verleger, 2013]. Please see Bishop and Verleger [2013] for a detailed analysis on flipped classroom research.

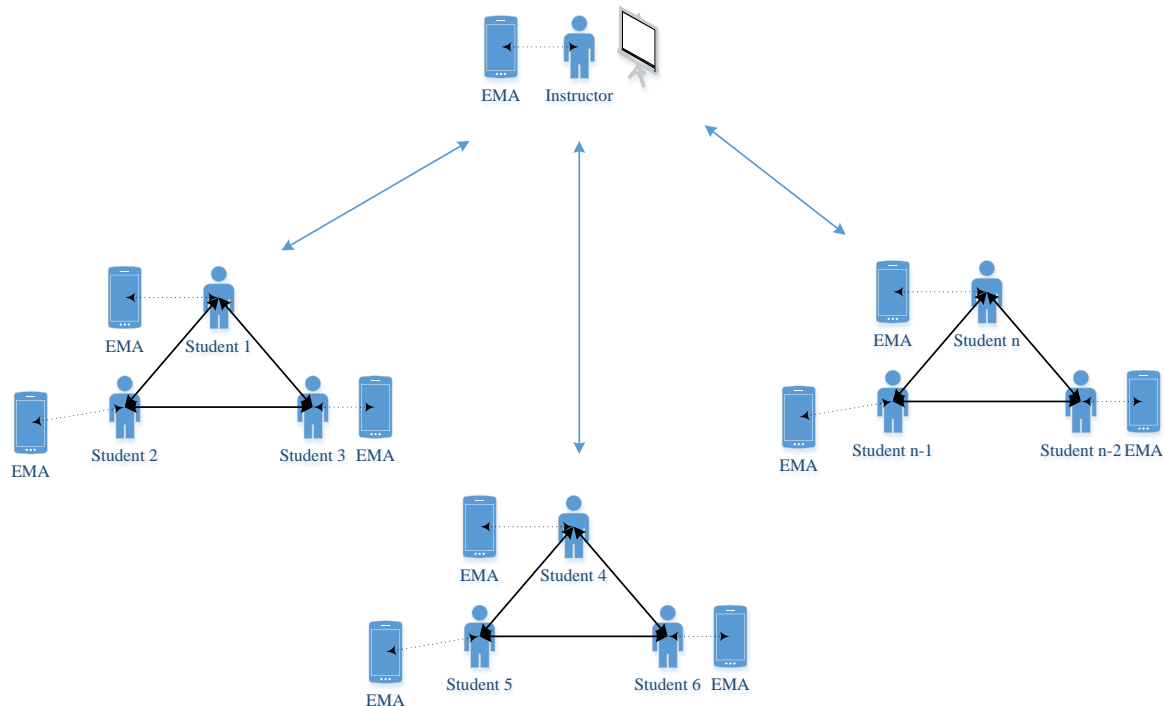


Figure 32: A Simple Flipped Classroom Setting (Note: EMA stands for External Media Access)

Peer based learning addresses the bidirectional peer relationships that help to facilitate professional and personal growth. The three key factors that makes peer based learning versatile are [Eisen, 2001]: (a) Peer learning stimulates a learners’ own knowledge, thereby motivating the learner to learn (b) Human interactions create a drive to improve both parties involved (here students and peers), creating a collaborative accountability for given tasks (c) Peer learning increasing a high knowledge transfer potential based on the parties involved and their background. Peer based learning, also known as Peer-Assisted or Corporative learning according to Topping and Ehly, as cited Bishop and Verleger [2013], can be defined as “the

acquisition of knowledge and skills through active helping and supporting among status equals or matched companions” [Bishop and Verleger, 2013]. It is a pedagogical method aimed for students to learn and work together in small groups towards enhancing their own and each other’s skills. Usually, in a classroom, peer learning consists of activities involving a group of students along with their peers to discuss conceptual questions along with actively engaged problem exploration and solving [Coetzee, D., et al., 2015]. “Peers’ here refers to either: senior students mentoring junior students, or, students from the same year forming partnerships.

Assessment of Classroom Structures

Current trends towards constantly increasing student motivation and to keep them actively engaged in a classroom is leading to a discovery of new methods and techniques to be used in a day to day classroom setting. Assessment is an inherent part of instructional practice for identifying how well a given technique influences students learning ability. One typically thinks of assessment as a quantitative index of success in educational settings and thereby a helpful aid to promote student achievement [Chappuis and Stiggins, 2002]. As cited in Liang and Creasy [2004], Angelo and Cross [1993] defined assessment as a multidimensional process of appraising the learning that occurs in a classroom before and after assignments are graded [Liang and Creasy, 2004]. Table 17 summarizes five different journal papers on approaches used and assessment techniques being adopted to evaluate student performance, to show how new assessment methods and practices for assessing student performance have emerged along with new educational settings.

Schools and universities need updated assessment methods in-order to make informed decisions. The most common examples of assessment methods observed are: tests, quizzes, open ended

questions and questions consisting of problems. These methods are also being used to measure student learning and to evaluate the type of educational settings used, with a direct correlation to quantitative student performance [VanLehn and Martin, 1998].

Table 17: Example Teaching approaches and Assessment Methods

<i>Authors</i>	<i>Title</i>	<i>Classroom Context</i>	<i>Subjects</i>	<i>Data Collection</i>	<i>Assessment Method</i>	<i>Conclusions/Remarks</i>
Xin Lang and Kim Creasy, 2004	Classroom Assessment in Web-Based Instructional Environment: Instructors Experience	Web-Based Instructional Environment	10 Faculties & 216 Students via 16 WebCT classes	Field notes, Transcriptions, class room artifacts, Interviews and Classroom observations, Written Assignments and Tests	Numeric Score, Evaluations Rubrics	Need to reengineer the assessment from traditional practices towards “assessment for learning” based on the shift in the mode of communication
Zappe, Leith, Messner, Leitzinger, and Lee, 2009	Flipping the classroom to explore active learning in a large undergraduate course	Flipped Class room via online videos and problem solving during class	100 students over a period of 2 years	2 Course Surveys	Student Feedback	Flipped Classroom via online videos and problem solving during class is a more effective method of teaching than regular setting.
Ruddick, 2012	Improving chemical education from high school to college using a more hands-on Approach	Flipped classroom by use of video lectures at home and problem solving in class	High School chemistry students	In class Exams, Course evaluations and SALG survey (Student assessment of their learning gains)	Final exam scores, survey analysis & student feedback	Flipped classroom students outperformed the standard lecture based students with higher final exam scores
He, Yi, Sandra Swenson, and Nathan Lents, 2012	Online video tutorials increase learning of difficult concepts in an undergraduate analytical chemistry	Use of video tutorials	27 undergraduate general chemistry course students	Oral and written feedback, pre and post video exams	pre and post video exam performance	Online tutorials are valuable, flexible and cost effective tool to improve student learning for mastery of chemistry problem solving

McLaughlin, Jacqueline E., et al, 2014	The flipped classroom: a course redesign to foster learning and engagement in a health professions school	Self-Paced online videos and student centered activities	First year pharmacy students	Exams, quizzes, Engagement, Projects, pre-post Students Survey	3-Midterm exam scores, Project evaluations, survey Analysis and Cumulative Final	Flipping the traditional classroom proved feasible and necessary to educate large cohorts of students. Improved exam performance, increase in confidence levels was observed
--	---	--	------------------------------	--	--	--

It can also be seen from Table 17 that sometimes tests are seen as measures of the underlying classroom setting to determine the validity of its use. Also, the rationale by the researchers to support the adoption of a given setting is highly skewed towards other modes of assessment such as pre- and post-course survey, course evaluations and cumulative scores. With an increase in technology, many fundamental changes are being observed in learning, teaching and assessment methods. For example, networks, an integration of Internet, computers, intranets and humans, offer new forms of instruction and assessment [Gibson, 2003]. A vast database of literature is available online regarding assessment of student learning. Assessment in literature is observed to be solely based on student performance. As cited in Gibson [2003], *“Assessment is most effective when it reflects a multidimensional, well integrated structure for improving student learning and evaluating performance.”*

As illustrated in Table 17, there is an increased trend emerging on identifying and applying several-flipped classroom learning techniques to keep students motivated. This brings into the picture a need for assessment techniques that can be applied for any type of flipped setting considered for a classroom. With this in mind, this paper uses student interaction patterns, individual knowledge levels and the concept of entropy for an assessment method that can potentially be applied to evaluate a given educational setting used in a classroom.

Information Processing Theory

Before understanding the concept of entropy, and its validness of use in educational settings of complex human interactions, information processing theory will be explored. The stakeholders involved in an educational setting are involved with Transmitting, Receiving, and Processing, Understanding and thereafter Storing information for future access. Information processing theory models, currently used in several research areas such as cognitive development, neuroscience, social learning and artificial intelligence, helps to understand how the human mind deals with input information from the environment using cognitive systems. According to this theory, quantitative and qualitative methods can be used to understand memory. Qualitative methods involve participants reading a list of words or numbers and later recalling the list based on what they remember from the original. An example of a Qualitative method would be to measure verbal representations of memory based on word usage of a participant to measure the change of word representations [Adams *et al.*, PSYSC613, accessed on 09 Feb.2015].

The approach to information processing theory explores the mind in detail, based on the flow of information [(Lachman *et al.*, 1979) as cited in Casey and Moran, 1989] resting on the following assumptions [Casey and Moran, 1989]:

- Human mind is regarded as a general-purpose symbol processing system.
- Information is represented symbolically in mind.
- Both computer and human mind are regarded to carry out tasks in a series of programmed steps that occur in a sequence of successively programmed states.
- Information processing involves the tracing and reduction of mental operations to component processes
- The information processing systems is organized into stages

The basis of the first two assumptions, and of qualitative and quantitative methods in information processing theory, will be used in the later sections for developing entropy based educational assessment model.

Concept of Entropy

Information can broadly be defined as a set of data that is either organized for a specific purpose, or presented within a specific context, giving it meaning and relevance. Information can also be defined as a collection of content in terms of text documents, audio recordings or data files [Machta, 1999]. In the context of this paper, information can be defined as data transferred among the participants of a classroom in an educational setting. This transfer of data can be looked at as either an instructor or a professor imparting knowledge in regards to a specific subject, instructors and students answering questions in a class, students understanding the content of a course and submitting homework's, or, conclusions drawn by group discussions among students. Acknowledging Machta [1999], "*information content, measured in bits (basic unit of information used in digital communications), of a text document, audio recording, or data file is the number of ones and zeros needed to store the text, sound, or data.*" History of relationship between information and entropy dates back to the first half of the 20th century, based on Szilard's analysis on Maxwell's Demon and Shannon's work on communication theory [Machta, 1999].

Information theory is a branch of mathematics based on the theory of probability and statistics. It is applied in various fields where probability and statistics play an important role. One such field is modern communications theory that formulates communication systems as a set of random or stochastic processes. Shannon played an important role in emphasizing the essential

mathematical and statistical nature of information theory [Kullback, 1997]. For a detailed and extensive review on the history and various definitions of Information Theory, please refer to [Kullback, 1997]. Information theory was developed by communication engineers for quantifying the information flow through information channels such as computers systems and telephone lines [Proctor and Van Zandt 2008]. Information Theory not only plays an important role in digital communications, but also in many other field involving complex human interactions. As cited in Proctor and Van Zandt [2008]: (a) Kang and Seong [2001] used the information theory approach to quantify the perceived complexity of control room interfaces in nuclear power plants, along with estimating the extent to which the interfaces would overload an operators capacity for processing information, (b) Strange et al. [2005] used information theory for a quantitative analysis on brain activity involved in visual perception, specifically, the hippocampus activity based on event uncertainty.

Information theory is a system of measurement used for quantifying information conveyed by the occurrence of an event. This event can be either a response or a stimulus that is a function of a number of possible events and their probabilities. For example, if an event is sure to occur, then it conveys no information and if the occurrence of the same is uncertain then it implies it has associated information [Proctor and Van Zandt 2008]. For example, if it is observed that a particular student never shows up to a class, and then on a heavily snowing day if the same student doesn't show up, then there is no information gained by this event. On the other hand, if it is observed that a student who shows up every day to the class doesn't show up on heavily snowing day, new information is gained from the same event. This implies that the uncertainty of an event is the amount of information gained by observing that event.

So what is Entropy?

Entropy is a broad term that can be looked at in several different ways. In thermodynamics, entropy is defined as a state function associated to the amount of heat involved in a transformation process in relation to the net work done in a system. In statistical mechanics, Boltzmann defines entropy as a measure of the number of possible microstates of a system in thermodynamic equilibrium. According to Gibbs, entropy is the summation over all the possible micro states of a system based on their probability distributions [Machta, 1999]. In information theory, according to Shannon, Entropy helps to quantify the information [Shannon and Weaver, 1949]. Quantifying information refers to the expected value of the information contained in a message. Shannon Entropy H is given as:

$$H = -\sum P_i \log_2 P_i \quad (1)$$

Where P_i is the probability of a symbol showing up in a given stream of symbols and the use of the logarithm base two corresponds to expressing information entropy in terms of bits.

Application of Entropy to Educational Settings

The concept of entropy is widely being adopted in various fields of research. It was first introduced for physical systems as a measure of disorder. In information theory it plays a crucial role for representing average information and degree of uncertainty [Hamada, 2007]. Its application in the field of educational assessment is fairly new. Despite being fairly new, the application of entropy in educational assessment research suggests that there are several ways the concepts of the measurement of information from information theory i.e. entropy, can be helpful to formulate promising assessment tools. Searching through the literature, limited journal papers were found on different horizons of application such as automatic assessment of student essays,

measuring student competency, assessing learners understanding, evaluating student innovative ability, and assessing student interaction patterns.

Kakkonen and Sutinen (2004) in their paper on “Automatic Assessment of the Content of Essays based on Course Materials” use Latent Semantic Analysis (LSA) for evaluating the content of essays by comparing the conceptual similarity between the essay and the selected text pages from the course material covering the essay specific subject matter. The concept of entropy is used here as a preprocessing technique in LSA for determining word importance. The theoretical information measure of entropy is computed here for each word mapped to a matrix over all the word entries and then it is divided by the summation of entropy of all the word entries in a row. The main aim of this approach is to give higher values to words that are more important and lower values to the less important [Kakkonen and Sutinen, 2004]. Al-Radaideh *et al.*, take a similar approach to build a classification model using a decision tree method. The decision trees here mainly depends upon information gain based metric depending on entropy measure to determine which attribute is most useful [Al-Radaideh *et al.*, 2006]. See Al-Radaideh *et al.* for a detailed explanation this approach.

Liu (2005) uses information theory for adaptive student assessment techniques based on their competency in concepts. A domain in which students should learn a set of concepts where in which some of them are basic concepts and the others are composite (integrated from basic concepts) is considered. Simple Bayesian networks are used to represent parent and child concepts. For example, if C represents parent concept of a test item X then there is no reason to assume that a student’s probability of answering X correctly would decrease when the student gets a hand on C . The concept of entropy is used here in the context where one assumes that the student’s responses to X and Y become independent given the information about students mastery

of C , where there is a positive influence of C on X and Y . Mutual information between two concepts i.e. either C and X or C and Y is computed as a difference of entropy of C and the conditional entropy of either C given X or C given Y [Liu, 2005]. Please see [Liu, 2005] for more details.

Lajis and Aziz attempt to address the assessment of a learners understanding based on a student's response in terms of short free text answers. A hybrid technique incorporating natural language processing, information extraction and artificial intelligence is proposed. They convert a textual answer into a node link representation for extracting the hidden knowledge structure and then apply entropy to compute the amount of information. Here, they calculate the amount of student understanding as a difference of entropy of a given node with the total entropy of all the nodes [Lajis and Aziz, 2010] [Lajis and Aziz, 2009] [Lajis and Aziz, 2008]. Refer to Lajis and Aziz [2008, 2009, and 2010] for a detailed explanation on this approach.

For determining if the students of a class individually or as a whole understood and acquired knowledge on solving a given set of problems, Bickel [2010] suggest the use of entropy where, entropy of each students assignments on each question is calculated. The authors suggest that this approach provides insight into how certain or uncertain a class is about a particular question where, lower entropies imply that class was more certain of a particular answer [Bickel, 2010]. Please see Bickel [2010] for a detailed study on scoring rules and decision analysis techniques in education.

For a quantitative evaluation of the innovative ability of engineering students, authors Tan and Zhou [2014] combine the method of entropy and co-ordination degree to evaluate the innovative ability. An evaluation index system incorporating factors such as Knowledge of a given theory, Judgment ability, Practice ability and Research ability is suggested for measuring innovative

ability. Based on the relationship between entropy a degree of order, entropy is used by the authors to determine the weights of their suggested indices. The 12 indices chosen by the authors were viz. Scores of Professional Subjects, Degree of mastering theoretical knowledge, Autonomous learning ability, Performance in class, Degree of understanding in subjects, Operating level of basic experiments, Social practice ability, Views on subjects, Invention of a patent, Publication of papers, Literature search ability, Instructor/mentor comments [Tan and Zhou, 2014]. Refer to Tan and Zhou [2014] for a detailed explanation on analysis of the context on which the indices were defined, their definitions and associated use of entropy concept.

Snow et al., in their paper on, “Entropy: A Stealth Measure of Agency in Learning Environments”, examine how student’s patterns of interactions with game based features which are more controlled and systematic influence the quality of their performance compared to students who interacted with the system in a more disordered fashion. The concept of entropy was used here to calculate Euclidian distances generated by students based on their random walks. According to the authors, their analysis is one of the first ones to use entropy as a means to provide a stealth assessment of student interaction patterns with in a tutoring environment. Refer to Snow et al., for a detailed analysis and explanation on how they employed random walks, Euclidian distances and entropy to capture student interaction patterns.

Entropy based Assessment Model

Entropy is widely used by researchers across several disciplines to measure the order of a system. Gibbs defined entropy as a distribution of microstates in statistical thermodynamics, Boltzmann used entropy in statistical mechanics, Shannon used entropy to measure information in communication channels, and, Mizutani (as cited in Kelly, 1969) used entropy towards

economic behavior. The current work attempts to use the concept of entropy to education for measuring knowledge and interactive effectiveness among students and facilitators in a considered educational setting.

Knowledge implies to the skills and information acquired by a person through experience, education and understanding of given subject or a concept. Awareness and the familiarity of knowledge can be quantitatively measured using several widely accepted techniques. Knowledge acquired by a student is a function of activities and interactions in a classroom. All the inputs from the multiple pieces of information acquired by a student in class and outside the classroom are structurally correlated together forming a meaningful cognitive structure of information on a given context as illustrated in Figure 33.

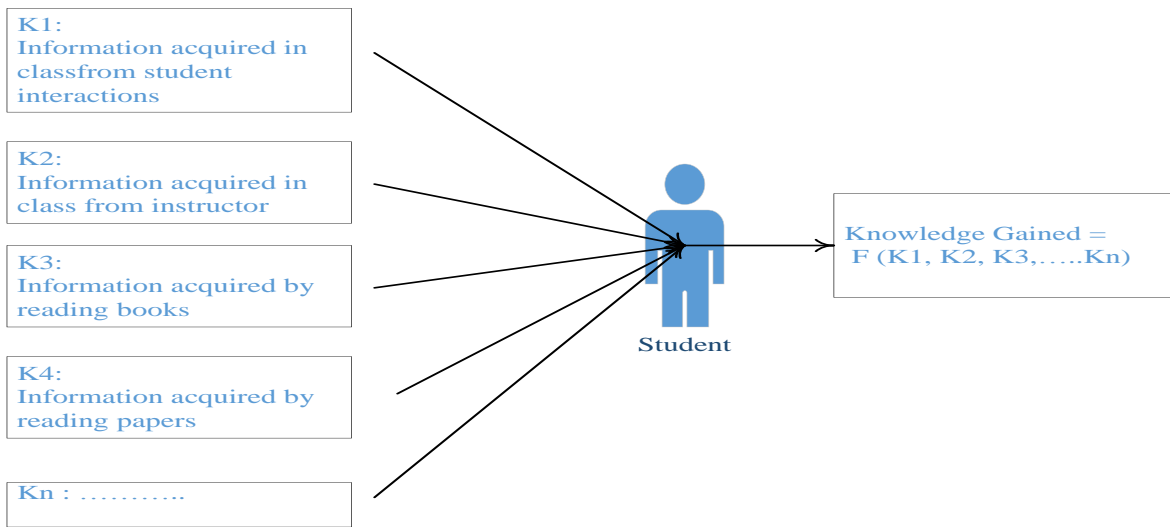


Figure 33: Student Knowledge Structure

Pencil/Paper tests, web based assessments, problem solving, modelling, measuring independent pieces of information, multiple answers, open constructed response, closed structure response and similar methods are the examples of a few knowledge measurement techniques used in education [Creasy and Liang, 2004], [Schleicher and Tamassia, 2000]. The mentioned

techniques mainly involve gathering recorded (written) text from students. The text content of each student significantly differs from others based on their understanding of a given concept. Though sharp changes may be observed among students textual recordings, their understanding is conveyed by the use of similar keywords from a concept, thereby establishing validity for measuring the context. Keywords help portray important information about the content of a document providing an effective search mechanism for the users in categorizing and retrieving information. The idea of using keywords that frequently occur in a document for measuring the information content is widely used. Refer to Liu et al., 2009 for details on different studies based on keyword extraction.

The concept of Entropy, a statistical parameter that measures the information produced on average, is initially applied in this model, based on the probability distribution of keyword occurrence from gathered text recording. These text recordings are solicited from students based on their understanding of a concept and instructors knowledge based used for imparting the concept. Entropy based on text for each participant is then calculated using the probability distribution of the keywords. Let TE be called the entropy calculated based on keyword occurrences observed in a text. Then,

$$TE_I = - \sum P_i \log_2 P_i \quad (2)$$

$$TE_{S_n} = - \sum P_i \log_2 P_i \quad (3)$$

In which, i represents a specific keyword,

P_i Is the probability distribution of the keyword i ,

TE_I Is the entropy calculated based on keyword occurrences in the instructor's knowledge base,

TE_{S_n} Is the entropy calculated based on keyword occurrences from text recording of a student n .

Sharing the same view as the authors, Vetromille-Castro [2013], states that in complex systems such as classrooms inter-individual interactions play a vital role in formation of meaningful structures. These interactions act as fuel to such complex systems, in this case – Classrooms. These continuous interactions among students and teachers bring the system to existence. Three different types of interaction patterns considered for formulating entropy based assessment. Three different entropy based assessment models are first developed for each setting individually which will then be generalized for application towards any considered classroom setting.

Traditional Classroom Setting

The assessment model for a traditional classroom setting is developed based on the following assumptions:

- A Traditional Classroom can be characterized into two different hierarchical levels as illustrated in Figure 34.
 - Level 1: Instructors are placed in level 1 (higher level) of a classroom
 - Level 2: Students of a class are all placed in level 2 of a classroom
- Entropy based on keyword occurrences of an instructor's knowledge base is always considered to be less when compared to that of a student's text recording. This is because entropy here, being associated with probability distributions, keywords with low probability distribution will have more entropy and keywords with high probability distribution will have less entropy.
- Validity of the above assumption is based on the fact that Instructors knowledge in a given subject will always be higher when compared to a student's knowledge attending a class on that subject.

- The interactions considered in this setting are limited to: Instructor-Student interactions and Student-Instructor interactions.

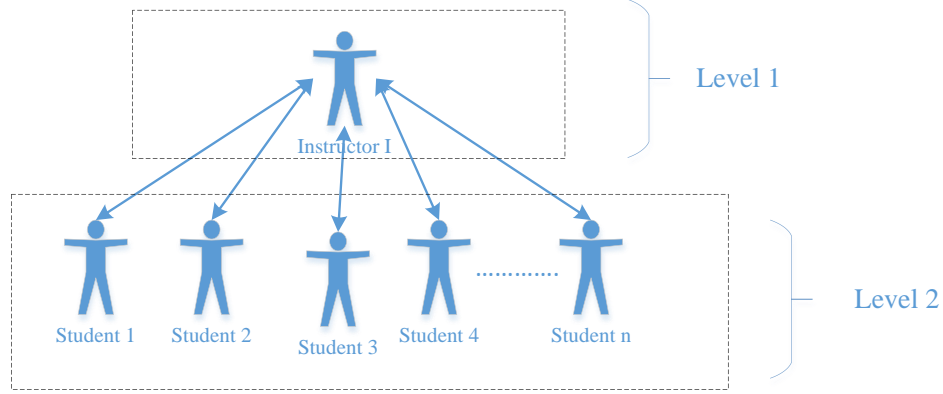


Figure 34: Hierarchical Representation of a Traditional Classroom Structure

The assessment model for this setting can be formulated as shown below.

$$E = E_1 + E_2 + \cdots \dots \dots + E_n \quad (4)$$

Where, n is the number of students in a class and,

$$E_n = \left[\frac{1}{\left(\frac{TE_I}{TE_{S_n}} \right)} \right] * (I_{S_n}) * (-\sum P_n \log_2 P_n) \quad (5)$$

In which,

P_n Is the probability obtained based on the output distribution of student n . This probability distribution is calculated based on the number of interactions made by the student n in a class,

I_{S_n} Is the number of inputs observed in the class to a student n ,

TE_I Is the entropy calculated based on keyword occurrences in the instructor's knowledge base,

TE_{S_n} Is the entropy calculated based on keyword occurrences from text recording of a student n ,

$\frac{TE_I}{TE_{S_n}}$ Acts as a weighing function to determine the importance of interaction between the student n and instructor.

Peer-based Learning Structure

The assessment model formulated for a peer based learning structure is based on the following assumptions:

- A peer based learning environment can be characterized into three different hierarchical levels as illustrated in Figure 35.
 - Level 1: Instructors are placed in level 1 (higher level) of a classroom
 - Level 2: Peers are placed in level 2 of a classroom
 - Level 3: Students of a class are all placed in level 3 of a classroom
- Entropy based on keyword occurrences of an instructor's knowledge base is always considered to be less when compared to that of a peer's knowledge which in turn is assumed to be lesser than a student's knowledge calculated based on text recordings. This is because entropy here, being associated with probability distributions, keywords with low probability distribution will have more entropy and keywords with high probability distribution will have less entropy.
- Validity of the above assumption is based on the fact that Instructors knowledge in a given subject will always be higher when compared to a peer facilitating the class, who in turn has higher knowledge than a student's knowledge attending a class on that subject.

- The interactions considered in this setting are limited to: Instructor-Peer interactions, Peer-Instructor interactions, Peer-Student Interactions, Student-Peer interactions and Student-Instructor interactions.

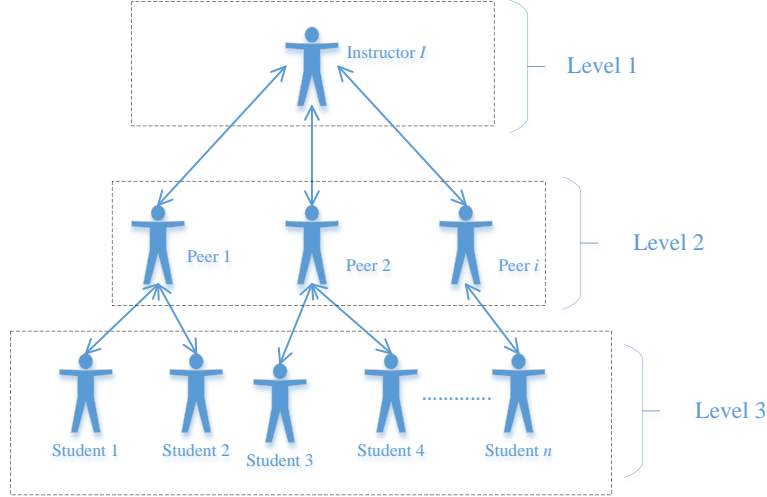


Figure 35: Hierarchical Representation of an ideal Peer based Learning Environment

The assessment model for this setting can be formulated as shown below.

$$E = E_1 + E_2 + \cdots \dots \dots + E_n \quad (6)$$

Where, n is the number of students in a class and,

$$E_n = \left[\frac{1}{\left(\frac{TE_I}{TES_n} \right)} \right] * (I_{S_n}) * - \sum P_n \log_2 P_n \quad (7.1) \text{ or}$$

$$E_n = \left[\frac{1}{\left(\frac{TE_{P_i}}{TES_n} \right)} \right] * (I_{S_n}) * - \sum P_n \log_2 P_n \quad (7.2)$$

In which,

i is the number of peers facilitating a class

P_n Is the probability obtained based on the output distribution of student n . This probability distribution is calculated based on the number of interactions made by the student n in a class,

I_{S_n} Is the number of inputs observed in the class to a student n ,

TE_I Is the entropy calculated based on keyword occurrences in the instructor's knowledge base,

TE_{S_n} Is the entropy calculated based on keyword occurrences from text recording of a student n ,

$\frac{TE_I}{TE_{S_n}}$ acts as a weighing function to determine the importance of interaction between the student n and instructor, and

$\frac{TE_{P_i}}{TE_{S_n}}$ acts as a weighing function to determine the importance of interaction between the student n and a peer i .

Based on the type of interactions being considered, equations 7.1 or equation 7.2 are to be considered. Equation 7.1 is used when the interactions among instructor and students are measured and equation 7.2 is considered to measure the interactions among peers and students.

Flipped Setting – Self-Organizing

The formulated model for a flipped setting is based upon the following assumptions:

- As illustrated in Figure 36 a flipped setting is characterized into a single level without any associated hierarchies where all the students are placed in a same level.
- Entropy based on keyword occurrences of a student's knowledge based on text recordings will significantly differ from each other.

- Validity of the above assumption is based on the fact that all the students taking a class of given subject will have their own perceptions either based on their experience or previous knowledge.
- The interactions considered in this setting are limited to inter student interactions.

The assessment model is formulated as show:

$$E = E_1 + E_2 + \cdots \dots \dots + E_n \quad (8)$$

Where, n is the number of students in a class and,

$$E_n = \left(\frac{TE_{S_n}}{TE_{S_{n-1}}} \right) * (I_{S_n}) * -\sum P_n \log_2 P_n \quad (9)$$

In which,

P_n Is the probability obtained based on the output distribution of student n . This probability distribution is calculated based on the number of interactions made by the student n in a class,

I_{S_n} Is the number of inputs observed in the class to a student n ,

TE_{S_n} Is the entropy calculated based on keyword occurrences from text recording of a student n ,

$TE_{S_{n-1}}$ Is the entropy calculated based on keyword occurrences from text recording of a student $n - 1$,

$\frac{TE_{S_n}}{TE_{S_{n-1}}}$ acts as a weighing function to determine the importance of interaction between the student n and student $n - 1$.

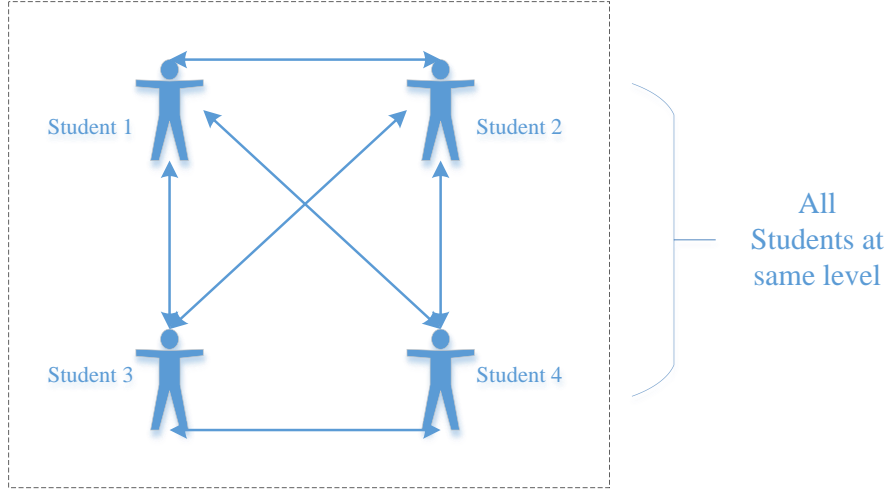


Figure 36: Representation of Student based Flipped Learning Environment

While using Equation 9, a special case is to be considered where, using the weighing function while measuring interactions among two given students there always exists a special case of the weighing function being less than one based on the entropy calculated from students' knowledge. If such a case exists, the weighing function will be equal to one because there will be no positive impact on the interaction. Mathematically it can be represented as:

$$iff \left(\frac{TES_n}{TES_{n-1}} \right) \leq 1 \rightarrow 1 \text{ \& \textit{if} } \left(\frac{TES_n}{TES_{n-1}} \right) > 1 \rightarrow \frac{TES_n}{TES_{n-1}}$$

Entropy based Classroom Structural Assessment Framework

The assessment models developed for the traditional classroom setting, Peer based learning environment and Flipped classroom are motivated by the different possible interaction patterns observed. In this section generalized Entropy based Classroom Structural Assessment Framework (ESAF) is proposed which is based on stakeholder's knowledge and interactions associated in a classroom. This entropy-based framework underlies a stakeholder centric probabilistic model for assessing the state and structure of a complex classroom structure. To

reflect the learning of the students, it is suggested to gather the text recording of the students involved along with the knowledge base used by the class instructor. The interaction component of the framework captures and reflects the features of what a student does in a classroom. The probability component of the framework specifies the rules based on which evidence is gathered for assessment, the weighing component of the framework helps to capture the importance of a considered interaction being evaluated based on the knowledge of stakeholders involved in the interaction. To establish a common platform for assessment, a generalized formulation is suggested in the framework. Illustrated in Figure 37 is ESAF.

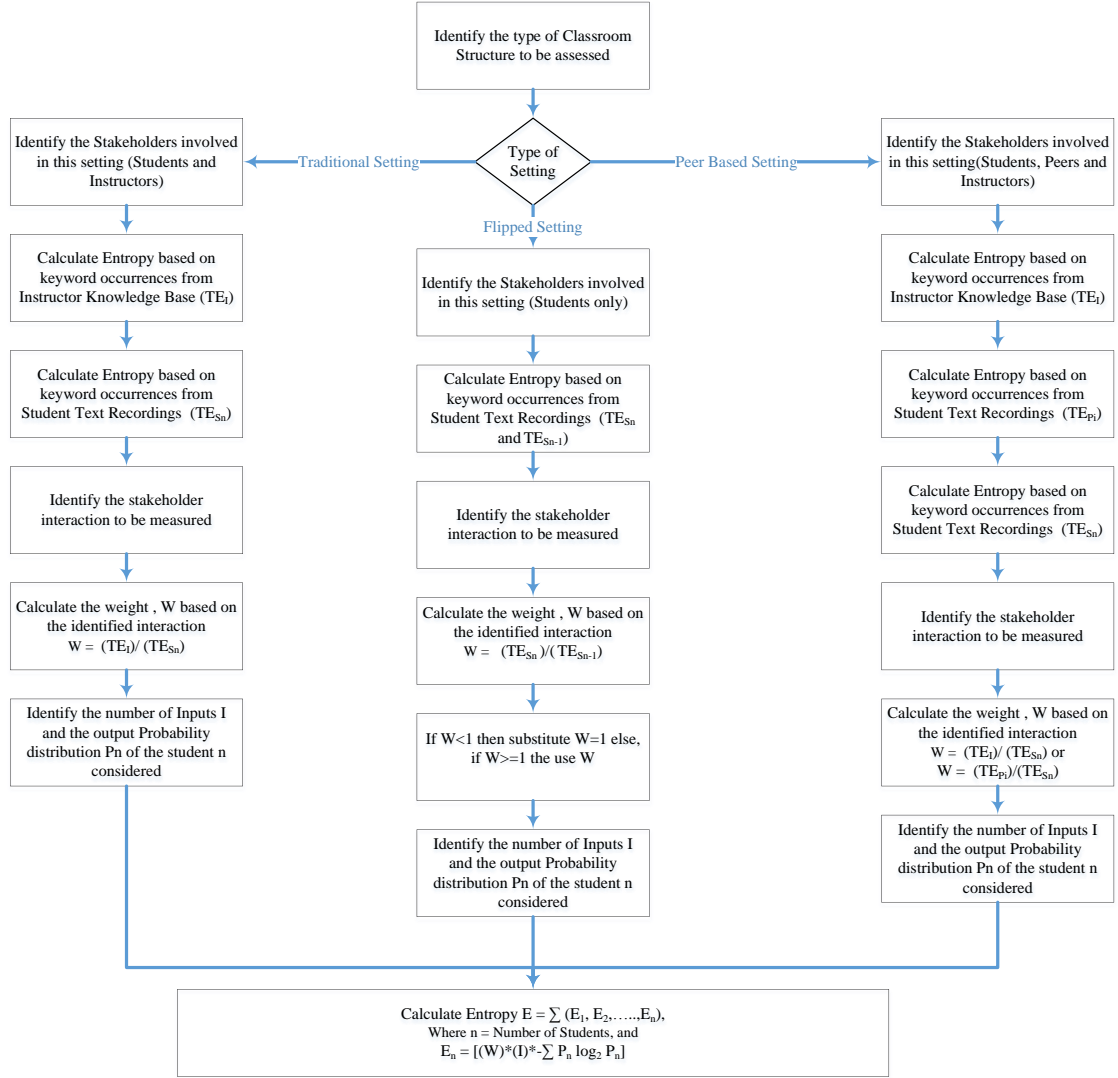


Figure37: Entropy based Classroom Structural Assessment Framework (ESAF)

Chapter 5 – Information Theory Applied Towards Complex Networks

Introduction to Complex Networks

A Network, as derived from graph theory, is a simplest form representing a collection of points that are joined together in pairs by lines. In every such representation the set of points are referred to as nodes or vertices and the lines used to connect the points together are referred as edges. Networks can be mapped from contexts across several domains such as physical sciences, biological sciences, and Social sciences that consist of several components linked together. It is observed in literature that physical, biological and social systems when mapped as networks often provide new insights to the behavior and structure of the system in question [Newman, 2010]. For simplicity, from now on we refer to the vertices of a network as nodes and the lines joining the nodes as edges. Illustrated in figure 38 is a simple network consisting of 7 nodes and 11 edges.

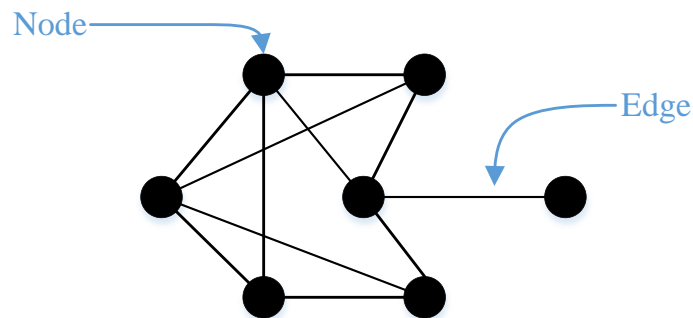


Figure38: A simple network consisting of 7 nodes and 11 edges

Several simple interaction patterns among the nodes of a given network when mapped provide an abstract representation of a given system without which one cannot fully understand on how the system works. These interaction patterns observed in a network emerge into a structure of the

system in question and thereby having an effect on its behavior [Newman, 2010]. For example: the interaction patterns observed among computers connected to internet on a network help to understand the efficiency of data transfer based on the route in which the data is transferred, In social networks the interaction patterns help to understand on how a given set of people interact and spread information.

The question on when a considered network is said to be complex can be addressed using the principles of complexity science.

Several contrasting views on complex and complicated systems have been defined throughout literature. Snowden and Boon (2007) suggesting a framework for decision making from a complexity science perspective address the characteristics of simple, complicated, complex and chaotic systems. The cynefin framework proposed here takes into consideration different operational scenarios wherein, simple and complicated system scenarios have at least one right answer whereas, complex and chaotic system scenarios with cause and effect relationship leading to emergence and thereby have no right answer.

Alderson and Doyle (2010) draw upon contrasting views of complexity by differentiating the basic constructs of simplicity, organized complexity, and disorganized complexity where, simplicity to models such as simple theorems and experiments which have a unique answer. Organized complexity refers to organization of architectures with specific protocols rather than being just random and disorganized complexity involves larger size and number of entities. Please refer to introduction chapter for more explanation and examples of organized and disorganized complexity.

Poli (2013) distinguishes the differences of complicated and complex systems states that complicated systems can be structurally decomposed by sub dividing the system structural parts

and implicitly identifying their relations. Author also identifies that complicated systems can be accurately modeled however, complex systems cannot be fully captured by models thereby having no specific blue prints to address a given problem.

Smith (2013) defines complexity in terms of mathematical formalizations perceived through Random numbers, Transcendental numbers and Imaginary numbers. A system is said to be complex requires characterization by these three types of numbers [Smith, 2013].

Rouse (2015) along the same guidelines provides a comprehensive view of complicated and complex systems. In his view, complicated systems are often engineered based on a pre-defined architecture where though humans play a role in these systems, they typically follow a prescribed set of guidelines. Complex systems on the other hand usually emerge from its constituents without any pre specified set of rules. Human biology is one such example.

Complex systems in nature are non-linear, constituent of individual agents, heterogeneous, self-organize and have no single point of control. Please see introduction chapter for more detailed explanation on complex systems and their characteristics. Drawing upon the above views and characteristics of complicated and complex systems and, looking back into networks, we define a complex network to be a network mapped from a system that primarily constitutes of interactions observed among systems components, interactions of human actors in the system, and the influence of human interaction on system components. Continual human interactions in such networks play a primary role in influencing the network structure and its evolution. The inherent ambiguity of human interactions and their individual perceptions introduce non-linear behavior, emergence and self-organization in networks thereby introducing complexity.

Table 18 illustrates the distinguishing characteristics of complicated and complex networks.

Table 18: Characteristics of complicated and complex networks

	Complicated Networks	Complex Networks
Nodes	Structured and predefined set of connections	Entity of Random node connections
Characteristic	Highly tuned	Minimally tuned
Topology	Highly organized structure	Continually emergent structure based on dynamics of interactions
Robustness	Robust to targeted perturbations	Robust to random perturbations
Relationships	Easier to understand network relationships	Difficult to understand network relationships
Human influence	Human interactions based on a predefined set of rules	Random human interactions
Network Examples	Software system, automotive system, electronic circuit networks	Transportation, Social, Communication & Biological Networks

Summarized in figure 39 is the representation of a complex network. This observation illustrates that the complexity in a network is a result of continual relationship between an actor (human) and the components (sub-systems) of a system. In other words complexity in a network depends upon the extent of interactions a human actor has in a system and the cascading influence human interactions initiate throughout leading to network emergence.

Complex networks to some extent can be characterized by the interactions among its constituent nodes. These interactions evolving together result in the dynamic nature of the network. Many complex networks have a number of properties that are common in nature, leaving aside their domain specific characteristics. Displaying small world phenomenon is one such characteristic and the other being highly heterogeneous nature of interaction patterns in many cases [Sole and Valverde, 2004]. Heterogeneity in complex networks can be characterized by observing the degree distribution based on the node interaction dynamics. Many networks observed are reported to be Scale free networks, the networks where the degree distributions are observed to be following a power law.

Complex Network = f (System component interactions, Human interactions, Human & system component interactions)

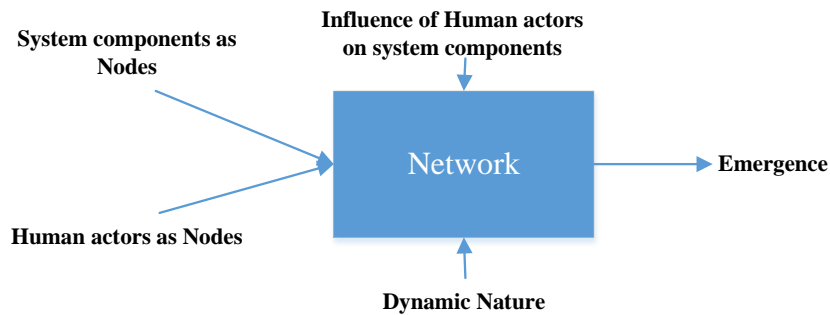


Figure39: Abstract fundamental representation of Complex network

Few examples of such networks are electronic circuits [Sole and Valverde, 2004], Cellular Metabolism networks, Research Collaboration Networks, Connectivity in World Wide Web [Barabasi, 2009]. Illustrated in Figures 40 and 41 are scale free network generated based on Barabasi-scale free network mechanism [Wilensky, 1999, 2005] along with its captured degree distribution portraying a long tail (power-law) distribution.

Looking into the dynamics of node interactions dynamic degree distribution is given by probability P_{k_i} where, it gives the probability of a particular node having k links at a given time-step i . This concept of dynamic degree distribution will be use later in this document to understand and get a better insight towards evolution in complex/social networks. The dynamics of network interaction patterns though they differ from a system to the other, relate to the propagation of information in the networks [Sole and Valverde, 2004].

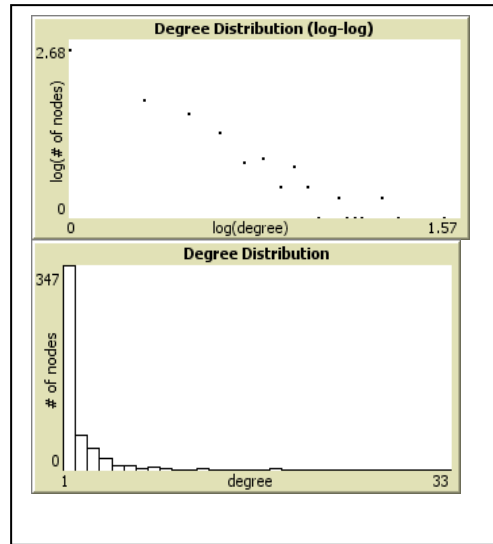
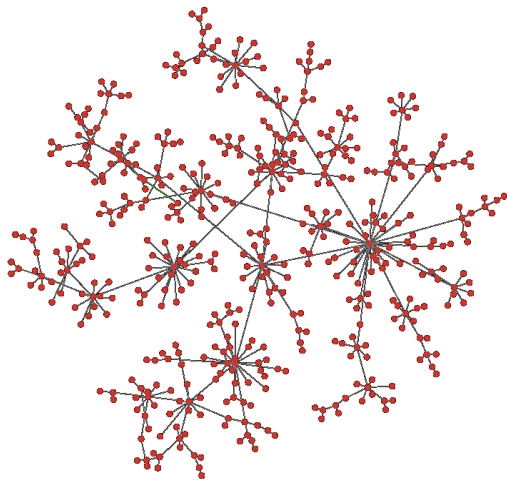


Figure 40: Scale Free Network of 500 nodes and its corresponding degree distribution

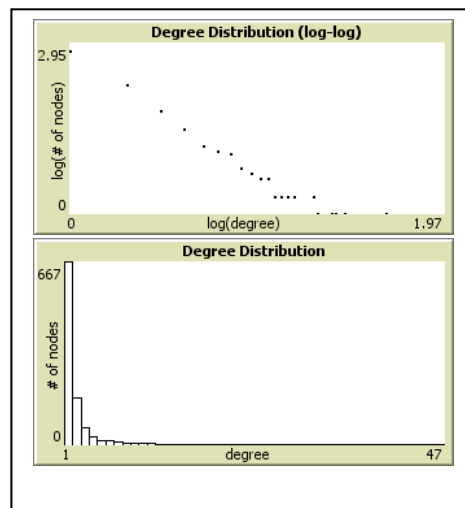
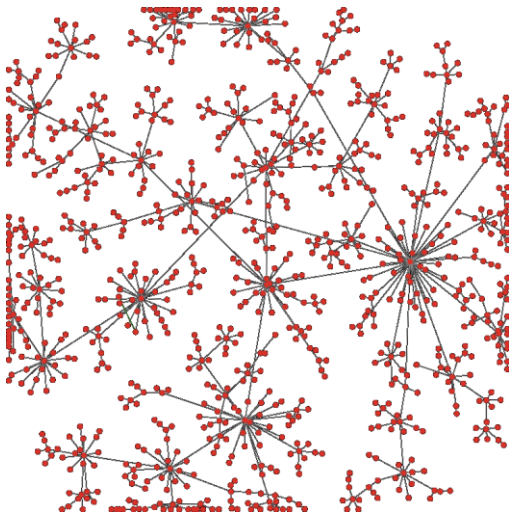


Figure 41: Scale Free Network of 1000 nodes and its corresponding degree distribution

Social networks are a special case of complex networks where in the structure of a social network is made up of a set of actors (individuals) acting as nodes and the interactions observed between them. These interactions represent the ties, relationships, and flows between groups of individuals or any other information processing units. Social networks are dynamic in nature as they evolve based on the interaction patterns observed among the individuals involved over a period of time. They are also inherently Emergent, Self-Organizing, and Complex in nature

[Newman et al, 2006] [Mehra and Freeman, 2005]. Figure 39 Illustrates a social network of e-mail communication of 436 employees from Hewlett Packard build based on traces of online communication.

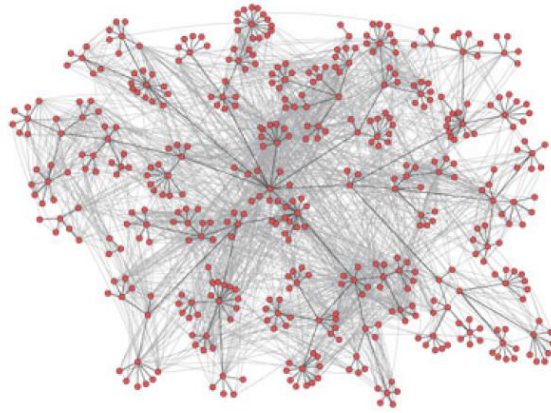


Figure 42: Social Network of online email communication among HP employees [Huberman and Adamic, 2004]

Figure 43 illustrates an example of a social network extracted from Facebook corresponding to two different individuals geographically separated. Nodes in figures 43a and 43b represent Facebook friends of the particular individuals from whom the network is extracted.

The edges connecting the nodes represent a friendship relation where if a line connects any two nodes it implies that the two particular nodes who are friend of the particular individual are also friends with each other on Facebook. It is observed that the degree distributions in both examples of social networks portray a long tail (power-law) behavior. To further provide insight into social networks and their use, illustrated in Figure 44 are the social networks mapped from official Facebook page of current United States presidential candidates Donald Trump and Hilary Clinton. This example illustrates on how one can understand the affiliations of an individual based on their social networks.

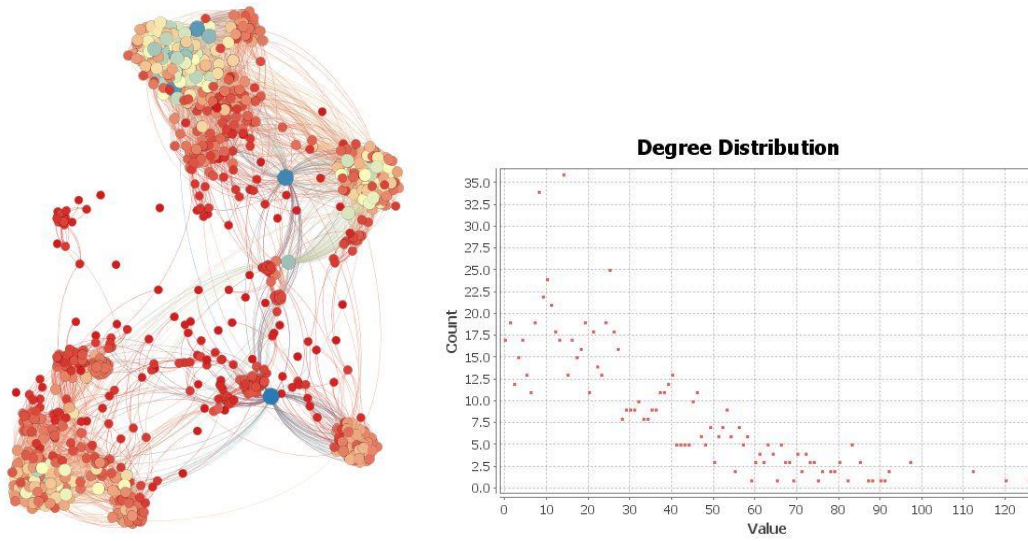


Figure 43a: Facebook Network with 826 Nodes, 11361 Edges, & Average degree of 27.508 per node

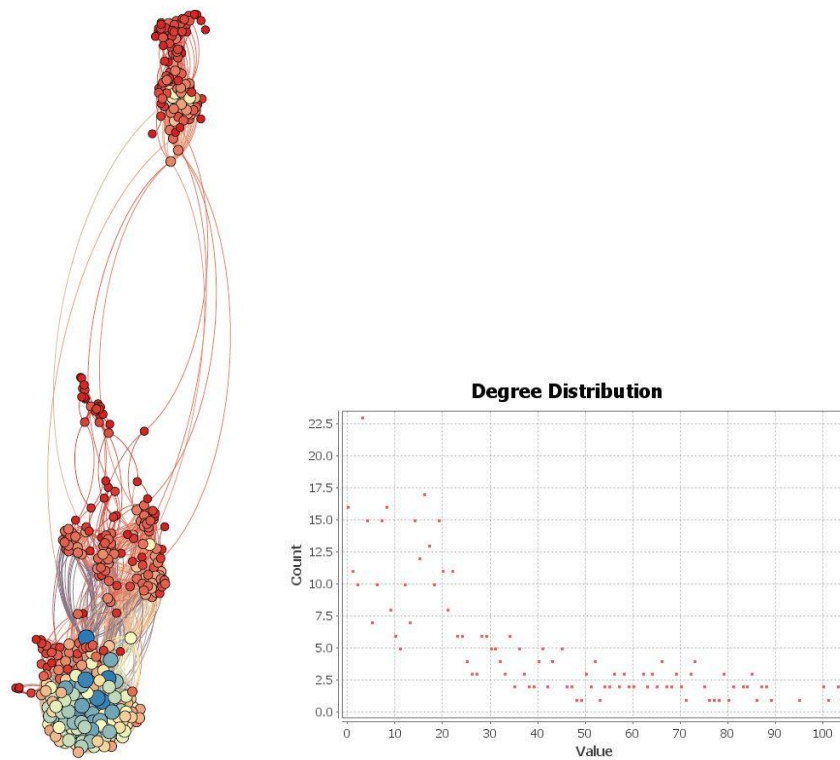


Figure 43b: Facebook Network with 456 Nodes, 6163 Edges, and Average degree of 27.031 per node

The nodes in figure 44 illustrate the Facebook pages that liked the Facebook page of the particular individual considered. The connection among the nodes indicates that the particular node which represents a Facebook page liked another node representing a different Facebook page. The nodes of the networks are clustered into different sets and color coded to understand their affiliations with respect to the Facebook page and the size of the nodes of the network are proportional to their individual degrees.

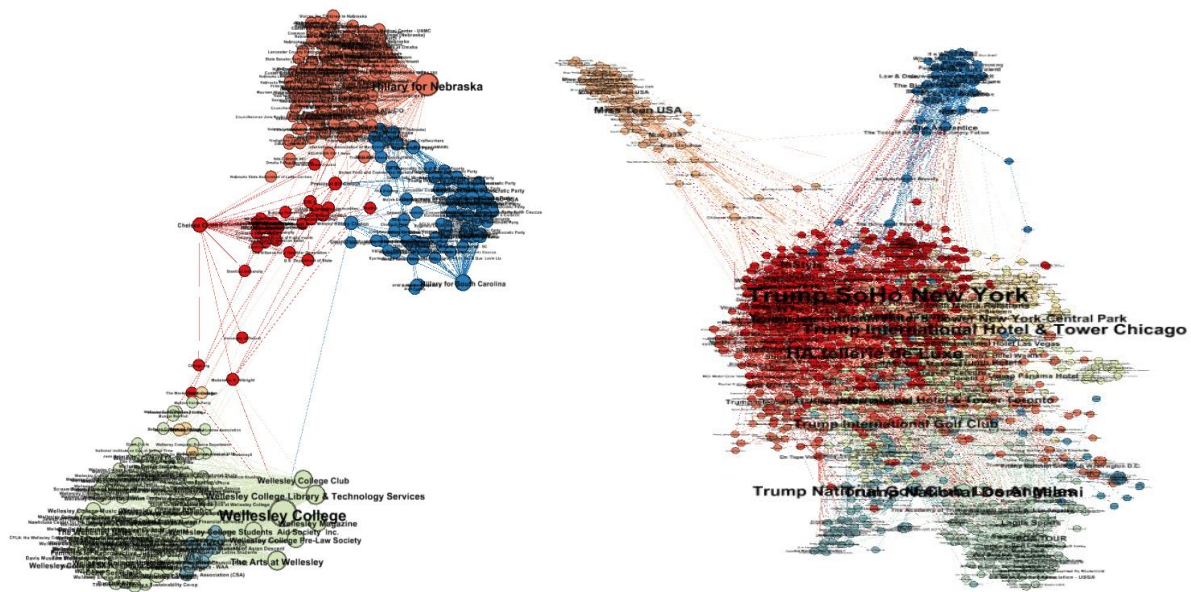


Figure 44: Social Networks extracted from official Facebook pages of Hilary Clinton (left most in the figure) and Donald Trump (right most in the figure).

It is seen that the network extracted from Facebook page of Hilary Clinton is more connected with Nodes affiliated to Wesley College which is her Alma matter, several non-Profit organizations, U.S. Department of state, and pages related to her election campaigns in Nebraska and South Carolina. Looking into the network extracted from the Facebook page of Donald Trump it is seen that his Facebook page is more affiliated with Trump SoHo New York, Trump Hotels, and Trump entertainment, Fashion Chains, TV Shows and Beauty Pageants.

To analyze social networks such as illustrated in figures 42, 43 and 44, the field of Social network analysis (SNA) pulls in together a multidisciplinary approach to map and measure the relationships between people, groups, entities, organizations and other similar connect information and knowledge generating entities. Using SNA statistical tools one can evaluate and understand a given network in identifying network leaders, bridges, clusters and similar attributes. Mentioned here are a few measure that are widely used to understand social networks [Scott, 2012].

- Degree Centrality: This measure captures the connectedness of a network based on the number of connections (also called as ties, degrees, or the links indecent upon the node) of a node [Freeman, 1978]. Higher degree centrality of a node indicates that the particular node has more opportunity and choices in the network making it powerful. This indicates that the nodes/actors with more ties often act as deal makers, have a control on exchange of information among others. Considering the directionality in network connections, the actors with high in-degree centrality are more prominent in network as they receive more information in the network and on the other hand the actor/node with high out-degree centrality has more influence on the whole network [Hanneman, 2005].
- Closeness Centrality: Degree centrality takes into account the number of connections a node as. However, sometimes there could be possibility that a node with several ties be disconnected from other actors in a network as a whole. Taking this into consideration, Closeness centrality is calculated based on the distance of a given node to all the nodes in the network and thereby considering the whole network [Hanneman, 2005]. Lower closeness centrality value of a node implies that it is more centrally connected in the network.

- **Betweenness Centrality:** Betweenness centrality measure helps to understand if a given node lies in the geodesic path of other node in a network. This measure helps to identify the node which acts as a bridge in a given network with a potential of control over the flow of information [Scott, 2012]. Nodes with high betweenness centrality are considered to be more powerful in the network [Hanneman, 2005].
- **Eigen Vector Centrality:** Based on the concept of degree centrality measure, Eigen vector centrality of a node considers if the particular node is connected to other important nodes in the network. Primarily different from in degree centrality, a node with high Eigen vector centrality may not be necessarily linked to many nodes in the network but the few number of connection it may have are with important nodes. In other words, this measure helps to identify the nodes that are will connected with other well connected nodes [Scott, 2012][Ruhnau, 2000].

Social networks are steadily growing in complexity as a result of social ties individuals establish with the help of technological platform advances that facilitate in seamless interaction among the stakeholders. Over the past decade it is observed that social networks are rapidly increasing and exploiting the geographical boundaries [Easley and Kleinberg, 2012]. *The amount information utilized, generated and transmitted in such networks is also growing in complexity incorporating information content from individuals, academia, and organizations from their own perspectives. Understanding such information on how it cascades over an emerging network and how it impacts the structural aspect of a dynamic network will give a better insight on network evolution/emergence dynamics. This would help in laying the foundation for answering the central question in complexity science on “How large networks with simple components, limited*

connections among the components and no central control give rise to a complex self-organizing behavior?”[Mitchell, 2006].

Information Theory in Complex Networks

The fundamental application of incorporating information theory concepts has been widely observed in the field of statistics, computer science, physics, biology, and electrical engineering. To understand the information generated in complex networks we use the concepts of information theory, a generalized statistical approach widely used to quantify information and communication of information. In information theory, Entropy is a key concept that helps to quantify the expected value of information contained in a message. Please see Chapter 1 of this document for a detailed overview on the background and the theory surrounding the concept of entropy. Outlined below are on how the key concepts of information theory can be defined for dynamic complex networks to understand the flow of information and its implications on network dynamic structural emergence.

Entropy to understand Interaction Dynamics

To understand the dynamic structure of a given complex network as a whole, it is important to analyze and explore the main conceptual underpinning of a network which is nodes and their interactions. Complex networks being continually emergent and non-linear in nature, exploring the dynamics of constituent nodes and their interaction patterns will help in providing insight on how a network structure evolves.

Considering the degree distribution of every individual node in a network, let P_{k_i} be the dynamic degree of a node where, it gives the probability of a particular node having k links at a

given time-step i . We first define an entropy measure for a static network which will then be expanded to consider the dynamic nature of complex networks.

Suppose there are ' n ' nodes in a network space ' N '.

$$N = (N_1, N_2, \dots, N_n)$$

Now let ' K ' be a class of ' N ' where it defines the number of links of each observed node ' n ' of a network.

$$K = (K_1, K_2, \dots, K_n)$$

Let ' I ' be the proportion of interactions that are represented by the number of links associated to each node ' n '.

$$I_n = \frac{K_n}{(K_1 + K_2 + \dots + K_n)} = \frac{K_n}{\sum_{i=1}^n K_i}$$

$$\text{Where, } \sum_{i=1}^n K_i = 1$$

Entropy is mathematically represented as

$$H(X) = - \sum_{i=1}^m p(x) \log_2 p(x)$$

Where, $0 < H(X) < \log_2 m$

Applying this measure to the number of nodes constituent in the network space we get,

$$H(X) = - \sum_{i=1}^n I_n \log_2 I_n$$

Now, considering the dynamic nature of a complex network, let ' j ' be the number of time-steps over which a network is observed. At every time-step ' j ' each and every node ' n ' of the network will have a degree ' k '. Now considering the degree of nodes at individual time-steps of a network, entropy is given by

$$H(X) = \sum_{t=1}^j \left(- \sum_{i=1}^n I_{n_t} \log_2 I_{n_t} \right)$$

Here we quantify a complex network to be a function of the number of nodes present (N_n) and the number of links (I_n) associated to the nodes. The uncertainty based on the distribution of the node degrees over a time period is used as a base indicator to calculate network entropy. This also takes into consideration the diversity in network interactions.

This index when calculated represents absolute order in the network when it takes a value of 0 and, when it represent disorder and absolute diversity it takes a maximum value (the maximum value depends upon the type of complex network being considered).

Transfer Entropy to understand the transfer of information in network over time

To further extend Entropy towards measuring uncertainty between 2 random variables X and Y, a measure of mutual information is used. Mutual Information is based only on the present state/symbol of each variable, considering all such present states, mutual information, $I(X, Y)$ is defined by

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{P(x, y)}{P(x) \cdot P(y)}$$

However, the fact that the measure of mutual information is symmetrical, i.e. $I(X, Y) = I(Y, X)$, implies that the future state of a random variable has a casual effect on the past state. In the case of complex networks, where a network consisting of several nodes evolves over a period of time, the structure of a network at a time 't' depends upon the evolution of the network until time ' $t-1$ ', not the other way around. Thus, in order to address the symmetrical limitation of mutual information measure, the concept of transfer entropy was proposed [Murcio et al., 2015].

Transfer Entropy (TE) was given by Schreiber to address the time symmetric limitation of a mutual information measure. Considering two sample spaces of information represented in time by $X = \{x_1, x_2, x_3, \dots, x_t\}$ and $Y = \{y_1, y_2, y_3, \dots, y_t\}$, transfer entropy is defined as the additional amount of information gained for the next observation of one of the two processes being considered, given the past observation of the other process.

Following Murcio et al., considering 2 systems X and Y, we first define the entropy rate (i.e. entropy based on time t) assuming that y_{t+1} depends upon both x_t and y_t as:

$$H_A = - \sum_t p(y_{t+1}, y_t, x_t) \log(p(y_{t+1}/y_t, x_t))$$

We now define entropy rate in which y_{t+1} depends only on y_t :

$$H_B = - \sum_t p(y_{t+1}, y_t, x_t) \log(p(y_{t+1}/y_t))$$

Hence, the transfer of information from X to Y is defined as $T(X, Y)$ i.e. Transfer of information from X to Y: $TE(X, Y) = H_B - H_A$.

TE, the transfer of information between two random variables X and Y is given by [Murcio et al., 2015] [Schreiber, 2000]:

$$TE(X, Y) = \sum_{t=1} p(y_{t+1}, y_t, x_t) \log \frac{p(y_{t+1}, y_t, x_t) \cdot p(y_t)}{p(y_t, x_t) \cdot p(y_{t+1}, y_t)}$$

Also, the transfer entropy from Y to X can be inferred similarly based on the above as:

$$TE(Y, X) = \sum_{t=1} p(x_{t+1}, x_t, y_t) \log \frac{p(x_{t+1}, x_t, y_t) \cdot p(x_t)}{p(x_t, y_t) \cdot p(x_{t+1}, x_t)}$$

To analyze the flow of information generated from one node to the other of a considered network based on the observed individual node degree evolution over several time-steps of network formation, the concept of transfer entropy is used. TE when applied to a network evolution

scenario helps in understanding how much of information generated at one node is responsible for the information obtained by the other node.

Network Models Used to Apply Metrics

To Calculate Entropy and TE values based on individual node degree evolution, data was generated using Facebook and NetLogo agent based simulation platforms. A student based interaction network and three random networks were generated, and the data was captured individually for each network based on the number of nodes present along with the individual node degrees over several time-steps. The characteristics of the networks generated along with the calculations of transfer entropy values are discussed below.

Agent Based Modeling as a Platform to Generate Networks

Agent based modeling (ABM) helps to develop models that enable is charactering real systems. ABM helps to model and represent individual components of a system considering all the possible attributes and their behavior instead of representing the state of a whole system. Tracing back to the principles of complexity science, a system is a sum of its parts as a whole but not the other way round. ABM models are used to generate network models where individual agents (in this case network nodes) are describes as individual entities that interact with each other locally in their environment [Railsback and Grimm, 2011]. The main focus is to model network emergence i.e. the dynamics of the system that arises when individual nodes interact with each other and on how such individual nodes affect the overall network.

Preferential Attachment Network

Generated networks using Netlogo software illustrate the behavior of real world networks (such as connection from and to a website, social networks, collaboration networks etc.) where, few nodes have a lot of connections while all the other have only a few. This phenomenon where network nodes prefer to connect to the popular ones between the existing is called preferential attachment. This model starts with 2 nodes connected at first and thereby every new node originating randomly picks a current existing node to connect with some inherent bias, i.e. the chance of a node being chosen to be connected with to the other is directly proportional to the number of connections (degree) it already has.

The networks that arise from this phenomenon often follow power law distribution i.e. the distribution of the number of connections of each node is not normal [Albert and Barabási, 1999]. Barabasi and Albert originally coined this mechanism for creating scale free networks and so the networks created by this mechanism are called Barabasi scale-free networks. Figures 45 illustrate the evolution of interactions in the network over fifteen different snapshots.

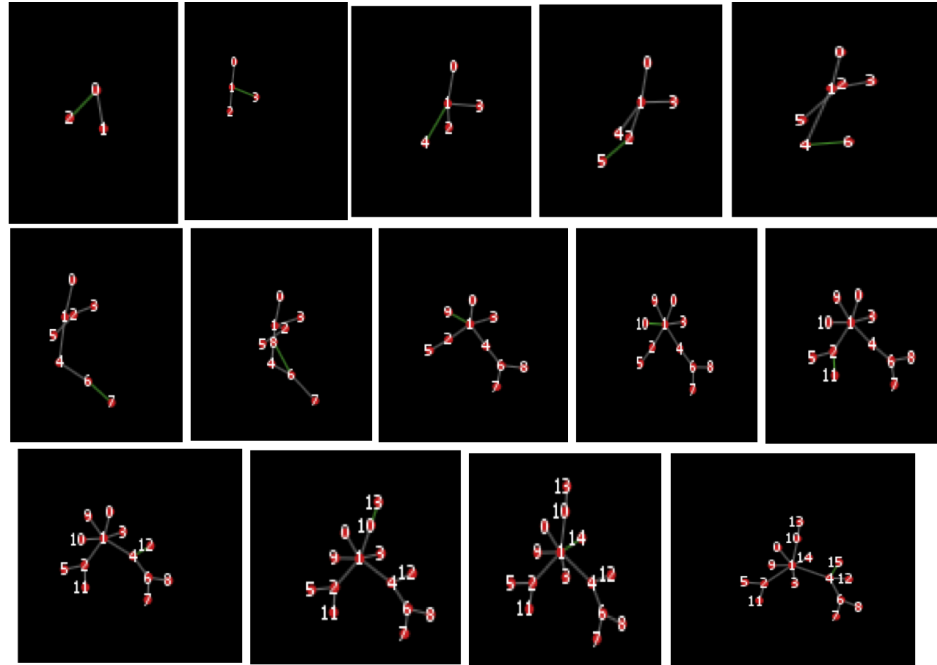


Figure 45: Emergence of Network based on Preferential Attachment Phenomenon

To analyze the flow of information generated from one node to the other of the considered Preferential Attachment network based on the observed individual node degree evolution over several time-steps of network formation, the concept of transfer entropy (TE) is used. To generate TE values based on individual node degree evolution, data on the number of nodes present along with the individual node degrees over 15 different time-steps were captured using NetLogo software. Illustrated in Table 19 is the degree evolution of nodes based on the Figure 42.

Table 19: Time series showing the node degrees of the nodes generated at each time-step.

Time-steps Node↓	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	3	4	4	4	4	4	5	6	6	6	6	7	7
2	0	1	1	1	2	2	2	2	2	1	1	1	2	2	2
3	0	0	1	1	1	1	1	1	1	2	1	1	1	1	1
4	0	0	0	1	1	2	2	2	2	1	3	1	1	1	1
5	0	0	0	0	1	1	1	1	1	2	1	3	1	1	1
6	0	0	0	0	0	1	2	3	3	1	2	1	3	1	1
7	0	0	0	0	0	0	1	1	1	3	1	3	1	3	1
8	0	0	0	0	0	0	0	1	1	1	3	1	3	1	3
9	0	0	0	0	0	0	0	0	1	1	1	3	1	3	1
10	0	0	0	0	0	0	0	0	0	1	1	1	3	1	4
11	0	0	0	0	0	0	0	0	0	0	1	1	1	3	1
12	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3
13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 20: TE values calculated based on node degrees from table 2

Source Node	Destination Node	Transfer Entropy (S-D)	Transfer Entropy (D-S)
0	1	0	0.1177471
1	2	0.1557854	0.077615
2	3	0.05859509	0.0326634
3	4	0.1435097	0.03558652
4	5	0.1850155	0.1041592
5	6	0.1661563	0.08439188
6	7	0.1142876	0.05446116
7	8	0.1934432	0.06450643
8	9	0.1820437	0.06138331
9	10	0.1833988	0.07069286
10	11	0.1868527	0.08600857
11	12	0.1631874	0.04300429
12	13	0.1319313	0.0162317
13	14	0.1093635	0.003653752
14	15	0.1117521	0.02150214

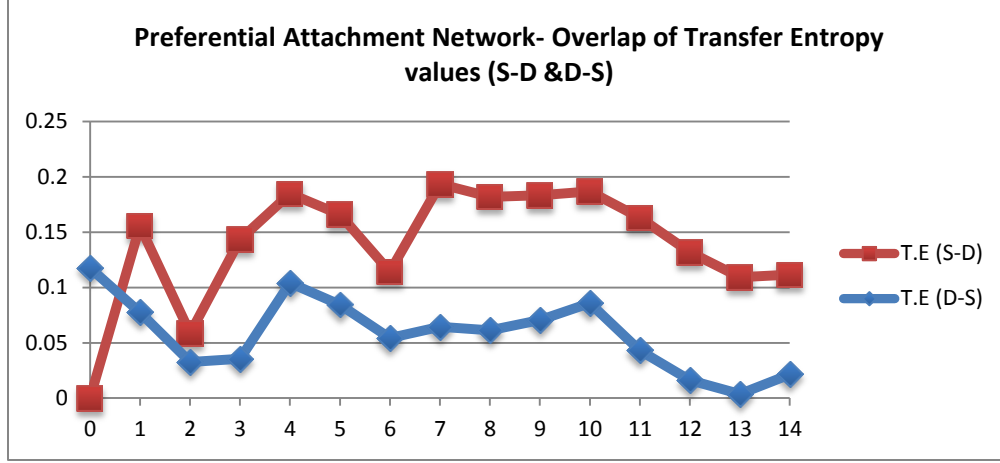


Figure 46: Transfer Entropy values between the nodes of Preferential Attachment Network

To calculate the TE values obtained in table 20, joint probabilities are calculated for emerging node degrees of table 2. Thus, for us to calculate the TE from node 0 to node 1 at time-step t1, we need to find the probability $p(y_2, y_1, x_1)$ as required for calculating TE. Where, y_2 corresponds to the value located in second column second row (c4); y_1 corresponds to the value in first column second row (b4); and x_1 corresponds to the value in first column first row (b3); i.e. $p(1,1,1)$. We now count the number of matching combinations of these values that exist in the two rows corresponding to node 0 and node 1. In the graph (Figure 43), the X-axis represents the node numbers and Y-axis represents the corresponding TE values. It is read as: for the red line, Mark 0 is the TE value from Node 0 to Node 1, Mark 1 is the TE value from Node 1 to Node 2 and so on. For the blue line, Mark 0 is the TE value from Node 1 to Node 0, Mark 1 is the TE value from Node 2 to Node 1 and so on.

It is seen that in general (from Figure 46); information flow from the source nodes to destination nodes initially rises and then dominates information flow from destination nodes to source nodes. At the early stages of the graph, the trend line of S-D, when observed; it is seen that TE between node 0 and 1 is zero with node 0 as the source and node 1 as the destination. This simply suggests the initial network structural formation where, at the initial time-step nodes 0 and 1 are

connected on which the simulated network builds upon over the considered time series. Before calculating the network entropy measure, we introduce a measure of connectedness also called as reproductive number in complex networks.

Reproductive number R_0 helps to understand the context of information spread in a network. R_0 also known as connectedness of a network can be calculated by

Consider a node ' n ' with degree ' k ' to communicate an idea or spread information to its neighbors with a probability ' r '. The expected number of nodes it will pass on the information to will be $r * (k - 1)$ by excluding the nodes it already communicated the information to. Taking a weighted average over all the nodes we get:

$$R_0 = r * \frac{\sum_{i=1}^n K_i(K_i - 1)}{\sum_{i=1}^n K_i}$$

If the value of R_0 is greater than the value of 1, it implies that the number of nodes receiving the information grows exponentially and if R_0 is less than 1 it implies that the information will dissipate rapidly over a network.

Now, to understand the structural interaction dynamics data from Table 19 is used to calculate the network entropy at each time-step. Along with calculating the dynamic network entropy, Reproductive number, a complex network structural measure is also calculated to correlate with the entropy measure. Table 21 shows the entropy measure and R_0 calculated at every time-step of the dynamic network.

Table 21: Dynamic Entropy and R_0 values for generated preferential attachment network model

Time-step	Dynamic Entropy	R_0
0	1	0
1	1.5	0.083333
2	1.792481	0.083333
3	2	0.075
4	2.321928	0.046667
5	2.584963	0.031746
6	2.807355	0.022959
7	2.95282	0.019097
8	3.038562	0.018519
9	3.108695	0.018182
10	3.23127	0.015152
11	3.344361	0.012821
12	3.478346	0.010566
13	3.524635	0.010544
14	3.601515	0.009444

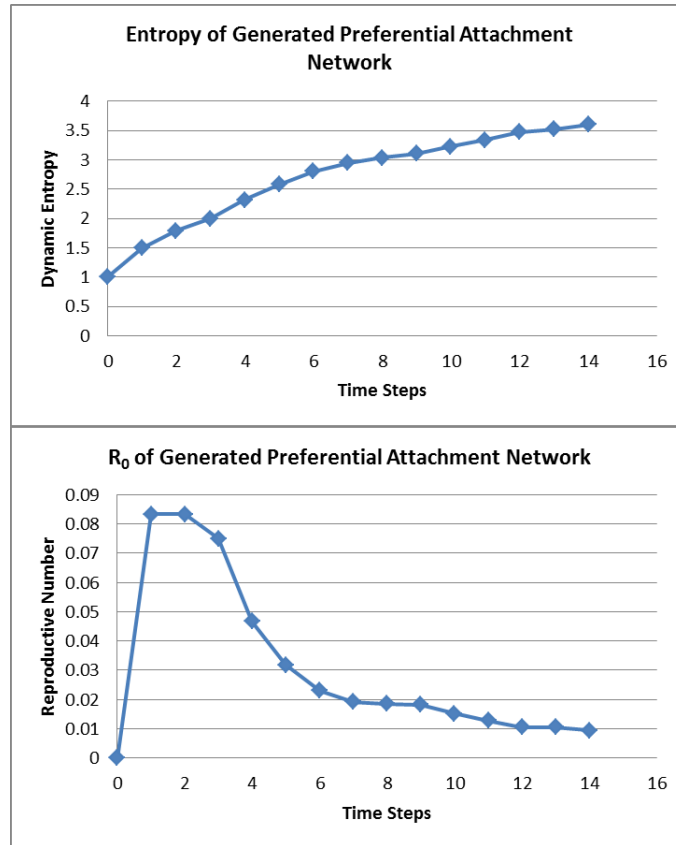


Figure 47: Dynamic Entropy and R_0 Measures of Preferential Attachment Network Generated

It is observed that as the network evolves, entropy increases implying that the preferential attachment network with more nodes evolving over a period of time the network as a whole tries to reach a disordered ordered/more diverse state from an ordered state initially. As entropy (diversity) increases the reproduction number initially increases and then decreases. This implies that when the network starts to evolve, there is a more chance of information spread, and when the networks starts evolving to more than 4 nodes at time-step 3, the information spread slowly decreases.

Small World Networks

The network generated here using Netlogo simulation software is based the phenomenon of small word networks. This phenomenon implies that a given person is only a couple (few) connection away of any other person in the world.

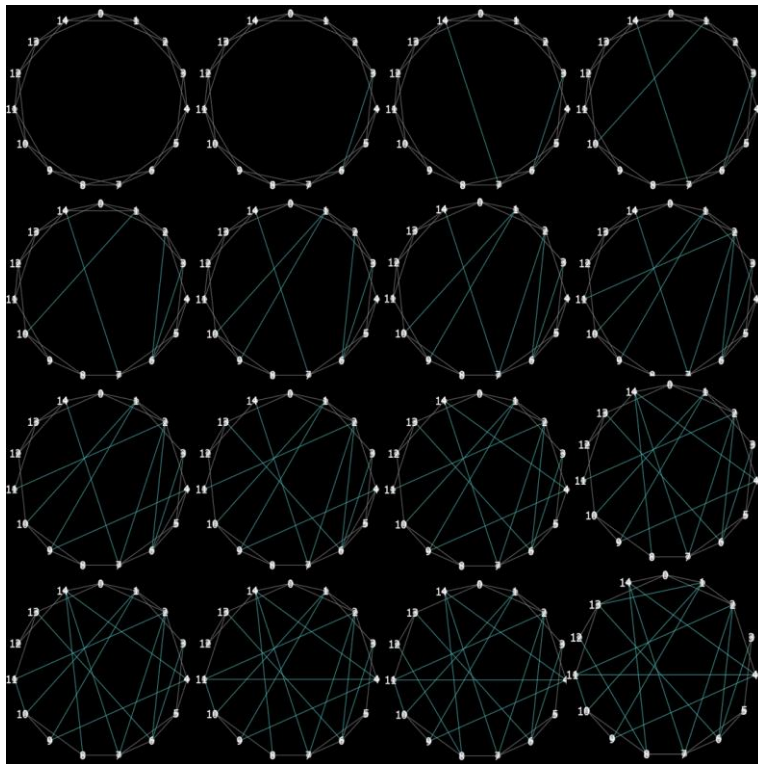


Figure 48: Small world Network Emergence generated using Netlogo software

A popular example of this phenomenon is the famous Kevin Bacon network (also known as a six degree separation network) where this is based on a network generated based on actors appearing in a same movie. However, small world networks are not only limited to networks of people but also apply to several other real time networks such as power grids.

The network model generated here is based on a few assumptions and conditions under which a small world network is formed. It is developed based on the model suggested by Duncan Watts and Steve Strogatz [1998]. The model starts by initially generating a network where each node is connected to its two neighbors on either of its sides. After this initial random network is generated, at each and every time-step a random connection is picked and then rewired i.e. a random end of a connected pair of nodes is changed. The probability that a random edge is chosen and rewired is based on the rewiring probability value assigned to the network generation model. Figures 48 illustrate the evolution of interactions in the network over fifteen different snapshots. Tables 22 and 23 illustrate the node degree evolution observed for the generated small world network and their respective transfer entropy values.

Table 22: Time series showing the node degrees generated at each time-step of Small world network

Time-steps Node↓	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5
2	4	4	4	4	5	5	5	6	6	6	6	6	6	6	6
3	4	4	4	4	4	4	3	3	3	3	3	3	3	3	2
4	4	3	3	3	3	3	3	4	4	4	4	4	4	4	4
5	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3
6	4	5	5	5	5	5	5	5	6	5	5	5	5	5	5
7	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5
8	4	4	4	4	3	3	3	3	3	3	3	3	3	4	4
9	4	4	3	3	3	4	4	4	4	4	3	3	3	3	3
10	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
11	4	5	4	3	3	3	3	2	2	2	2	3	4	4	4
12	4	3	4	4	4	4	4	3	3	3	3	2	2	2	2

13	4	4	4	4	4	4	4	4	4	4	4	4	4	4	5
14	4	4	5	5	5	4	4	4	3	4	5	5	5	4	4

Table 23: TE values calculated based on node degrees from table 22

Source Node	Destination Node	Transfer Entropy (S-D)	Transfer Entropy (D-S)
0	1	0	0.000272211
1	2	0.06977687	0.000695039
2	3	0.04530374	0.08823196
3	4	0.2065937	0.01069302
4	5	0.02291414	0.01371527
5	6	0.01057148	0.005931286
6	7	0.006480284	0.01851988
7	8	0.01268821	0.00772787
8	9	0.02148589	0.06094668
9	10	0.00096725	0
10	11	0	0.002274648
11	12	0.08492251	0.1661563
12	13	0.05251608	0.02677259
13	14	0.02899597	0.02269644

As observed in figure 49, Node 0 has no information originating from them. This simply implies to the rigidity of the node in not changing (connected to the same neighbors throughout) the degree over the network emergence, thereby not generating any new information. The necessary aspect of understanding a network in social scenarios is based on how frequently actors interact in a network. These measures when calculated at each time-step for all the nodes in the networks (see appendix A), it is observed that the results on the flow of information at each node using transfer entropy calculations does not necessarily correlate to the results obtained by the calculated centrality measures.

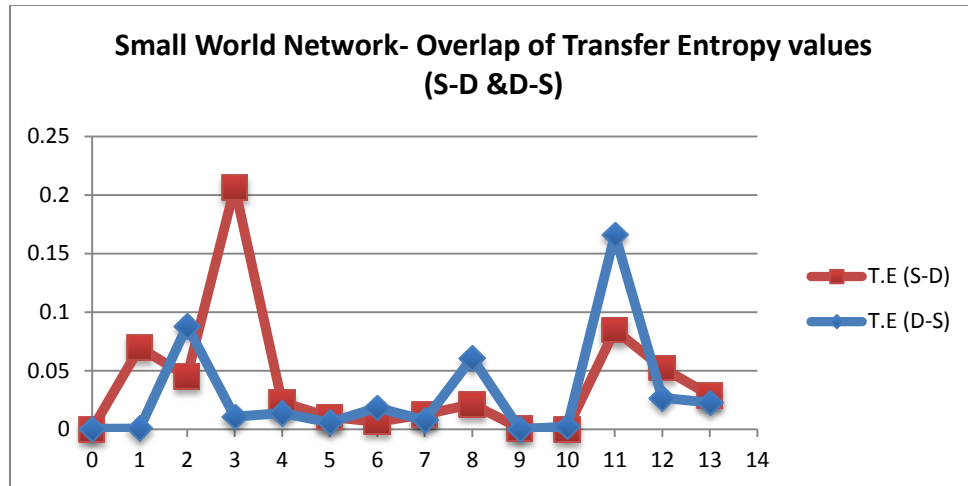


Figure 49: Transfer Entropy values between the nodes of Small world Network

Below mentioned are few key points observed:

- Node 2 has the highest average degree centrality measure indicating that it acts as a local hub of the network, but necessarily not the best connected. On the other hand, node 3 is observed to have comparatively the least average degree centrality.
- Node 2 has the highest average closeness centrality measure indicating it can more easily spread information to the rest of the network along with having a high visibility on the network; possibly influencing node 3 with a high associated information flow based on their tight neighborhood.
- Highest value of betweenness centrality is associated with node 2 implying that it acts as a key bridge with in the network for the flow of information, whereas; the least betweenness centrality value is observed at node 8. This may imply the influence of node 2 as a key bridge on node 3 based on their tight neighborhood.
- Node 2 with the highest Eigenvector centrality value acts as the leader of the network. However, it does not imply that it has strongest local influence.

It is seen that the observations made based on the centrality measures when applied to the network; there exists a vast difference between the transfer entropy measures and centrality measures. This implies to a possibility that the networks in figure 45 are based on generating an algorithm that replicates small world network phenomenon with its nodes rewired randomly based on an assigned value of probability, but not based on real world data. Table 24 shows the entropy measure and R_0 calculated at every time-step of the dynamic network.

Table 24: Dynamic Entropy and R_0 values for generated Small world network model

Time-step	Dynamic Entropy	R_0
0	3.906891	0.014286
1	3.89474	0.014603
2	3.89474	0.014603
3	3.888664	0.014762
4	3.882589	0.014921
5	3.888664	0.014762
6	3.882589	0.014921
7	3.863501	0.015397
8	3.852584	0.015714
9	3.863501	0.015397
10	3.857426	0.015556
11	3.857426	0.015556
12	3.857426	0.015556
13	3.863501	0.015397
14	3.849255	0.015714

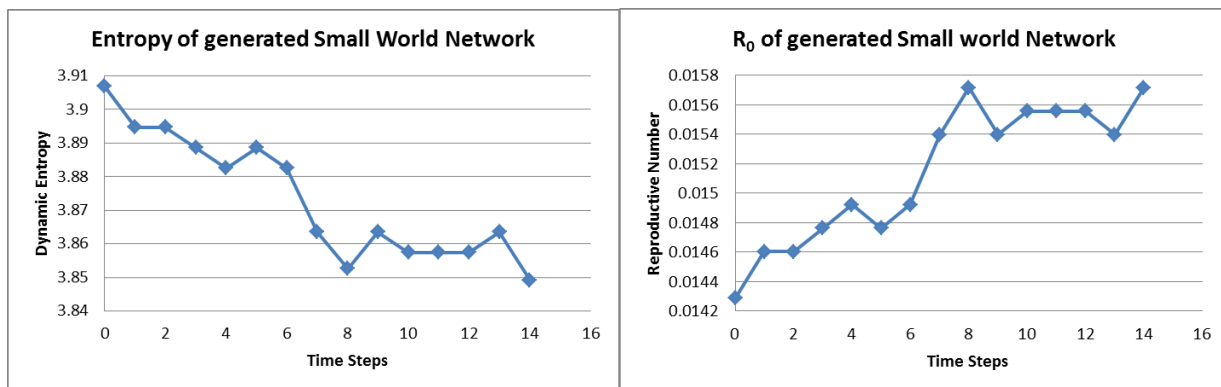


Figure 50: Dynamic Entropy and R_0 Measures of Small World Network Generated

It is observed that as the network evolves, entropy decreases implying that the small world network tries to reach an ordered state from a high disordered state initially. As entropy (diversity) decreases the reproduction we see that number increases. This implies that when this network tends to reach absolute order (i.e. less entropy) the spread of information increases over the network.

Team based Assembly Networks

The networks generated here illustrate on the behavior of individuals on how they give rise to large-scale networks by assembling in small teams for short-term projects. This model is developed based on the team assembly model given by Guimera, Uzzi, Spiro & Amaral [2005], which is based on observing the behavior of various collaboration networks. This model captures almost all the features usually found in networks of enterprises based on the number of new comers to a team along with the possibility of collaborators working together again with each other.

At every time-step of the network evolution, a new team is formed compromising new comers and incumbents. Once a new team is formed, all the participants are linked with each other. However, if a new agent doesn't participate in a new team for a long time, the agent's links are removed from the network. Agents are colored blue if they are newcomers and yellow if they are incumbents. This model helps to explore several scenarios such as, if all the agent are new comers then it implies that the organization is not taking advantage of experienced professionals and similarly, if the team are full of incumbents that implies that new ways of thinking is not being encouraged in the organization. Tables 25 and 26 illustrate the node degree evolution observed for the generated small world network and their respective transfer entropy values.

Table 25: Time series showing the degrees of the nodes generated at each time-step

Time-steps Node ↓	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	3	5	5	7	7	7	7	7	7	7	7	7	7	7
1	3	3	5	5	5	5	5	5	5	5	5	5	5	5
2	5	6	6	8	8	8	8	8	8	8	8	8	8	8
3	5	5	5	5	5	5	5	5	9	8	11	11	11	11
4	3	5	6	6	6	6	9	9	3	9	9	9	9	9
5	3	3	3	3	3	3	3	3	7	3	3	3	3	3
6	0	3	4	4	4	4	4	7	6	7	7	7	7	7
7	0	0	0	3	3	3	3	6	6	6	6	6	6	6
8	0	0	0	3	3	3	3	3	3	6	6	6	8	11
9	0	0	0	0	3	3	3	3	9	3	3	3	3	3
10	0	0	0	0	3	6	6	6	3	9	9	9	9	9
11	0	0	0	0	3	3	3	3	3	3	3	3	5	5
12	0	0	0	0	3	3	3	3	3	3	3	3	3	3
13	0	0	0	0	0	3	3	3	3	3	3	3	3	3
14	0	0	0	0	0	3	3	3	3	3	3	3	5	5
15	0	0	0	0	0	3	3	3	3	3	3	3	3	3
16	0	0	0	0	0	0	3	3	3	3	3	5	5	5
17	0	0	0	0	0	0	3	3	3	3	3	5	5	5
18	0	0	0	0	0	0	3	3	3	6	6	6	6	6
19	0	0	0	0	0	0	0	3	3	3	3	3	3	3
20	0	0	0	0	0	0	0	3	3	3	3	3	3	3
21	0	0	0	0	0	0	0	0	3	3	3	3	3	3
22	0	0	0	0	0	0	0	0	0	3	3	3	3	3
23	0	0	0	0	0	0	0	0	0	3	3	6	6	6
24	0	0	0	0	0	0	0	0	0	3	3	3	3	3
25	0	0	0	0	0	0	0	0	0	0	3	3	3	3
26	0	0	0	0	0	0	0	0	0	0	3	3	3	3
27	0	0	0	0	0	0	0	0	0	0	3	3	3	3
28	0	0	0	0	0	0	0	0	0	0	0	0	3	3
29	0	0	0	0	0	0	0	0	0	0	0	0	0	3
30	0	0	0	0	0	0	0	0	0	0	0	0	0	3
31	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Table 26: TE values calculated based on node degrees from table 8

Source Node	Destination Node	Transfer Entropy (S-D)	Transfer Entropy (D-S)
0	1	0.04644672	0.04631231
1	2	0.04631231	0.04644672
2	3	0.0171095	0.003008511
3	4	0.09237862	0.05438259
4	5	0.02586634	0.04631231
5	6	0	0.03303726
6	7	0.06423679	0.008441066
7	8	0.04399981	0.03089852
8	9	0.1321734	0.09031211
9	10	0.04631231	0.03856231
10	11	0.0369023	0.006662312
11	12	0.008522109	0
12	13	0.08358539	0.000199787
13	14	0	0.008322322
14	15	0.008322322	0
15	16	0.09031211	0.004387512
16	17	0	0
17	18	0.002727503	0.03727308
18	19	0.09698611	0.01748029
19	20	0	0
20	21	0.1006949	0.000444193
21	22	0.1048817	0.00063586
22	23	0	0.002727503
23	24	0.002727503	0
24	25	0.1086013	0.000986449
25	26	0	0
26	27	0	0
27	28	0.06867657	0.005675863
28	29	0.117776	0.02315615
29	30	0	0
30	31	0	0

Behavior of the nodes when observed in figure 51, it is seen that the flow of information from the source to the destination nodes initially stays the same and then gradually increases for every two nodes. The network analyzed here is based on team assembly where, at every given time-step a new team of four based on new comers and incumbents is formed and is linked with the existing network.

Initially, nodes 0 and 1 being from the same team have the same amount of information from one to the other. A sudden spike in the graph is observed at node 3 indicating its connectivity and flow of information to node 4.

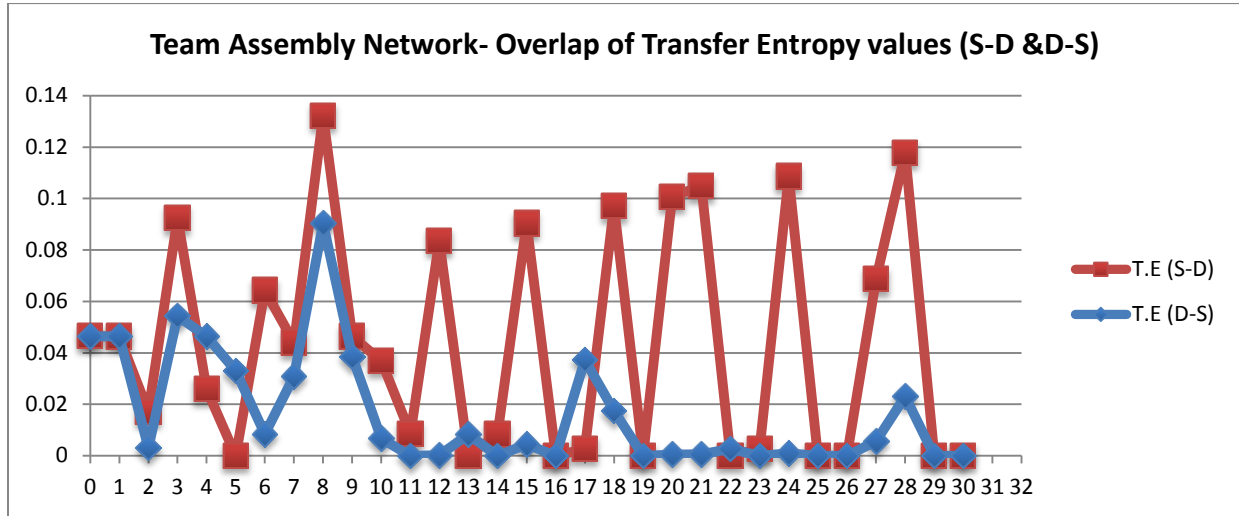


Figure 51: Transfer Entropy values between the nodes of Team based Assembly Network

Starting from node 6 we see an increase in the information flow to node 7 and from node 8 to node 9 implying the new information emergence at nodes 7 and 9 based on the already existing network till that time-step. There on, we see a gradual increase at every 2 nodes until node 29 implying that for every two nodes there is a comparatively increased flow of information indicating the new nodes being connected to the teams and gaining new information about the network until the particular time period it emerges. Nodes 29 and 30 have no associated information flow implying the end of network emergence and no further information transfer. Table 27 shows the entropy measure and R_0 calculated at every time-step of the dynamic network.

Table 27: Dynamic Entropy and R_0 values for generated Team Assembly network model

Time-step	Dynamic Entropy	R_0
0	2.35388575	0.096556474
1	2.8505577	0.085714286
2	2.89599737	0.096638655
3	3.82569805	0.0625
4	5.2500542	0.025412088
5	5.99648194	0.016176471
6	6.79920026	0.011842105
7	7.35759472	0.010093168
8	7.60345154	0.009794919
9	8.18642457	0.007758621
10	8.82604346	0.006324405
11	8.66617109	0.006505977
12	8.99269767	0.006191845
13	9.61364096	0.005269618
14	2.35388575	0.096556474

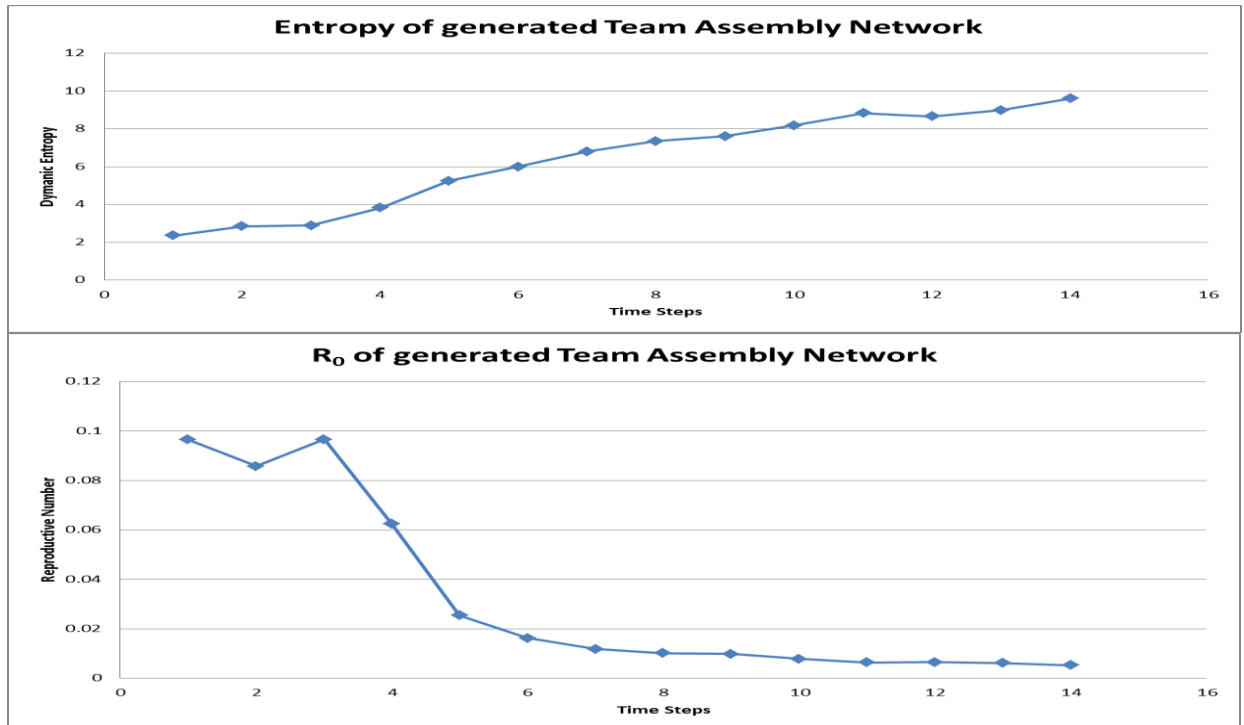


Figure 52: Dynamic Entropy and R_0 Measures of Team Assembly Network Generated

It is observed that as the network evolves, entropy increases implying that the team assembly network tries to reach a disordered state from a high ordered state initially. As entropy (diversity) increases we see that the reproduction number decreases. This implies that when this network is at a high ordered state the spread of information increases and when at a high disordered state the spread of information decreases.

Student Interaction based Network from Facebook platform

Here, we try to analyze the information flow based on undergraduate and graduate student online interaction patterns. An online Facebook group was created as a part of this study to initiate a computer mediated communication platform for the geographically separated students to connect and engage in classroom based meaningful discussions. We use Netvizz, a data extraction tool to collect data from student groups in Facebook social networking platform. The networks constructed using the data gathered at five random time-steps are analyzed using individual node degrees for understanding the interaction patterns observed both qualitatively and quantitatively. Social Networks emerge from a patterned arrangement of interactions based on the actions by individuals in a society. From a hierarchical perspective, any society portrays a network of ties among a set of individuals at a lower level and at a higher level it represents ties and patterns of emergence in a social structures. Two broader divisions of social networks fall into ‘ego centric’ and ‘socio centric’ networks where; ego centric networks portray an individual actor and the effect of the network on that individual whereas, a social centric network portrays patterns of interactions [Carrington et al., 2005] among a set of individuals. Facebook is one of the most used freely available social networking tools among students. Taking into consideration its user friendly interface to connect and share information among a set of actors, this social networking

platform was used to understand information flow in education centric student interaction patterns.

A private Facebook group for the students of Engineering Technology Dept. from DU and the students of the Industrial, Manufacturing and Systems Engineering Dept. at UTEP enrolled for a Green Energy Manufacturing class was created. Creation of this group was aimed at providing collaborative interactive sessions among the enrolled, but geographically separated, students to participate in class related discussions, share freely available resources, and provide individual student insights in the topics and trends related to Green Energy & Green Manufacturing. Students with access to the Facebook group were encouraged to post their insights on emerging green energy technologies and share their in-class and project based experiences. The group moderators posted discussion boards weekly based on student curriculum progress, which enabled students to discuss their individual insights weekly based on the posts. Discussions on student class projects were also encouraged and observed in this group¹¹. Interactions of the enrolled students in the Facebook group over the semester emerged into a student centric academic interaction network. This network consists of a set of actors (23 students in total who joined the group created) -- in this case, enrolled students from both universities, linked/connected by the relationships observed. . It is important to note that all the course enrolled students at both universities were encouraged to join the group however; it was observed that only 23 participated out of 35 students.

As a first step towards analyzing student interactions, some of the topics discussed upon in the group were [Ruane, Chiou and Tseng, 2015]: (a) Benefits of implementing Green energy systems; (b) Students' perceptions of what a Green energy systems is, and the potential of green energy system implementation; (c) Benefits of wind energy and the implications of Wind energy

systems; (d) Benefits of solar energy and implications of Solar energy systems; (e) Benefits of Green Energy and Green Energy Manufacturing.

In order to map of the network at the conclusion of the semester, based on the discussions, posts, and comments observed in the group by both the students and moderators, Netvizz® a data extraction tool, was used to extract data from the Facebook group. This data was then imported into Gephi® software, an interactive network visualization platform. Figure 50 illustrates the raw network extracted from the Facebook group. To better visualize, understand and see interactions in the network, social network analysis tools available in Gephi were used which resulted in the network illustrated in Figure 51.

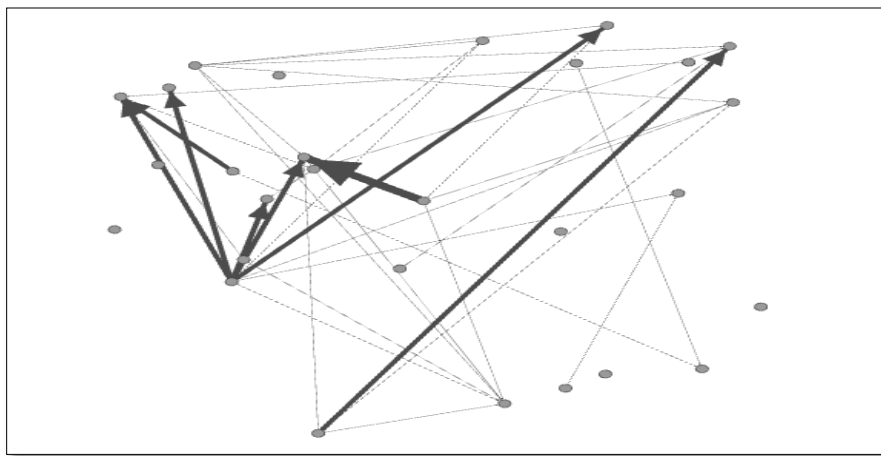


Figure 53: Raw extracted network based on student interaction in the Facebook group

The nodes illustrated in Figure 53 represent the actors (students) of the network and the edges, i.e. the connections between the students (nodes), represent that a given node commented, liked or shared the post made by the node at its corresponding edge (connection). The thickness of the edges represents the number of interactions among the nodes, the thicker the edge, the more often the nodes interacted with each other.

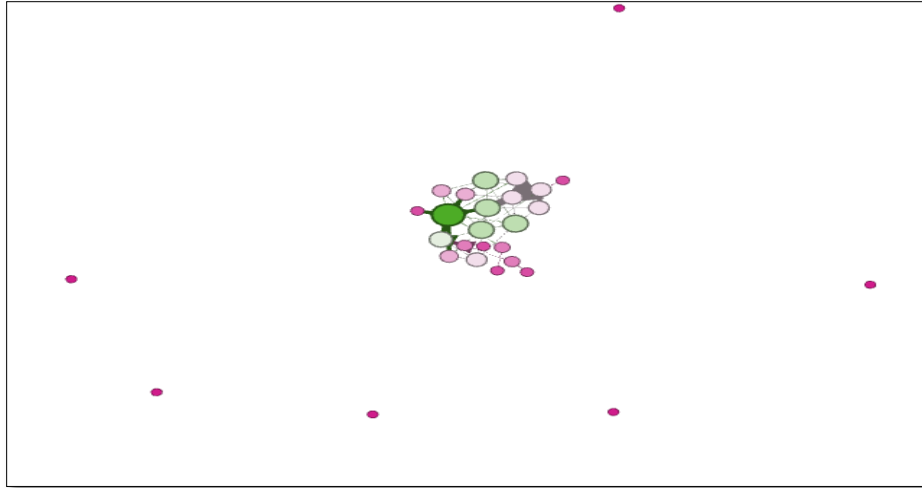


Figure 54: Visualization of the network from Figure 50

From Figure 54, it can be seen that there are outliers in the network, i.e. not all students registered in the Facebook group were actively participating in the discussions. To further understand the student interaction behavior, the outliers observed in Figure 54 are omitted from the analysis, thereby resulting in the network illustrated in Figure 55.

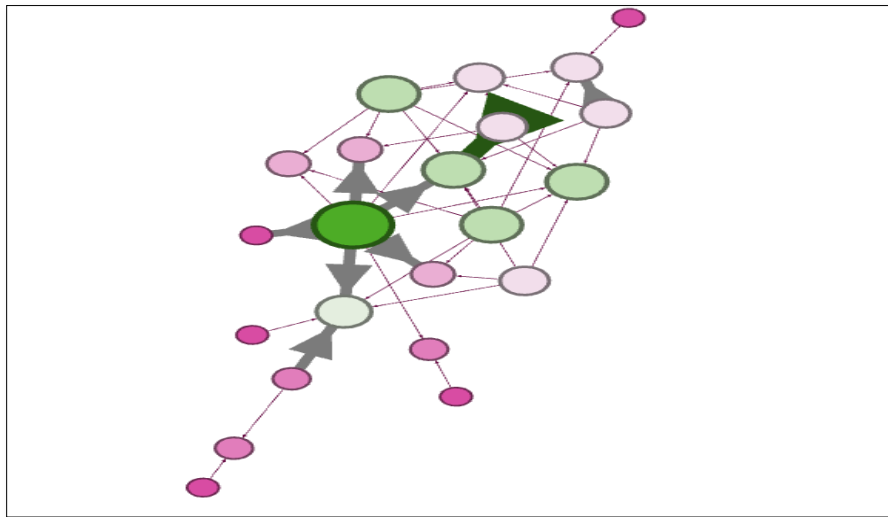


Figure 55: Illustration of student group interactions after omitting the outliers

To analyze the flow of information generated from one node to the other of a considered network based on the observed individual node degree evolution over several time-steps of network

formation, the concept of transfer entropy (TE) is used. TE, when applied to the current scenario, helps to understand how much information generated at one node (student) is responsible for the information obtained by the other node (student). To generate TE values based on individual node degree evolution, data on the number of nodes present along with the individual node degrees over 5 different time-steps were captured based on the student Facebook group discussions over the semester. Figures 53-57 illustrate the evolution of interactions in the student Facebook group over five different snapshots. It also illustrates how students start interacting with each other over the semester. To protect student identity, the nodes are marked with numbers.

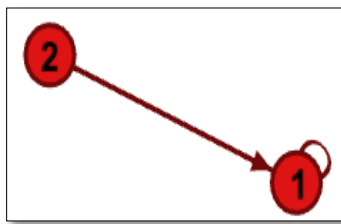


Figure 56: Network observed at time-step 1

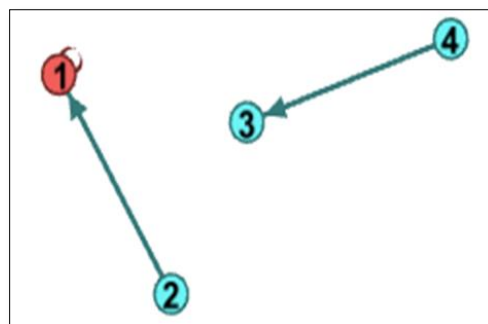


Figure 57: Network observed at time-step 2

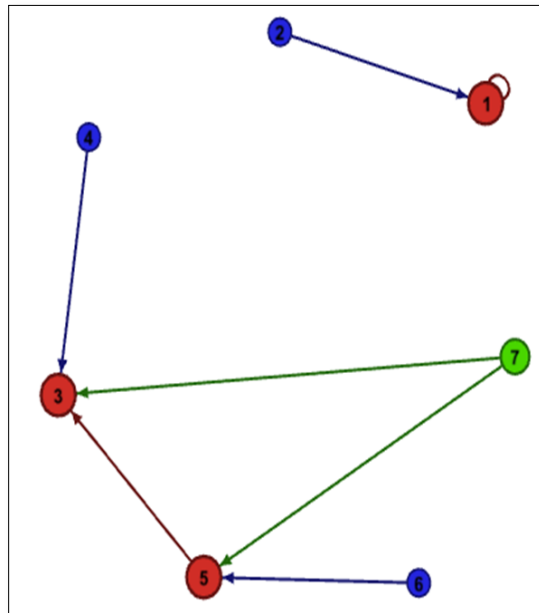


Figure 58: Network observed at time-step 3

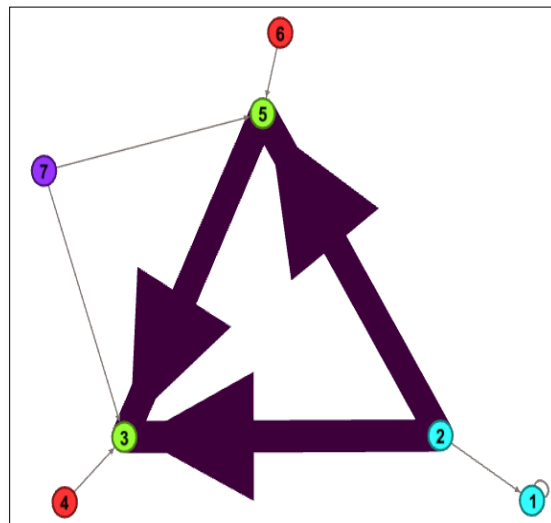


Figure 59: Network observed at time-step 4

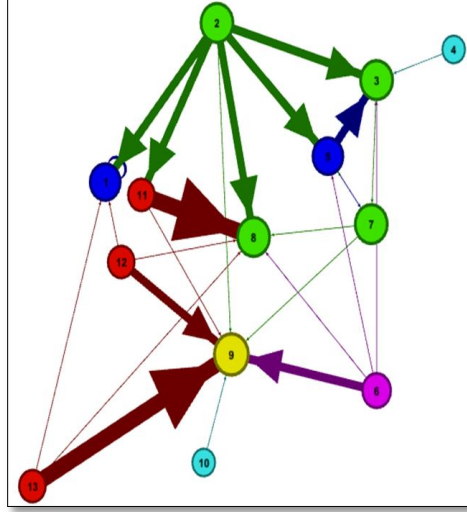


Figure 60: Network observed at time-step 5

To read the captured data that is illustrated in Table 28 based on the 5 different network snapshots captured and illustrated in Figures 56-60, the degree evolution of node 1 over 5 different time-steps is represented by the second row of the table, degree evolution of node 2 is represented by the second row of the table and so on. For calculating the TE values represented in table 29, based on TE equation, joint probabilities are calculated for emerging node degrees observed in Table 11.

Table 28: Node degree evolution over the time-steps captured. (Note: N1 to N13 represent the different nodes in the network; T1 to T5 represent the time-steps)

	T1	T2	T3	T4	T5
N1	3	3	3	3	5
N2	1	1	1	3	6
N3	0	1	3	4	6
N4	0	1	1	1	1
N5	0	0	3	4	5
N6	0	0	1	1	4
N7	0	0	2	2	6
N8	0	0	0	0	6
N9	0	0	0	0	7

N10	0	0	0	0	1
N11	0	0	0	0	3
N12	0	0	0	0	3
N13	0	0	0	0	3

Table 29: Transfer Entropy values calculated based on table 28

Source Node (S)	Destination Node (D)	Transfer Entropy (S-D)	Transfer Entropy (D-S)
N1	N2	0	0.2442191
N2	N3	0	0.2073259
N3	N4	0.09370405	0
N4	N5	0.150515	0.09370405
N5	N6	0.150515	0
N6	N7	0	0
N7	N8	0.09370405	0
N8	N9	0	0
N9	N10	0	0
N10	N11	0	0
N11	N12	0	0
N12	N13	0	0

Analyzing the graph illustrated in Figure 58, based on table 12; TE values suggest the flow of information from a given node to the other. In the graph (Figure 61), the X axis represents the node numbers and Y axis represents the corresponding TE values. It is read as: for the red line, label 1 is the TE value from Node 1 to Node 2; label 2 is the TE value from Node 2 to Node 3 and so on. For the blue line, label 1 is the TE value from Node 2 to Node 1, label 2 is the TE value from Node 3 to Node 2 and so on.

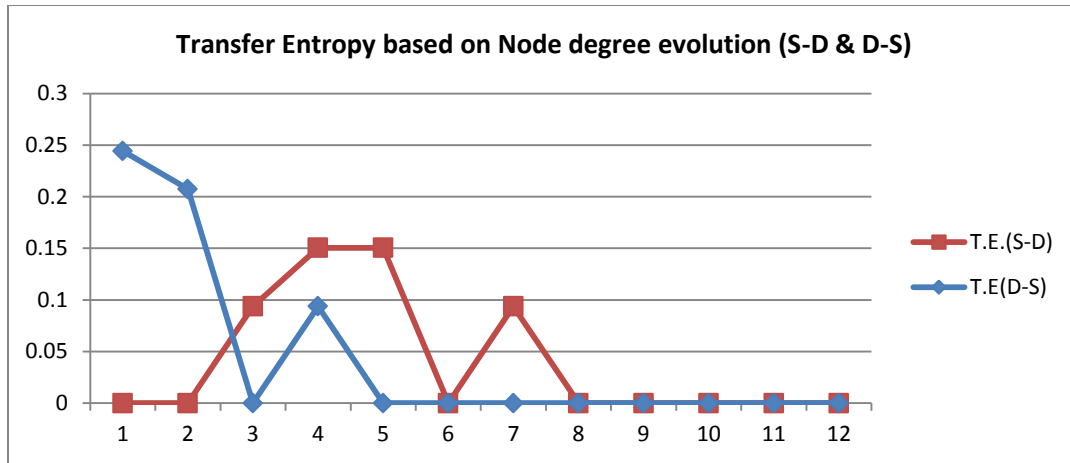


Figure 61: Transfer entropy values between the nodes

At early stages of the graph, the red line when observed, the value of TE from node 1 to node 2 is zero and TE from node 2 to node 3 is zero. This simply suggests the initial network structural formation where, at the first time-step, node2 is connected to node1 and at the second time-step two sub networks are observed with a disconnect between nodes 1,2 and nodes 3,4. Holistically, when the whole network is observed, the graph indicates that while the network starts to evolve, the flow of information from one node to the other increases and then decreases to stabilize at 0. This may imply that the students were active initially, by posting and sharing information in the group. Once new information reached all the group members, the transfer of information among the students slowly decreased. Towards gaining more insight on the flow of information at a lower level from each and every node of the network to the other, transfer entropy values were calculated from each node across the whole network. The results obtained signify that, even if there is more than one interaction between two given nodes it doesn't necessarily represent high information flow between them.

Analyzing networks based on network analysis principles help in identifying important nodes such as network leaders, local connectors/hubs, nodes with high visibility, and nodes with more control over information flow. To better understand the spikes observed in Figure 58, centrality

measures were applied to the networks at each time-step. The node labels and their average respective centrality measures are illustrated in Table 30.

Table 30: Centrality Measures of Nodes (Note: Values highlighted in red are the maximum & values highlighted in green are the minimum)

Node Label	Eigenvector Centrality	Closeness Centrality	Betweenness Centrality
1	0.854015179	0	0
2	0	1.0285714	0
3	0.321968118	0.4375	1.25
4	0	1.475	0
5	0.222004566	1.1666667	1
6	0	1.4	0
7	0.332832107	1	2.6666667
8	0.649487607	0	0
9	0.655746469	0	0
10	0	1	0
11	0.006258862	1	0
12	0	1	0
13	0	1	0

- Node 4 with a higher closeness centrality measure acts as the node that can spread more information in the network, justifying the spike observed in figure 24. The high value of closeness centrality and low value of betweenness centrality implies it has more reach throughout network with its multiple possible paths to reach all the other nodes.
- Node 7 with a higher betweenness centrality measure has more control over the flow of information. Its low closeness and high betweenness values imply that this node also has more control over the node connections from few to many nodes.
- Node 1 with a higher eigenvector centrality measure acts as the network leader (justified by its presence from the initial time-step to the last) even though it may not have strong local influence.

Based on the limited data used to map a network for this case, it shows that even if there is more than one interaction observed among two given nodes, it doesn't necessarily signify high information flow among them. However, the Facebook group provided a very effective platform for students to discuss and observe the topic related to the trends in green energy and green manufacturing. All the students involved in the group actively participated with an exception of few outliers observed in the network. There are several other network characteristics that can be explored, however to limit the scope of this study the analysis was based only on considering the node degrees of the network [Akundi, Tseng and Smith, 2016].

Table 31 shows the entropy measure and R_0 calculated at every time-step of the dynamic network.

Table 31: Dynamic Entropy and R_0 values for captured student interaction network

Time-steps	Entropy	R_0
1	0.811278	0.75
2	1.792481	0.083333
3	2.645593	0.034014
4	2.641604	0.050265
5	3.536381	0.026557

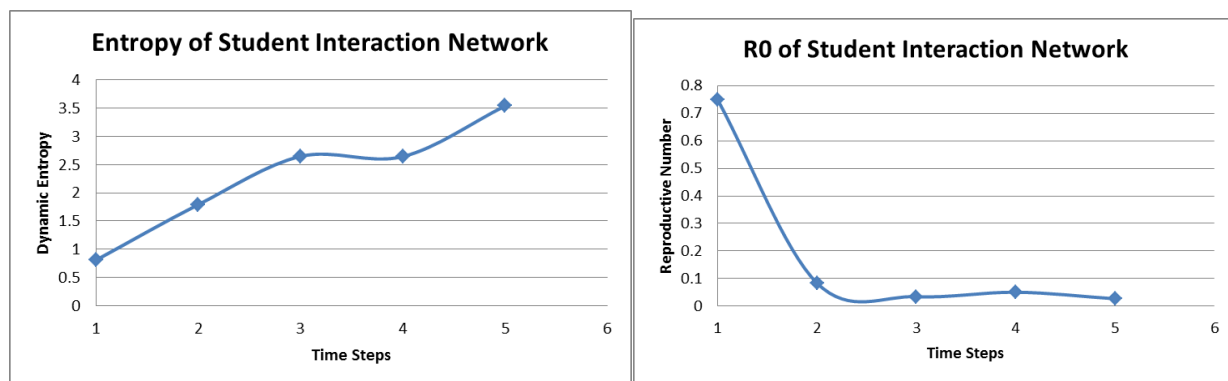


Figure 62: Dynamic Entropy and R_0 Measures of Small World Network Generated

It is observed that as the Facebook based student interaction network evolves, entropy increases implying that network with more nodes evolving over a period of time, the network as a whole

tries to reach a more diverse state. As entropy (diversity) increases the reproduction # decreases and then remains constant. This implies that when the network starts to evolve at the first time-step, there is a more chance of information spread. From the second time-step the reproductive number becomes almost stable indicating that the spread of information in the network remains the same.

The observed results of the entropy based measures applied to Small World network, Preferential Attachment network, Team based Assembly network, and Facebook platform based student interaction network positively correlate the results of Doerr et al [2012]. Doerr et al.,[2012] when analyzing why rumors spread fast in social networks conclude that small-degree nodes quickly learn a rumor once one of their neighbors knows it, and then again quickly forward the news to all their neighbors.

Self-Organization and Criticality in Complex Networks

Self-organization in a system refers to the tendency of a process where in the interactions of the system components tend towards an ordered state from a highly disordered state. Several systems that are spatially and temporally distributed are usually correlated with a critical point of phase transition in statistical physics [Valverde et al., 2015]. It is observed that complex systems enhance their functionality in terms of information processing, tolerance, and emergence when they operate at near criticality [Valverde et al., 2015]. Self-Organized Criticality (SOC) refers to a phenomenon in dynamic systems where the system shifts towards a critical point as an attractor. There have been several attempts to understand failures in complex systems by applying the phenomenon of SOC using several analytical methods to simulate critical phenomenon [Cajueiro and Andrade, 2010].

The Bak-Tang-Wiesenfeld (BTW) model was the first ever example explaining the concept of SOC in a dynamic system [Bak, Tang and Wiesenfeld, 1987]. BTW model is based on observing dynamics in a sand pile where, when grains of sand are continually poured on a given plane, they buildup upon each other, forming a structure defined to the plane. These additions of sand grains develop to a critical state, which, when exceeding a specific threshold, triggers an avalanche in the sand pile where it collapses and transfers the sand grains across the plane creating a cascading effect. This shows that the transition in the sand pile is a result of the interactions among the individual grains.

The SOC nature in the sand pile is portrayed in the process where the sand pile self-organizes itself into a minimal stable state, but where the addition of a single grain of sand may initiate an avalanche. Over a period of time, the sandpile self-organizes again back to its critical point (minimally stable state). The principle of SOC states that many complex systems emerge towards criticality under many conditions [Valverde et al., 2015].

Motivated by BTW model and the goal to understand criticality in a complex networks, a NetLogo based simulation was developed to generate a network and simulate its emergence towards criticality. Criticality here is defined as the state of a network where the network shifts from an initial disordered state towards a highly ordered state. Translating the sand pile model towards networks, Table 32 illustrates the comparisons drawn to develop an agent based simulation model.

Table 32: Comparisons drawn towards networks from the sandpile model

BTW Sandpile Model	Networks
Sand Grains	Network Nodes
Sand grains stacked on each other	Network Nodes connected to each other
Continual changing structure of sand pile	Continual dynamic network emergence
Sand pile at a critical state	Fully connected network at critical state

When a sand pile is at a critical state, addition of a new grain triggers an avalanche and slowly it reorganized itself towards a critical state again	When new nodes are introduced to a maturely connected network, the nodes tend to interact with each other, losing some connections, and thereafter self-organizing to reach a highly connected state gain
Random displacement of grains after cascading avalanche	Random removal of edges / connection between the nodes of a network while trying to get maturely connected

A given network is said to be at a critical state when all the network constituent nodes are in a state of information transfer maximizing connectivity to each other, thereby making best use of the resources available in a given scenario. This is correlated to the fact that complex networks at a critical state are highly functional in terms of information processing and we believe in complex networks (networks mapped from social, human interaction, organizational interaction, educational setting based interactions and so on) high functionality is achieved when all the available resources are fully utilized.

The notion of criticality has found a high interest recently towards understanding system behavior. This is because it helps to bridge a connection between the changes that are observed at a microscopic scale and how they cascade and affect the system at a macroscopic scale. NetLogo based simulation model is used towards understanding the structural implications on information processing capabilities in a system at criticality. Figure 63 illustrates the interface developed to generate a network and Figure 64 shows the conceptual view used.

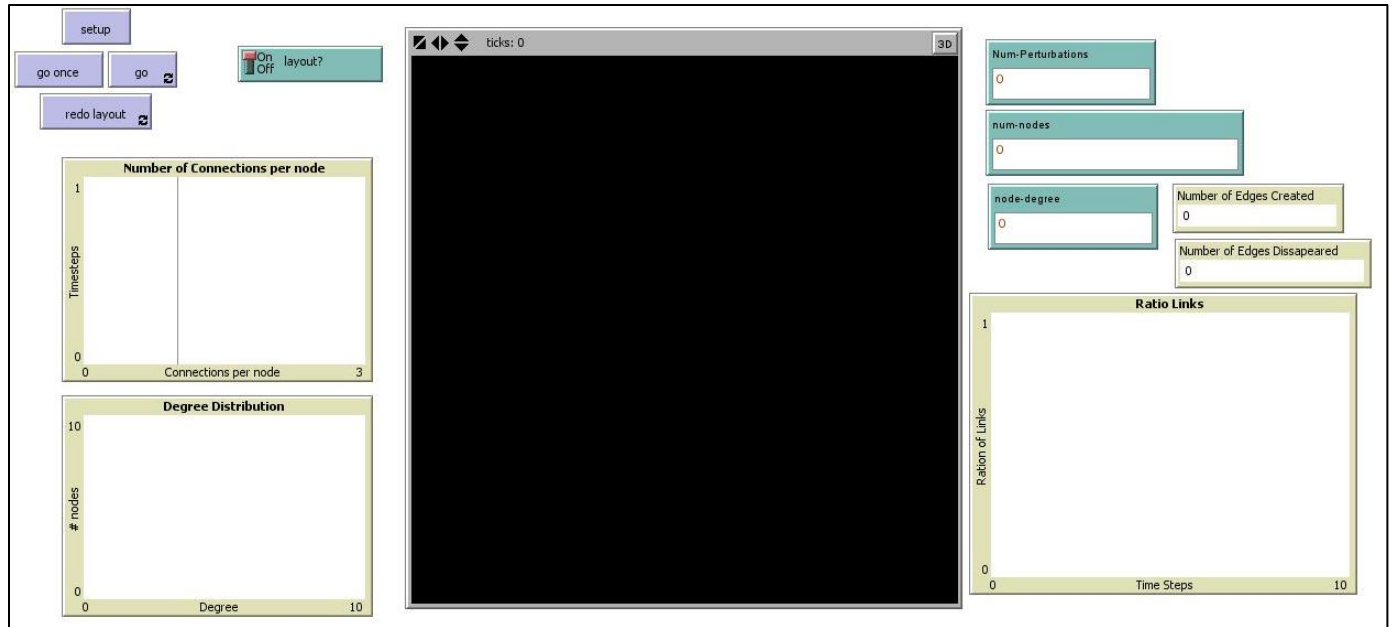


Figure 63: Screenshot of Netlogo Interface Developed

Conceptual view of Netlogo model

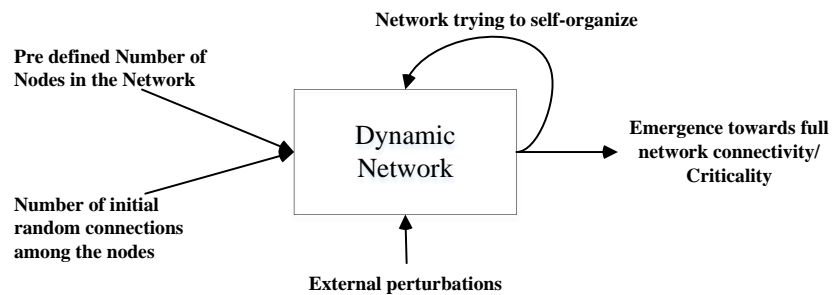


Figure 64: Conceptual view based on which the netlogo model is developed

The main aim of the network model developed is to reach a highly ordered state from an initial disordered state. Every node (actors) in a network is either directly or indirectly reachable to every other node. Initially the model has a predefined set of nodes without any connections between them. As the model initiates and starts running, edge (links) are formed between the nodes of the network at each time-step. As the model continues, at each time-step a random missing edge between nodes is picked and then created. If two given nodes already have an edge

connecting them then new nodes are picked to be linked by an edge. The model continues until the network is fully connected which is determined by meeting the following criteria:

$$\text{if } 2 * (\text{number of Links in network}) \geq (\text{number of nodes}) * (\text{number of nodes} - 1)$$

Once the network is fully connected, perturbations (external nodes) are introduced in the model to simulate a cascading effect where in all the nodes in the network try to get fully connected again. To portray the effect of avalanche cascading onto the nodes, at every time-step two nodes are randomly picked and an edge is removed if there link between them. This repeats until the network is fully connected again thereby displaying self-organizing behavior once it reaches a critical state and an avalanche is triggered by introducing external nodes in the network as perturbations.

Mentioned below are the defined inputs and the outputs of the model:

- Number of Nodes (num-nodes): This is a user defined input which determines the number of nodes in a network.
- Initial Number of connections in the network (node-degree): This input defines the number of random initial edges to be created among the nodes of network.
- Number of Perturbations (num-perturbations): This input defines the number of external nodes to be introduced in the network once it is fully connected.
- Number of Edges Created: This factor defined in the model reports the total number of edges created in the network.
- Number of edges disappeared: This factor defined in the model reports the total number of edges removed between the nodes in the network while trying to reach full connectivity.

- Layout: This defined procedure in the model helps the nodes to position in the network at every time-step based on the emergent dynamic connectivity.

Illustrated in figure 62 is an example of the model on how a network generated emerges to its critical state of full connectivity and how it self organizes to a critical state again after introducing an avalanche.

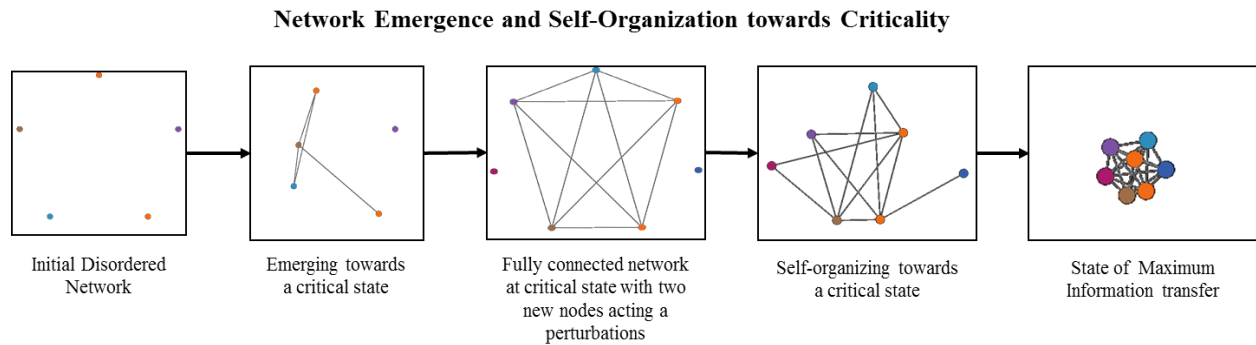


Figure 65: Network evolution towards a self-organized critical state

For a better insight on network node interaction dynamics, figure 66 illustrates average number of connections per node along with the ratio of number of links that appear and disappear among the node across emergence when 100 nodes are initially considered in the network and once the network reached critical state (i.e. full connectivity), 20 nodes are introduced into the network to create an avalanche and thereby the network gradually self organizes itself to reach the critical state again.

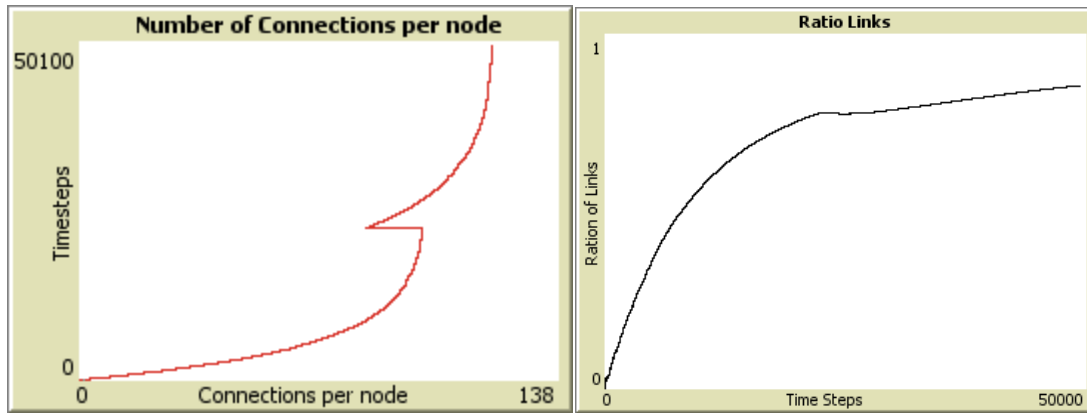


Figure 66: Graphs illustrating Number of connection per node and the ratio of links appearing to disappearing over the network evolution

The spike observed in the graph illustrating the number of connections per node represents the start of an avalanche in the network when external perturbation are introduced and the gradual self-organization of the network after the avalanche is seen by a smooth transition in the graph after the spike. Also, the graph on ratio of links appearing to disappearing when observed overtime exponentially increases at first and then slowly stabilizes.

To understand the structural implication on the network while moving towards criticality and self-organizing once an avalanche is introduced, a network of 10 nodes over 96 time-steps is initially generated using the developed model. The network once fully connected, 5 external nodes were introduced as perturbations to create an avalanche in the network (A total of 316 time-steps were taken for the network to be fully connected again after the avalanche). Over the emergence of the network, the change in the number of connections to node (i.e. variation of individual node degrees) is captured to calculate network dynamic entropy. Figure 67 illustrates the dynamic change in entropy of the network.

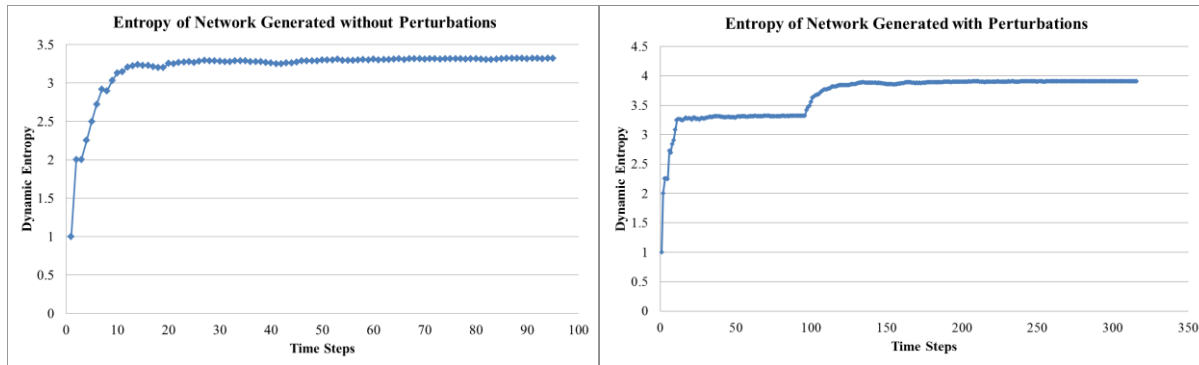


Figure 67 (a, b): Dynamic Entropy Measure of the Networks Generated (30(a)-Graph to the extreme left illustrates entropy change in network of 10 nodes without any perturbations; 30(b)-Graph to the extreme right illustrates the entropy change in network of 10 nodes where 5 external nodes were introduced to create an avalanche)

The entropy values calculated clearly illustrate the change in network dynamics where a sudden spike is observed in Figure 67(b) at time-step 97 representing the avalanche in the network. The graph trend line after the spike slowly stabilises representing the self-organization in network nodes trying to reach full connectivity.

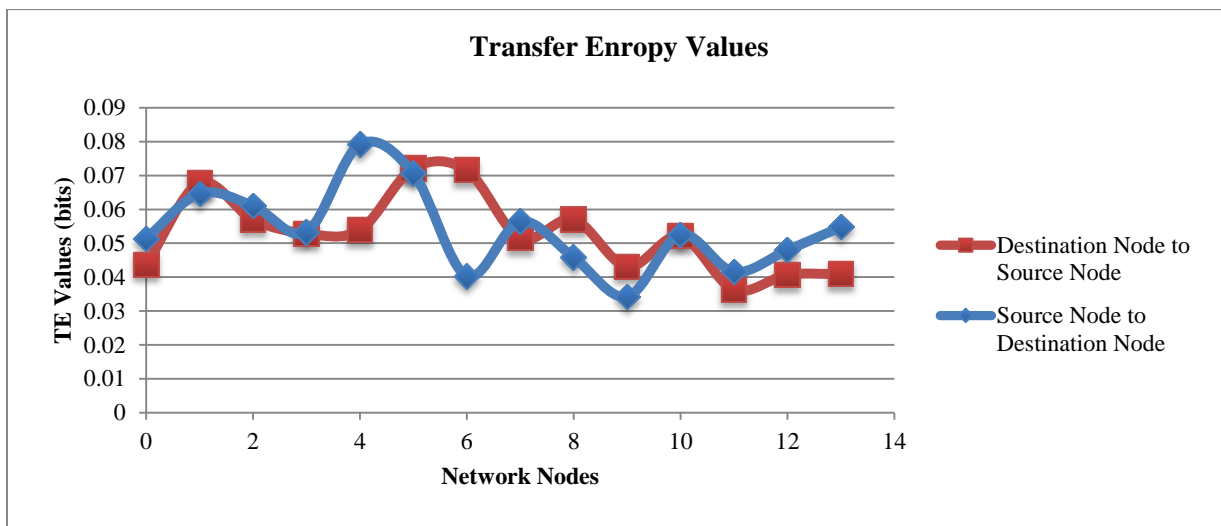


Figure 68: Transfer Entropy Values across Contiguous Nodes

Also, to understand the information dynamics over the network evolution transfer entropy values are calculated to identify the flow of information from one node of the network to the other. These values are calculated based on observing the change in node degrees over time (15 Nodes

over 316 time-steps). Figure 68 illustrates the graph showing the amount of the information flow from one node to the other. To read the Figure, along the X-axis blue trend line at label 0 represents TE in bits from Node 0 to Node 1 and the red trend line at label 0 represents the TE value from Node 1 to Node 0. Further to evaluate the nodes, the TE values calculated from each node of the network across all the other nodes are represented in Figure 66.

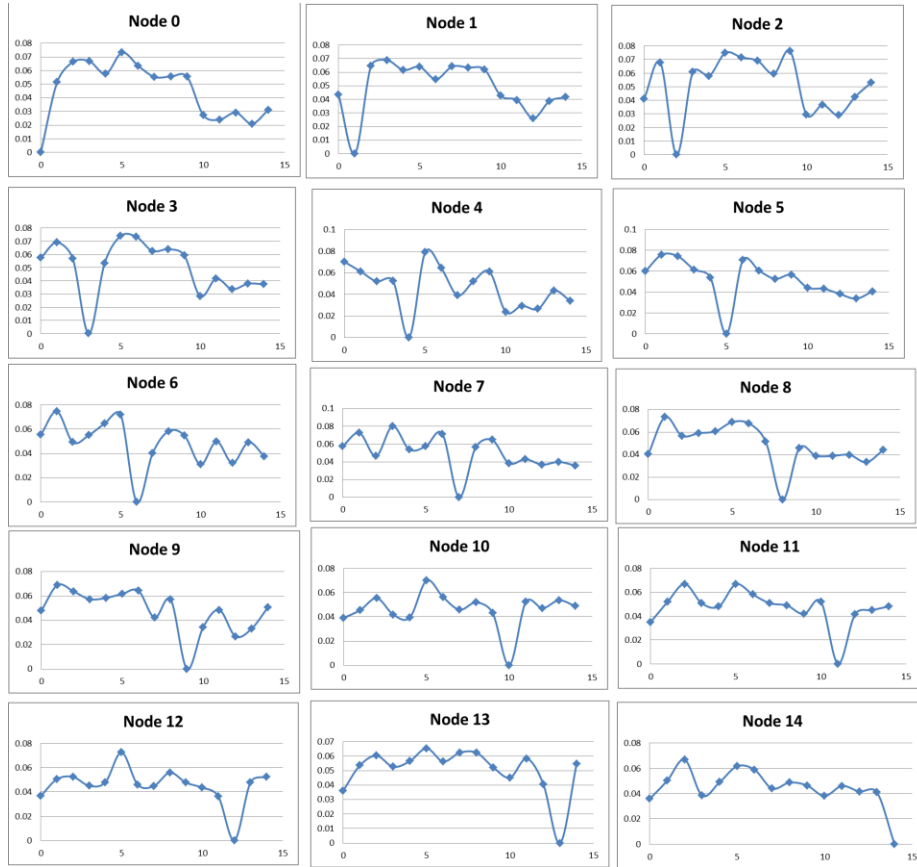


Figure 69: Node TE values of network generated ((A total of 15 Nodes over 316 time-steps considered)

Observing figures 68 and 69 it can be seen that, when a network evolves from a disordered state to a highly ordered state where it is fully connected all the contiguous nodes have almost numerically equal transfer entropy values. Subsequently, every node in the network follows the same information flow trend across all the other nodes. To analyze if this trend observed holds true throughout the network, a network with 10 initial nodes is generated and once it is fully

connected, a set of 5 new nodes are introduced to the network twice. The observed dynamics of the node degrees is captured over 880 time-steps. Figure 70 illustrates the transfer entropy values calculated from every node in the network to the other.

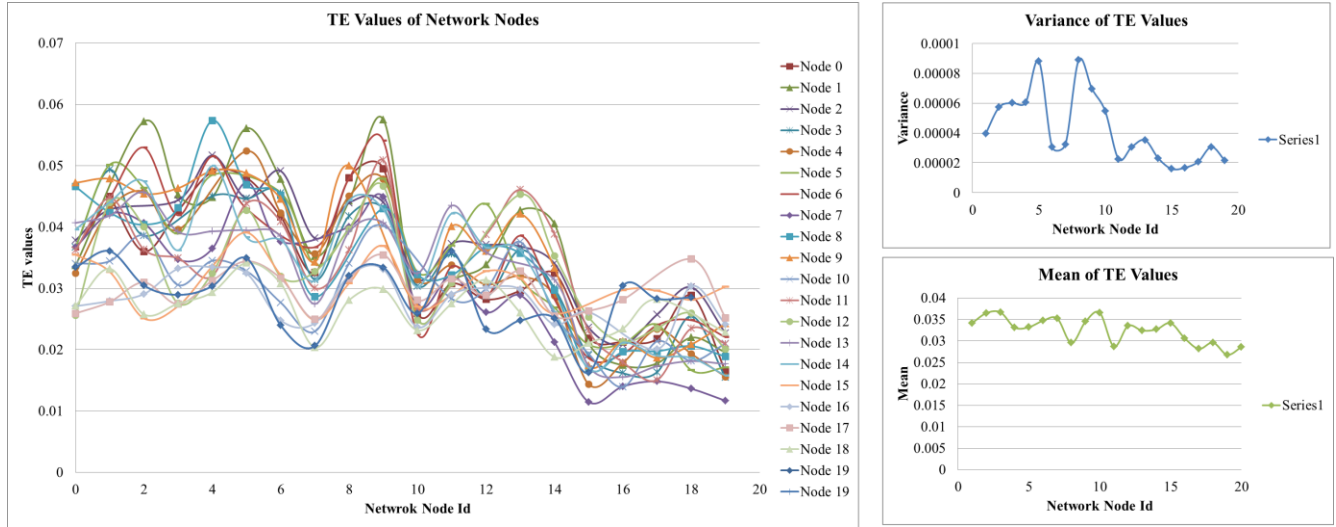


Figure 70: TE Values among Network Nodes (A network with 10 initial nodes where a set of 5 new nodes are introduced twice in the network over its evolution for 880 time-steps)

A decreasing trend in the graph is observed with an initial variability of TE values between nodes 0 to Node8. The fact that the trend line of variation among the TE values is slowly reduced indicates that the bidirectional flow of information among the nodes is almost equal throughout the network making it a highly desirable setting for efficient flow of information among the nodes.

Information transmitted across the nodes of a complex network over its emergence, based on the considered examples is explored using the concept of transfer entropy. The cascading effect of network nodes and their interactions on the structural aspect of network evolution is also explored using the defined concepts of dynamic entropy and reproductive number across various network scenarios.

To further understand the implication of emerging network structural aspects on individual nodes, the concept of information processing using Transfer entropy is used. The information available by the nodes of a network is obtained by calculating the transfer entropy values across all the constituent nodes of a network based on their individual node degree evolution. Once the transfer entropy values are calculated, the information available by a node is given by:

$$I_{node} = \sum_i \sum_j TE(i, j) + TE(j, i)$$

Where ‘*i*’ is the label of the node in question and ‘*j*’ corresponds to the label of all the other nodes across the network.

Accordingly, the transfer entropy values are calculated for all the constituent nodes of Preferential Attachment network, Small World Network, Generated Network in Netlogo of 10 nodes where and 5 external perturbations are introduced at its critical state, and a network consisting of 10 nodes evolving towards a critical state. Figures 34 and 35 illustrate the total information available by nodes mapped across their average degree over the network dynamics. In Figure 71, the green line shows the trend observed in the nodes of preferential attachment network and the blue line represents the trend observed in the nodes of Small world network.

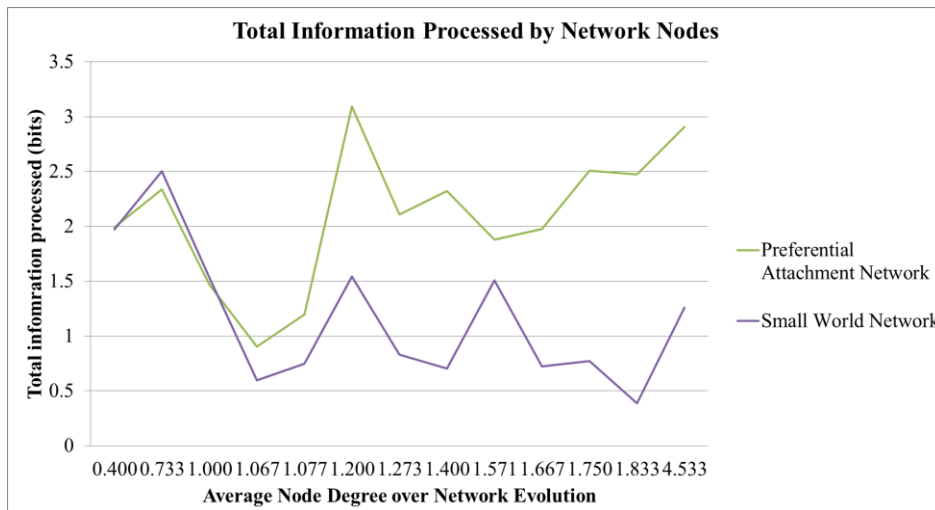


Figure 71: Information available mapped across the average node degrees observed (Green Line represent preferential attachment network, and Blue line represents Small World network)

Comparing the trends observed it is seen that, in preferential attachment network as the average degree of a node increases the total information available by the node increases where as in small world networks the information available by a node gradually decreases as the average degree of the node increases. These trends observed in the graph are evident based on the fact that as the structure of a preferential attachment network evolves, the new nodes that appear at every time-step of its emergence attach to the node with highest degree increasing the reachability of the node for processing more information. Similarly, the structure of the generated small world network at each time-step moves from an initial ordered state to a state where a random connection is picked and then rewired. This illustrates that the nodes that are initially ordered are able to process more information though they have a less comparative average degree.

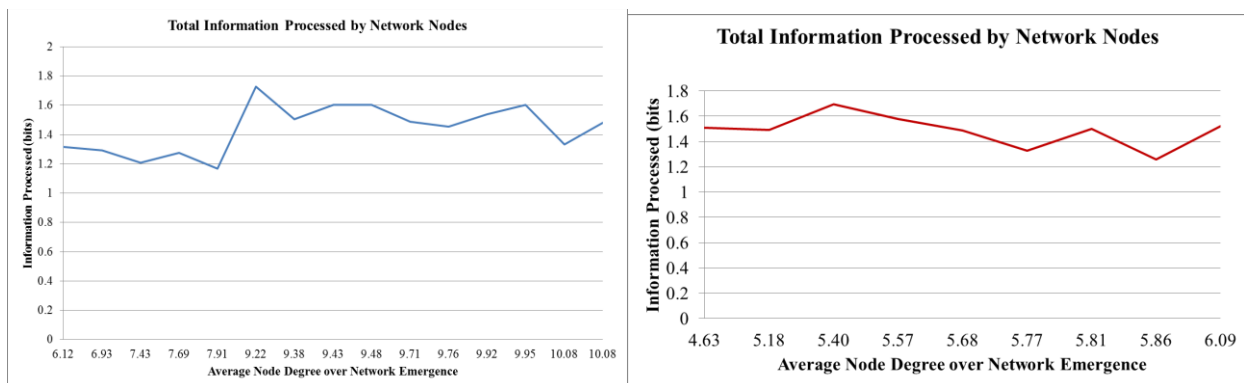


Figure 72: Information available mapped across the average node degrees observed (Blue Line represents the network generated where perturbations are introduced, and Red line represents the generated network free of perturbations)

In Figure 72, the Blue Line represents the network generated where perturbations are introduced, and Red line represents the generated network free of perturbations. Based on the trend observed, it is clearly evident that there is a sudden increase in the information available in the nodes of the network where perturbations are introduced. This trend (blue line) later slowly stabilizes

illustrating the shift in network towards criticality where all the nodes are connected to each other, processing the same amount of information on an average. The trend (red line) in the nodes of the generated network free of perturbation illustrates that all the nodes on average process the same information across the network moving from an initial disordered state to a highly ordered fully connected state.

Expected Utility Value and Prospect Theory

Drawing a parallel to the NetLogo model developed and illustrated in Figure 60, an example of a highly constantly emerging complex network is considered. Based on the defined notion of complexity, complexity may come to play in a network by introducing human actors. This inclusion of humans into a network introduces decision making, where in a node is contemplated to be a decision maker, and the creation of an edge between two given nodes to be an act of communication. Deciding whether a node communicates with another in a given network depends upon discretion and/or several other influencing factors. To set a relevant scenario, an example of a commonly observed complex network in an academic scenario (i.e. a classroom structure) is considered. In a classroom, when observed, the willingness of a student to communicate with another person is a totally independent act of free will. When mapped as a network, the creation of an edge among two given actors (students, in this example) is based on several factors where students personalize their decision to interact, based on past influences, personal experience, values and sometimes even on personal bias (phenomenon in which a genuine limitation in a thought process is observed depending upon situational contexts, false senses, and attributions) [Akundi and Smith, 2013]. To facilitate in assessing such human

decision-based interaction structures, Expected Value theory in human decision-making is incorporated to the NetLogo interface developed.

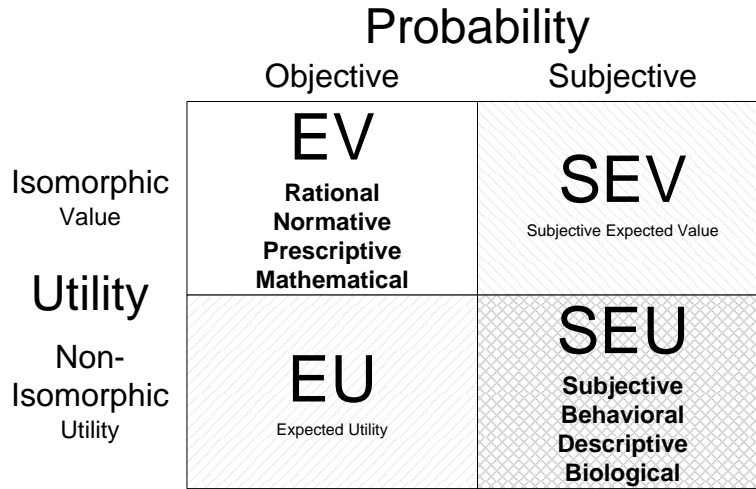


Figure 73: Domains of heuristics and biases in the Expected Value Model as described by Smith [2006].

The domains of heuristics and biases in the expected value model in Figure 70 developed by Smith [2006] are based on 4 different variations of the expected value model created by Edwards [1955]. In the figure, the top left quadrant represents the theory of expected values based on objective probability and objective value or payoff considered. Here, the values of both the probability and payoff values of decision making considered are objectively based, and do not consider any opinion of individuals. In contrast, the lower right quadrant in Figure 73 illustrates the subjective expected utility value, where subjective probabilities and subjective utilities are considered. To summarize the two main theories of human behavior illustrated in Figure 73, the mathematical theory based model (in the upper left quadrant) illustrates the view that humans should utilize to make decisions using objective values of probability and payoff to calculate the expected value of a choice in question whereas the behavioral view (in the lower right quadrant) describes the actual decision making by individuals [Smith, 2006].

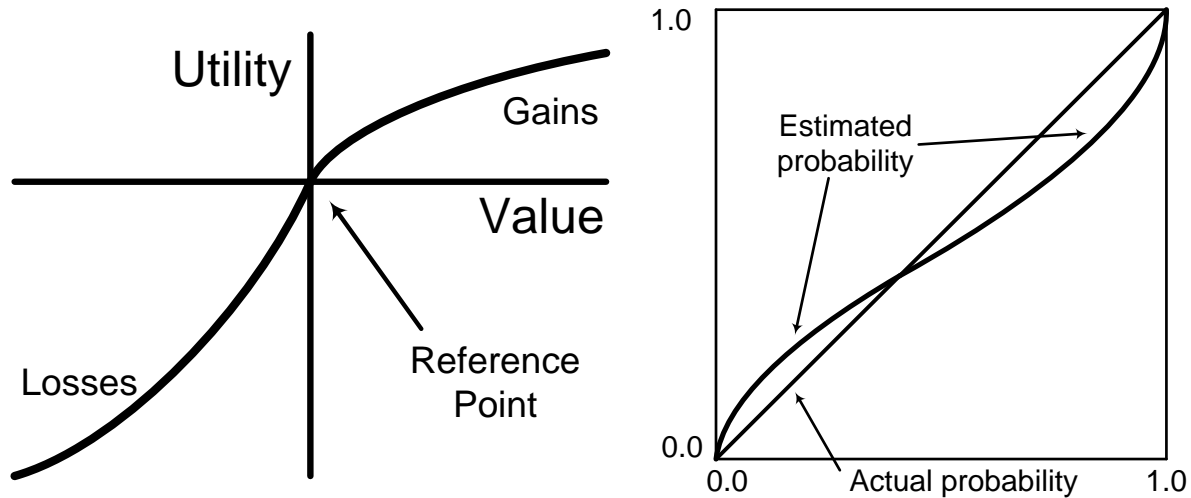


Figure 74 (a, b): Subjective Utility and Subjective Probability as defined by Prospect Theory [Smith, 2006]. (74 (a) - Left most figure shows subjective utility versus values based on prospect theory; 74 (b) – Right most figure shows the translation curve of objective to subjective probability values based on prospect theory)

To further understand the subjective human decision making process, and its variation from rational decision making, Prospect Theory, a widely accepted descriptive theory of human judgement and decision making under uncertainty, is introduced. Subjective utility according to prospect theory captures the utility values in a subjective manner considering a subjective reference point based on a given individuals perspective. Subjective probability and subjective utility as described by Prospect Theory is illustrated in Figure 74. Figure 74 (a) shows the translation of objective values into subjective utilities and Figure 74 (b) illustrates the translation of actual probability values to estimated subjective probability values, where it is observed that, small probabilities are overestimated and large probabilities are underestimated subjectively by individuals [Smith, 2006].

To facilitate the scenario of human interactions in the Netlogo model developed, the concept of utility theory is incorporated to assess a given state of a dynamic complex network over its emergence. To limit the scope, top left quadrant illustrated in figure 73 is considered where only the objective utility and objective probability values are used to calculate an expected value.

Based on the previously mentioned views of complex networks and criticality in complex networks, mentioned below are few conditions identified in order to incorporate expected utility values into the models:

- *Condition A:* At any given time over the network emergence half of the available nodes in the network look for a payoff of highest connectivity, signifying their willingness to make use of all the available resources in the network. The payoff for each node in this case is represented as: $(N - 1)$ with N being the total number of nodes in the network.
- *Condition B:* At any given time over the network emergence the other half of the available nodes in the network (which do not fall under condition A) look for a payoff of a comparatively less connectivity, signifying their willingness to make use only few of the available resources in the network. The payoff for each node in this case is represented as: $\frac{(N-1)}{2}$ with N being the total number of nodes in the network.

To calculate the expected utility value at each time-step, mentioned below are few assumptions for determining objective probabilities and objective utility values for each node.

- In a network of n nodes at any given time, it is assumed that the nodes with their corresponding identification numbers from 0 to $\frac{N}{2}$ follow condition 1 and the nodes from $\frac{N}{2} + 1$ to N follow condition 2.

Considering N to be the total number of nodes in the network and E_t to be the number of Edges of a given node at a given time-step t

With Regards to the nodes following condition 1

- The objective utility value of every node will always remain to be $N - 1$

- The objective probability value of every node changes based its number connections at every time-step based on the formulation $(\frac{1}{N-E_t-1})^{(N-E_t-1)}$

With regards to nodes following condition 2

- The objective utility value of very node will always remain to be $\frac{N-1}{2}$
- The objective probability value of every node changes based its number connections at every time-step based on the formulation $(\frac{1}{\frac{N-1}{2}-E_t})^{(\frac{N-1}{2}-E_t)}$

Following the said conditions and assumptions, the Netlogo model previously developed is enhanced. A simple dynamic network of 4 nodes is simulated and presented to illustrate the model. Figure 75 illustrates the dynamics of a sample network of 4 nodes.

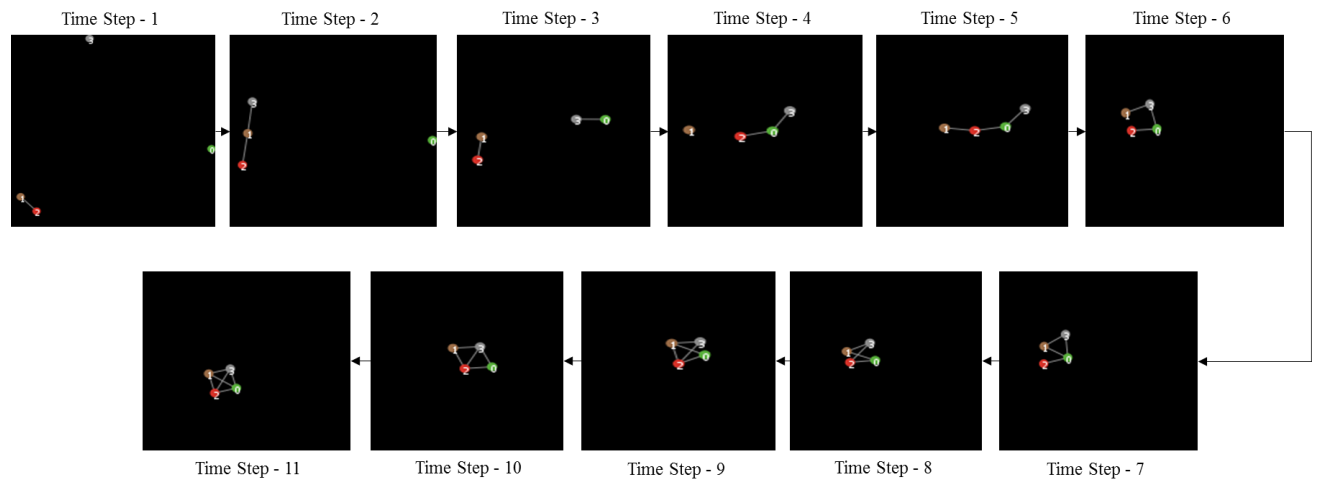


Figure 75: Network Dynamics of a simple network with 4 nodes. At every time-step the corresponding network state is represented.

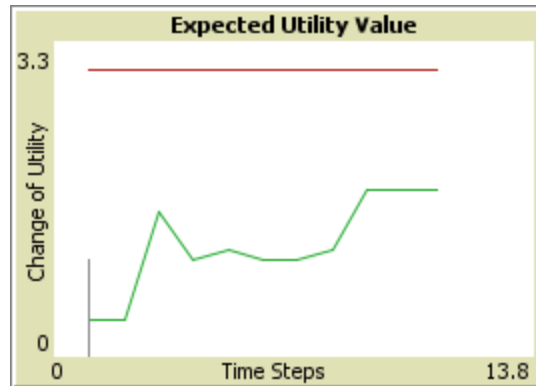


Figure 76: Expected utility calculated at every time-step of the network illustrated in figure 75

When observing the represented network dynamics, it is seen that at the time-step 1 only two nodes are connected to each other showing the interaction among the two nodes while the other two are isolated. At time-step 2, node 0 is connected to node 1, which in turn is connected to node 2, isolating node 3. The expected values when observed for these two states, remains the same. Moving on to the third time-step, node 1 is connected node 2, and node 0 to node 3, which is a better connected state comparative to the previous two states where every given node in the network is connected to at least on other with an increase in the expected value. Node 1 at time-step 4 is isolated whereas the other three nodes have a connection each resulting in a decreased expected utility value compared to the previous state. Similarly observing the state of the network at other time-steps, whenever the nodes of the network are efficiently connected an increase in the expected utility value of the network is observed. Finally, at time-steps 10 and 11, the expected utility value increases and remains the same portraying that the network connectivity for information flow at both the states remains the same. To corroborate with the expected utility values calculated, the concept of clustering coefficient is used to see if expected utility value obtained indeed reflects upon the network connectivity.

Clustering coefficient, a concept from social network analysis is a measure illustrating the degree to which the nodes in a given network tend to cluster together. It can also be seen as a measure

of number of triangles in a network. The clustering index of a node is given by [Saramaki et al., 2007];

$C_i = \frac{2t_i}{k_i(k_i-1)}$ Where, t_i is the number of triangles around the node and k_i represents the degree of the node in question. When $C_i = 0$ it represents that none of the neighbors of the node are connected and when $C_i = 1$ all the neighbors of the node are connected.

The average clustering coefficient values of the network in figure 72 when calculated at each time-step, it is seen that indeed it is positively correlated to the obtained expected utility values. Please see Figure 77.

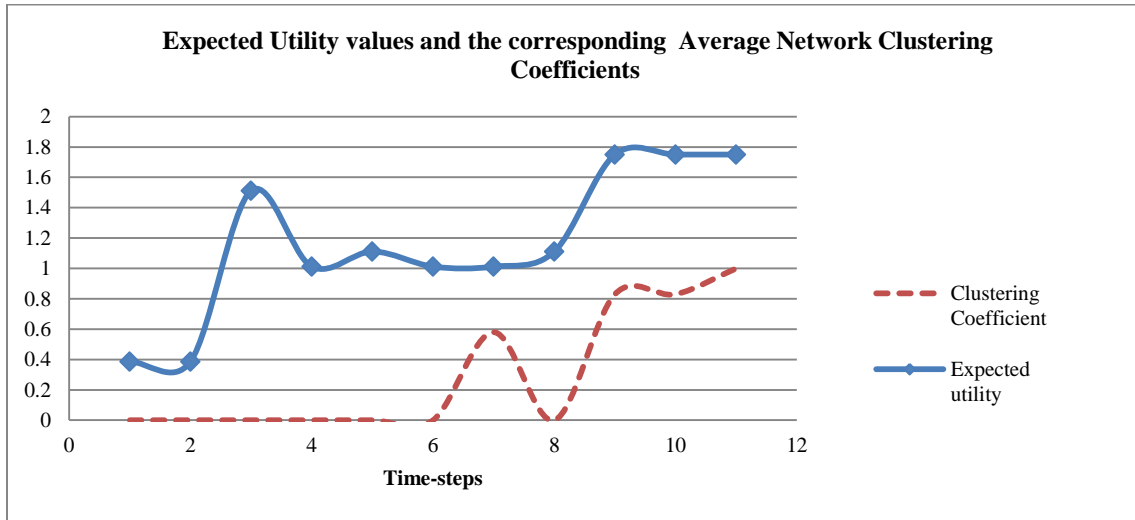


Figure 77: Expected Utility values with reference to corresponding average network clustering coefficient values

The observed dynamic of network emergence and corresponding Expected Utility value illustrates that the state of the network associated with the highest expected utility value when calculated strictly based on objective utility and objective probability values help to comparatively identify a better network state over its emergence.

Considering the applicability of this model, we draw back to the example of classroom structures in assessing constantly changing classroom structural states where, expected value theory is

incorporated to the model for introducing the heuristic of human decision making. Toward this example, the nodes can be contemplated to be actors of a classroom and the edge appearing between two given nodes to be representative of a classroom based interaction. Please see Figure 78.

The developed model can be used as tool by the evaluators to simulate all the possible interaction criteria in a classroom on given N number of students. The number of students once identified in a given class, this can be used as an input to the netlogo model. Running the model helps to generate several interactions patterns among the defined number of nodes. Each time-step of the simulation helps to generate a unique node interaction pattern similar to the illustrated in figure 75. The possibility of an interaction disappearing and reappearing among two given nodes (in this case student) over a period of time is also considered in the model.

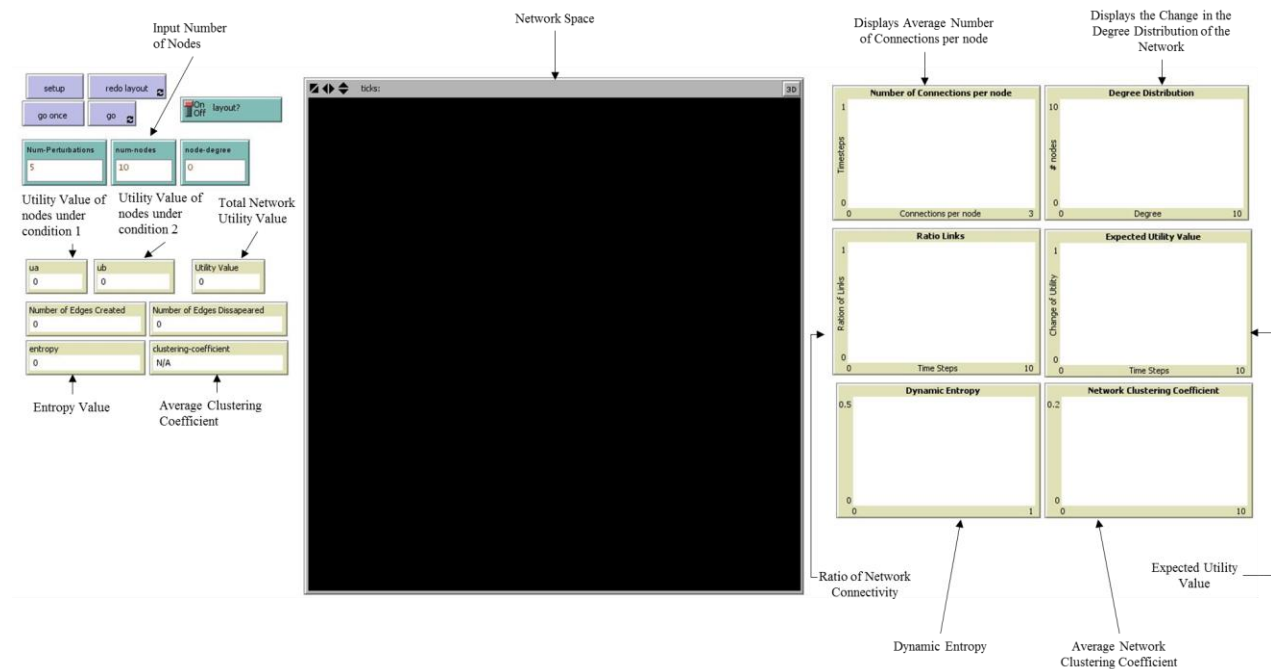


Figure 78: Netlogo model Interface incorporating Expected Utility Value, Clustering Coefficient, Dynamic Entropy, Average Degree Distributions

The model while running, exports the data into the monitors developed to keep a track of Dynamic Entropy, Expected Utility Value, Clustering Coefficient, Ratio of links appearing to disappearing, and the Number of Connections on average per node. The network of student's interactions when evaluated based on the reporting output values of the model; expected utility value at every time-step helps to identify a specific network state of high expected utility value facilitating in narrowing down efficient classroom student interaction patterns, dynamic entropy help to understand network structural coherence and diversity of student interaction patterns. The data on individual node degree evolution when exported to from the model, facilitates in a transfer entropy based analysis (please see previous examples) in identifying a given node (i.e. students) information processing capability along with the information processing capability of the class as a whole.

Chapter 6 - Conclusion

Complex systems are relatively large when compared to complicated systems and tend to have many constituent subsystems and interfaces. Systems are moving towards increased complexity day by day with the addition of new functions and capabilities towards addressing and finding better solutions. To address the challenge faced in finding better solutions in the complex systems domain, a better understanding of what it is that leads to complexity is required [Eisner, 2011]. To better understand what a complex system is, we refer to the following definition:

“Complex systems are networks made of a number of components that interact with each other, typically in a nonlinear fashion. Complex systems may arise and evolve through self-organization, such that they are neither completely regular nor completely random, permitting the development of emergent behavior at macroscopic scales.” (Sayama, 2015, p.3)

Complex systems science enables the development of conceptual and mathematical tools to understand and describe the systems constituents of interdependent sub-systems. Systems science also provides a platform for interdisciplinary applications of structural and dynamical properties of complex systems [Sayama, 2015]. The field of complex systems, in high attention among the research community, is currently explored across several domains. Quantifying the complexity of a system constituting several interacting sub-systems, both holistically and specifically applied scenarios, is seen as an important challenge [Bar-Yam et al., 2013]. For understanding complex systems and towards an effort for quantifying complex systems, information theory, a science of quantifying information [Chen and Janicke, 2010], is used in this dissertation. Figure 79 illustrates the roots of complex systems science and the concepts on which this dissertation is developed.

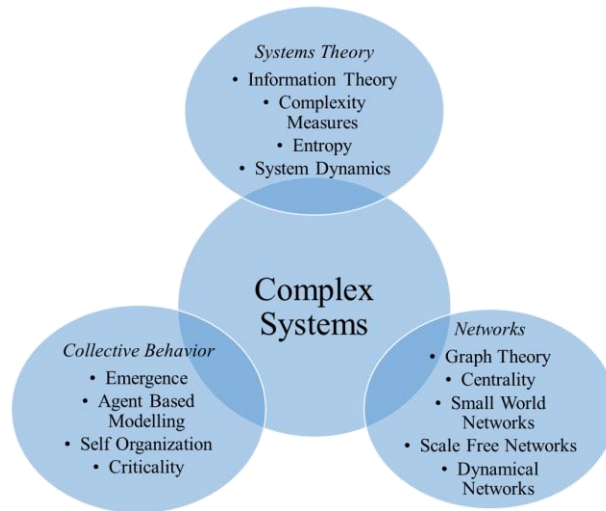


Figure 79: Principles of Complex System Science used in this Dissertation

Fundamental concepts from information theory were explored and applied in specific scenarios classified as case studies based on a strict assumption that, in a system, wherever a link or an interface is observed among the sub-systems, there exists a flow/ exchange of information from one to the other.

Based on the premises of systems engineering and the principles of systems thinking, this research presents interdisciplinary case studies for examining and analyzing complexity of a considered system in terms of its constituent sub systems. The three main concepts underlying the systems thinking principles addressed for the case studies used in this research are:

1, Interaction: Characteristics of systems can be mapped to the capabilities and behavior of their sub systems and their interactions among the sub-systems where, they define the interaction and influence of the system in consideration with other external systems and sub systems [SEBoK, 2014].

2, Relations: A systems is characterized by the interconnections among its sub systems, giving rise to a set of identifiable relations defining a network [SEBoK, 2014].

3, Network: Networks form basic building blocks of systems and sub systems giving rise to their togetherness, connections, and dynamic interactions among the constituents of complex systems [SEBoK, 2014].

The case studies considered in this research are mapped on to a complexity classification graph illustrating their placement across Information and Rigidity. To examine the case studies using information theoretic approach, we map the similarity and the differences of the case studies used to a general communication channel. A typical communication channel consists of a source that sends information, a channel that acts as a medium to transmit the information and a destination which receives the transmitted information. Considering the case studies explained in this dissertation document: In Software Based Control Flow Graphs, the nodes represented as software code components act as both transmitter and receiver, and the arcs between the nodes represent the flow of information between the code components. In the case of Classroom Structures, either teachers, students or peers act as source transmitting information (knowledge), students act as destination receiving and processing information (knowledge) and the type of classroom setting (traditional, peer-based, flipped) considered acts as a channel for relaying information. In terms of complex networks, network nodes and actors behave as both source and destination constantly transmitting and receiving information, whereas the edges connecting the nodes of a complex network act as the medium of information transfer.

The application of the concepts from information theory to each of the identified case studies, helped in understanding, at a system and sub system level, the interdependencies among the considered system constituents along with leading to system analysis insights which led to the development of tailored entropy based case study specific metrics.

Case Studies Revisited

Entropy Applied to Software Based Control Flow Graphs

Complexity (subjectively i.e. human based) in software refers to the interaction among the program and the programmer working on developing a programmable task. Measuring complexity in software development helps in mitigation reoccurring software maintenance costs [Kearney et al., 1986]. In the analyzed case study, information entropy and its application towards measuring software complexity are explored, along with the formulation of an information entropy based complexity measure that considers logical decision-making, processes, and software statement interaction patterns in control flow graphs mapped from actual software code. To broaden the application of the proposed metric, the execution times of nodes in the control flow graphs are also incorporated. Further, the metric is evaluated against eight different axioms that a software complexity measure should satisfy. Based on the conducted analysis, a positive correlation was observed for FORTRAN based CFG's and for CFG's based on Matlab code programmed to perform basic linear algebraic computations. The evaluation of the metric against Weyukers criteria helped to support the metric's validity for use.

Finally, key properties of the developed metric include its sensitivity to how code components interact, sensitivity to program syntax, correlation with the size of software and its corresponding complexity measure. However, a key draw back observed is that it is unable to distinguish in which a given program appears. The metric formulated based on the said analysis can be used by a programmer in the design phase of a software code to understand how complex is a code that is being developed so that, in the later stages of implementation, testing and maintenance it would take less time and effort to debug a code when required.

To establish the proposed metric in the body of knowledge, further exploration of the metric is required to understand its applicability in answering how complexity varies with the size of software, if software complexity increases over time, how complexity changes in piece of code written today when compared to that were written previously in another context.

Application of Information Entropy for Classroom Structural Assessment

Identifying and applying new and innovative classroom-learning techniques in academia, for the purpose of keeping students motivated, is currently a top priority. Assessment techniques are needed which can be applied to quantify the theoretical effectiveness of communication for any given classroom information dissemination structure. This case study explores the application of information entropy to enable the assessment of various classroom structures. Tailored classroom setting specific metrics have been formulated considering, the information gained by a motivated individual (student) in a given classroom structural setting throughout out a session to be a function of knowledge (it acts as weighing function that is calculated as a ratio of text entropy from professors, lecturers, peers and student) , number of inputs, and number of outputs. Knowledge here acts as a weighing function and the number of inputs and outputs depends upon the interactions made by the individual.

Conceptually developed based on several interdependent factors such as interaction patterns among actors of a classroom, knowledge of the actors, number of actors and types of classroom interfaces; proposed metrics will acts as a framework that portrays a given structural classroom setting as an aggregative result of individual factors of its stakeholders. To successfully realize the study based on this framework, data gathering will include observing student communication and their interaction patterns in a given setting. The proposed metrics doesn't stress upon the

intended meaning understood by a student but however, we look to understand if a student is able to comprehend the vocabulary. Additionally, data can be gathered based on a pre-specified set of key words relevant to the topic covered. The Text Entropy can then be calculated based on student understanding of the topics in the form of text recordings that capture whether students are adopting the vocabularies of instructors.

Information Theory applied to Complex Networks

This case study sets by exploring what complexity is, in complex networks. Social networks, a special case of complex networks are explored. To understand how information grows with complexity in networks, the concept of transfer entropy is used to understand how information cascades through an emerging network. Also, an attempt to explore and understand the structural importance of the nodes in a complex network on information processing over the network structural emergence is presented. Following the concept of entropy, a network entropy measure incorporating the diversity of network node dynamic interactions is suggested to reflect upon network order. Examples of preferential attachment network, small world network, and a team based network are used to illustrate the application of information theory concepts. Furthermore, a sample social network of geographically separated student interaction (UTEP and Drexel University students enrolled in Green Energy Manufacturing class) is extracted from Facebook to illustrate the applicability transfer entropy and dynamic entropy measures.

To further explore complex networks, a sample network is generated using an agent based simulation model to simulate self-organization and criticality. Comparing a network to a Bak-Tang-Wiesenfeld (BTW) sand pile model, the concepts of information theory, when applied to the generated network data, helped to identify that the bidirectional flow of information among

the nodes in the generated network is almost equal throughout its dynamics, implying that the critical state of a network to be a highly desirable setting. This concurs with several network studies where networks at critical states have high functional capability. Also, to explore the structural implication of a network on its nodes information processing capabilities preferential attachment and small world networks were generated where it was observed that in preferential attachment network as the average degree of a node increases the total information available by the node increases where as in small world networks the information available by a node gradually decreases as the average degree of the node increases. Considering the generated network model portraying network dynamics towards critical state, it is observed that all the nodes on average process the same information across the network moving from an initial disordered state to a highly ordered state. On the other side, in the nodes of the network where external perturbations were introduced it is observed that there is a sudden increase in the information available which later slowly stabilizes illustrating the shift in network towards criticality, processing the same amount of information on an average.

Finally, an agent based model tool incorporating expected utility theory is developed which enables in identifying the state of a network associated with the highest expected utility value when calculated strictly based on objective utility and objective probability values for identifying a better network state over its emergence.

Refer to Table 33 for a summarized representation of how information theoretic concepts and their applications are relevant in understanding the case studies used.

Table 33: information theoretic concepts and their application towards complex systems

Concepts and Measures from Information Theory	Complex System Characterization
<i>SOFTWARE SYSTEMS</i>	
Entropy	<ul style="list-style-type: none"> Measuring complexity of a software code developed, to be used in the design phase of a software code to understand on how the complex code is being developed. We show how Entropy can be used to measure complexity of a software code when converted to a control flow graph using software code characteristics.
<i>CLASSROOM STRUCTURES</i>	
Entropy	<ul style="list-style-type: none"> Enables in a stakeholder centric assessment of a considered classroom structure. We show how information entropy can be used to assess classroom structures by developing tailored metrics for Tradition, Flipped and peer based classroom settings.
Transfer Entropy	<ul style="list-style-type: none"> Enables to understand the implication of a given classroom interaction pattern on the information processing ability of stakeholders. Enables to calculate the total information available of a given classroom structure over its occurrence. Helps to identify a student's potential to understand based on the location in a classroom.
<i>COMPLEX NETWORKS</i>	
Entropy	<ul style="list-style-type: none"> Helps to understand the diversity of network node interactions Helps to identify network structural emergence Helps to understand the context of information spread in a network over its emergence.
Transfer Entropy	<ul style="list-style-type: none"> Helps to identify the structural implication of nodes in a complex network on their information processing ability. Helps to identify the information processing ability of a network. Helps to identify the state of criticality in a complex network over its emergence. Helps to quantify how much information is needed to describe a network of a given scale.

Research Questions Explored

This research surrounding the case studies is framed along the application of information theory concepts with a goal to better understand and assess complex systems. Mentioned below are the insights drawn from the explored case studies across different complexity classifications in answering the following questions.

How Complex is a Complex System?

Based on the fact that not all complex systems have same characteristics and behave in a similar fashion, it is indeed not possible to provide a generalized metric to quantify a complex system. However, using information theory concepts tailored to a specific system being considered helps to answer and quantify the complexity, based on a set of assumptions. In such a case, a thorough understanding of the considered systems sub-systems and their interactions is required. Using the concept of entropy, complex systems can be mapped as a network of interactions, and the characteristics of the complex systems can be quantified, thereby setting a foundation for understanding how complex a given complex system is. This shows that a first step towards addressing this question is to use the concept of entropy which helps to quantify the complexity of a given complex system.

Complex systems are highly non-linear where continual emergence is exhibited. Now, with using the concept of entropy to quantify a complex system the second question to be addressed is on

Can Information Entropy help in identifying Emergence in a Complex System?

Using the concept of entropy the complexity of a given complex system can be quantified based on a set of assumptions. For example, when a complex system is mapped as a network,

information entropy can be used to quantify the network at a given snapshot based on the observed sub-systems interaction patterns. Considering the network dynamics, at every instant the network emerges with either new nodes introduced into the network or rewiring the connections among the nodes of the networks.

Provided that the data on node interaction patterns at every instance of the network is available, entropy based measures can detect the precise time instance where the network is exposed to external stimuli. This is because entropy can be seen as a measure of surprise where in with sudden external stimuli to the network a rapid change in the tailored entropy based measure can be detected when observed over the system dynamics.

Addressing the structural dynamics of complex systems, the third question to be addressed is on if there is an influence of system emergence on the information processing capabilities of its sub systems.

What is the implication of structural change in a complex network on its information processing capabilities?

With a complex network mapped as a network of interactions over its evolutionary dynamics, the concept of Transfer Entropy can be used as a tool to quantify how much information is processed at each and every component. From the examples used in this research, it is seen that in networks that follow preferential attachment phenomenon, the total information available by a node increases with an increase in its number of interactions, and, in the networks that portray small world phenomenon, the total information available by a component decreases as the average number of connections increases.

The answers provided to the questions above are strictly based on the observations and insights drawn from the case studies explored in this research. Though the insights gained directly may

not address complex system characteristics from every possible scenario, but set a foundation for a fact based theoretical approach framework for explaining, understanding and analyzing complex systems using information theory concepts.

Research Contributions

In this dissertation, the findings on the use of information theory in the different case studies (please see figure 77) analyzed are reported. Based on the insights drawn upon the case studies explored, the contributions to complex systems body of knowledge are:

- Examination of information theory and its application towards understanding complex systems
- Presented case study based theoretical frameworks portraying how major concepts of information theory measures can facilitate in understanding various aspects of a complex systems (see table 33)
- Application of network based approach to analyze complex systems
- Established tailored information entropy based metrics for assessing Software based systems, Classroom structures and Complex Networks
- Developed a tool using an agent based modeling platform (NetLogo) to identify effective classroom based interaction patterns
- Translated the application of complex networks to the subjective domain by incorporating expected utility theory and Prospect Theory
- Established a link between network analysis and data mining through the Transfer Entropy measure

References

1. Adams, Heather, David Adams, and Korey Paul. "Information Processing Theory." *PSYSC613*. N.p., n.d. Web. 09 Feb. 2015.
<<http://psysc613.wikispaces.com/Information+Processing+Theory>>.
2. Al-Radaideh, Qasem A., Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. "Mining student data using decision trees." *International Arab Conference on Information Technology (ACIT'2006)*, Yarmouk University, Jordan. 2006.
3. Bansiya, Jagdish, Carl Davis, and Letha Etzkorn. "An entropy-based complexity measure for object-oriented designs." *Theory and Practice of Object Systems* 5.2 (1999): 111-118.
4. Baranger, Michel. "Chaos, complexity, and entropy." *New England Complex Systems Institute, Cambridge* (2000).
5. Benjamin, S. Blanchard, and Walter J. Fabrycky. "Systems engineering and analysis." (1998).
6. Ben-Naim, Arieh. *Entropy Demystified: The Second Law Reduced to Plain Common Sense with Seven Simulated Games*. World Scientific, 2008.
7. Berlinger, Eli. "An information theory based complexity measure." *Proceedings of the May 19-22, 1980, national computer conference*. ACM, 1980.
8. Berrett, Dan. "How 'flipping' the classroom can improve the traditional lecture." *The chronicle of higher education* 12 (2012).
9. Bickel, J. Eric. "Scoring rules and decision analysis education." *Decision Analysis* 7.4 (2010): 346-357.
10. Bishop, Jacob Lowell, and Matthew A. Verleger. "The flipped classroom: A survey of the research." *ASEE National Conference Proceedings, Atlanta, GA*. 2013.
11. Boccardi N. (2004), "Modeling complex systems", Springer Publ., Berlin.

12. Brown, Ann L. "Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings." *The journal of the learning sciences* 2.2 (1992): 141-178.
13. Browning, Tyson R. "Applying the design structure matrix to system decomposition and integration problems: a review and new directions." *Engineering Management, IEEE Transactions on* 48.3 (2001): 292-306.
14. Casey, Gerard, and Aidan Moran. "The computational metaphor and cognitive psychology." *The Irish Journal of Psychology* 10.2 (1989): 143-161.
15. Chappuis, Stephen, and Richard J. Stiggins. "Classroom assessment for learning." *Educational Leadership* 60.1 (2002): 40-44.
16. Chaturvedi, K. K., et al. "Predicting the complexity of code changes using entropy based measures." *International Journal of System Assurance Engineering and Management* 5.2 (2014): 155-164.
17. Coetzee, D., et al. "Structuring Interactions for Large-Scale Synchronous Peer Learning." *CSCW: ACM Conference on Computer Supported Collaborative Work*. 2015.
18. Cohen, Elizabeth G. "Sociology and the classroom: Setting the conditions for teacher-student interaction." *Review of Educational Research* (1972): 441-452.
19. Cover, Thomas M., and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
20. Creasy, Kim, and Xin Liang. "Classroom assessment in web-based instructional environment: Instructors' experience." *Online Submission* 9.7 (2004).
21. Davies, J., & Graff, M. (2005). "Performance in e-learning: Online participation and student grades." *British Journal of Educational Technology*, 36(4), 657–663.
22. Davies, Paul. *The fifth miracle: The search for the origin and meaning of life*. Simon and Schuster, 2000.

23. E Standard Methodology for Reliability Prediction and Assessment for Electronic Systems and Equipment, IEEE Standard, 1413-1998.
24. Eisen, Mary-Jane. "Peer-Based Learning: A New-Old Alternative to Professional Development." *Adult Learning*, Vol. 12, No. 1, Winter 2001
25. Eisner, Howard. *Managing complex systems: thinking outside the box*. Vol. 34. John Wiley & Sons, 2011.
26. Ellis, Rod. "The Interaction Hypothesis: A Critical Evaluation." (1991).
27. Entropy (Information Theory) "*Wikipedia, The Free Encyclopedia*". Wikimedia Foundation, Inc. 22 July 2004. Web. 1 March. 2015.
28. Entropy in Statistical Mechanics. Lecture Notes. URL<<http://tfy.tkk.fi/kurssit/Tfy-3.363/lectures/lecture03.pdf>>> (Retrieved on 2015-04-1)
29. Entropy "*Wikipedia, The Free Encyclopedia*". Wikimedia Foundation, Inc. 22 July 2004. Web. 1 March. 2015.
30. Eppinger, Steven D., and Tyson R. Browning. *Design structure matrix methods and applications*. MIT Press, 2012.
31. Fitzsimmons, Ann, and Tom Love. "A review and evaluation of software science." *ACM Computing Surveys (CSUR)* 10.1 (1978): 3-18.
32. Freedman, R. A. *Sears and Zemansky's university physics: with modern physics*. Addison Wesley, San Fransisco, USA, 2003.
33. Gibson, David. "Network-based assessment in education." *Contemporary issues in technology and teacher education* 3.3 (2003): 310-323.
34. Haglund, Jesper, Fredrik Jeppsson, and Helge Strömdahl. "Different senses of entropy—implications for education." *Entropy* 12.3 (2010): 490-515.
35. Hamada, Mohamed. "Web-based tools for active learning in information theory." *ACM SIGCSE Bulletin* 39.1 (2007): 60-64.

36. Hamer, Peter G., and Gillian D. Frewin. "MH Halstead's Software Science-a critical examination." *Proceedings of the 6th international conference on Software engineering*. IEEE Computer Society Press, 1982.
37. Hannum, Wallace H., and Leslie J. Briggs. "How Does Instructional Systems Design Differ from Traditional Instruction?." *Educational Technology* 22.1 (1982): 9-14.
38. Harrison, Warren. "An entropy-based measure of software complexity." *Software Engineering, IEEE Transactions on* 18.11 (1992): 1025-1029.
39. He, Yi, Sandra Swenson, and Nathan Lents. "Online video tutorials increase learning of difficult concepts in an undergraduate analytical chemistry course." *Journal of Chemical Education* 89.9 (2012): 1128-1132.
40. Henriksson, Anders. *Kemi. Kurs A*. 1st edition; Gleerup:Malmo, Sweden, 2001.
41. Henry, Sallie, and Dennis Kafura. "Software structure metrics based on information flow." *Software Engineering, IEEE Transactions on* 5 (1981): 510-518.
42. Herreid, Clyde Freeman, and Nancy A. Schiller. "Case studies and the flipped classroom." *Journal of College Science Teaching* 42.5 (2013): 62-66.
43. Holland, John H., (2006). Studying Complex Adaptive Systems, *Journal of Systems Science and Complexity* 19 (1): 1-8. < URL: ><http://hdl.handle.net/2027.42/41486>>
44. IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software, IEEE Standard, 982.2-1988.
45. IEEE Standard Dictionary of Measures to Produce Reliable Software, IEEE Standard, 982.1-1988.
46. Information Entropy vs Thermodynamic Entropy. "Markov Chain Universe". N.p., n.d. Web. 01 Apr. 2015.
47. Johnson, John J., Andreas Tolk, and Andres Sousa-Poza. "A Theory of Emergence and Entropy in Systems of Systems." *Procedia Computer Science* 20 (2013): 283-289.

48. Jung, Woo-Sung, et al. "An Entropy-Based Complexity Measure for Web Applications Using Structural Information." *J. Inf. Sci. Eng.* 27.2 (2011): 595-619.
49. Kakkonen, Tuomo, and Erkki Sutinen. "Automatic assessment of the content of essays based on course materials." *Information Technology: Research and Education, 2004. ITRE 2004. 2nd International Conference on.* IEEE, 2004.
50. Kang, Changhui. "Academic interactions among classroom peers: a cross-country comparison using TIMSS." *Applied Economics* 39.12 (2007): 1531-1544.
51. Kang, Hyun Gook, and Poong Hyun Seong. "Information theoretic approach to man-machine interface complexity evaluation." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 31.3 (2001): 163-171.
52. Kastens, Kim A., et al. "How geoscientists think and learn." *Eos, Transactions American Geophysical Union* 90.31 (2009): 265-266.
53. Kullback, Solomon. *Information theory and statistics*. Courier Corporation, 1997.
54. Lachman. R., Lachman, J. L., & Butterfield, E. C. (1979). *Cognitive Psychology and Information Processing: An Introduction*. Hillsdale, NJ: Lawrence Erlbaum.
55. Lage, Maureen J., Glenn J. Platt, and Michael Treglia. "Inverting the classroom: A gateway to creating an inclusive learning environment." *The Journal of Economic Education* 31.1 (2000): 30-43.
56. Lajis, Adidah, and Normaziah Abdul Aziz. "Assessing Learner's Understanding Through Node-Link Analysis: An Information Theory Approach." *Proceedings IEEE-International Symposium of Information Technology*. 2008.
57. Lajis, Adidah, and Normaziah Abdul Aziz. "Assessment of Learner's Understanding: An Experimental Result." *Computer Technology and Development, 2009. ICCTD'09. International Conference on.* Vol. 1. IEEE, 2009.

58. Lajis, Adidah, and Normaziah Abdul Aziz. "NL Scoring Technique for the Assessment of Learners' Understanding." *Computer Research and Development, 2010 Second International Conference on*. IEEE, 2010.
59. Liu, Chao-Lin. "Using mutual information for adaptive item comparison and student assessment." *Educational Technology & Society* 8.4 (2005): 100-119.
60. Liu, Feifan, et al. "Unsupervised approaches for automatic keyword extraction using meeting transcripts." *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, 2009.
61. Machta, J. "Entropy, information, and computation." *American Journal of Physics* 67.12 (1999): 1074-1077.
62. Magee, Christopher, and Olivier de Weck. "Complex system classification." (2004).
63. Maxwell, Joseph A. *Qualitative research design: An interactive approach: An interactive approach*. Vol. 41. Sage, 2012.
64. McCabe, Thomas J. "A complexity measure." *Software Engineering, IEEE Transactions on* 4 (1976): 308-320.
65. McCall, Jim A., Paul K. Richards, and Gene F. Walters. *Factors in software quality*. General Electric, National Technical Information Service., 1977.
66. McLaughlin, Jacqueline E., et al. "The flipped classroom: a course redesign to foster learning and engagement in a health professions school." *Academic Medicine* 89.2 (2014): 236-243.
67. Mills, Harlan D. "The management of software engineering, Part I: Principles of software engineering." *IBM Systems Journal* 19.4 (1980): 414-420.
68. Ni, Anna Ya. "Comparing the Effectiveness of Classroom and Online Learning: Teaching Research Methods." *JPAE Journal of Public Affairs Education Volume 19 number 2* (2013): 199-215.

69. Oviedo, Enrique I. "Control flow, data flow, and program complexity." *Proceedings of IEEE COMPSAC*. 1980.
70. Percivall, George S. "System architecture for evolutionary system development." *INCOSE International Symposium*. Vol. 4. No. 1. 1994.
71. Pimmler, Thomas Udo, and Steven D. Eppinger. *Integration analysis of product decompositions*. Cambridge, MA: Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1994.
72. Proctor, Robert W., and Trisha Van Zandt. *Human factors in simple and complex systems*. CRC press, 2008.
73. Relan, Anju, and Bijan B. Gillani. "Web-based instruction and the traditional classroom: Similarities and differences." *Khan* 62 (1997): 41-46.
74. Roca, Jose Luis. "An entropy-based method for computing software structural complexity." *Microelectronics Reliability* 36.5 (1996): 609-620.
75. Rosen, Robert. "COMPLEXITY AS A SYSTEM PROPERTY†." *International Journal Of General System* 3.4 (1977): 227-232.
76. Ruddick, K. W. Improving chemical education from high school to college using a more hands-on approach. Unpublished doctoral dissertation, University of Memphis. 2012
77. Schleicher, Andreas, and Claudia Tamassia. *Measuring Student Knowledge and Skills: The PISA 2000 Assessment of Reading, Mathematical and Scientific Literacy. Education and Skills*. 2000.
78. Schmidt III, Robert, Jason Deamer, and Simon Austin. "Understanding adaptability through layer dependencies." *Proceedings of the 18th International Conference on Engineering Design (ICED11)*, Vol. 10. 2011.
79. Selvarani, R., et al. "Software Metrics Evaluation Based on Entropy." *arXiv preprint arXiv:1001.3473* (2010).
80. Sethna, James P. *Statistical mechanics: entropy, order parameters, and complexity*. Vol. 14. New York: Oxford University Press, 2006.

81. Shannon, Claude E., and Warren Weaver. "The mathematical theory of information." (1949).
82. Shannon, Claude Elwood. "A mathematical theory of communication." *ACM SIGMOBILE Mobile Computing and Communications Review* 5.1 (2001): 3-55.
83. Snider, Greg. "Measuring the entropy of large software systems." *HP Laboratories Palo Alto, Tech. Rep* (2001).
84. Snow, Erica L., et al. "Entropy: A Stealth Measure of Agency in Learning Environments."
85. Software Process Improvement Capability Determination (SPICE) - Software Process Assessment, ISO Standard, 15504:1998.
86. Standard, Military. "MIL-STD 499A." *Systems Engineering* (1974).
87. Strange, Bryan A., et al. "Information theory, novelty and hippocampal responses: unpredicted or unpredictable?." *Neural Networks* 18.3 (2005): 225-230.
88. Suh, Eun Suk, et al. "Technology infusion for complex systems: a framework and case study." *Systems Engineering* 13.2 (2010): 186-203.
89. Tan, Wei, and Shuqing Zhou. "Evaluation of the innovation ability of engineering students based on entropy theory." *World Transactions on Engineering and Technology Education* 12.2 (2014): 171-175.
90. Thebeau, Ronnie Emile. Knowledge management of system interfaces and interactions from product development processes. Diss. Massachusetts Institute of Technology, 2001.
91. U-Print® SE and U-Print® SE plus Personal 3D Printers' Introductory Information Manual. ©Stratasys, Inc., 2011.
92. U-Print® SE and U-Print® SE plus Personal 3D Printers' Printer Assembly Instructions. ©Stratasys, Inc., 2011.
93. Van Vliet, Hans, Hans Van Vliet, and J. C. Van Vliet. Software engineering: principles and practice. Vol. 3. Wiley, 1993.
94. VanLehn, Kurt, and Joel Martin. "Evaluation of an assessment system based on Bayesian student modeling." *International Journal of Artificial Intelligence in Education* 8.2 (1998): 179-221.

95. Venkatachalam, Saranya, N. Sairam, and B. Srinivasan. "Code Optimization Using Graph Mining." *Research Journal of Applied Sciences, Engineering and Technology* 4(19) (2012): 3618-3622.
96. Vetromille-Castro, Rafael. "Social interactive entropy and interaction in the language teacher education classroom." *Revista Brasileira de Linguística Aplicada* 13.2 (2013): 625-641.
97. Victor S. Adamchik. "Algorithmic Complexity." School of Computer Science, Carnegie Mellon University (2009). <URL: <http://www.cs.cmu.edu/~adamchik/15121/lectures/Algorithmic%20Complexity/complexity.html>>
98. Watson, Arthur H., Thomas J. McCabe, and Dolores R. Wallace. "Structured testing: A testing methodology using the cyclomatic complexity metric." NIST special Publication 500.235 (1996): 1-114.
99. Weaver, Warren (1948). Science and Complexity. *American Scientist* 36: 536 (Retrieved on 2007-11-21.)
100. Web.mit.edu, (2014). MATLAB Teaching Codes. [online] Available at: <http://web.mit.edu/18.06/www/Course-Info/Tcodes.html> [Accessed 14 Jul. 2014]
101. Weyuker, Elaine J. "Evaluating software complexity measures." *Software Engineering, IEEE Transactions on* 14.9 (1988): 1357-1365.
102. Weyuker, Elaine J. "Evaluating software complexity measures." *Software Engineering, IEEE Transactions on* 14.9 (1988): 1357-1365.
103. Woodfield, Scott N. "An experiment on unit increase in problem complexity." *Software Engineering, IEEE Transactions on* 2 (1979): 76-79.
104. Yassine, Ali A. "An introduction to modeling and analyzing complex product development processes using the design structure matrix (DSM) method." *Urbana* 51.9 (2004): 1-17.
105. Yates, Roy D., and David J. Goodman. *Probability and stochastic processes*. New York: Wiley, 1999.

106. Zappe, Sarah, et al. "Flipping" the classroom to explore active learning in a large undergraduate course." *American Society for Engineering Education*. American Society for Engineering Education, 2009.
107. Smith, Eric D. "Gödel's Incompleteness and Consistency Theorems Elucidated with Principles of Abstraction Levels, Complementarity, and Self-Reference." (2013).
108. Eisner, Howard. *Managing complex systems: thinking outside the box*. Vol. 34. John Wiley & Sons, 2011.
109. Sayama, Hiroki. "Introduction to the modeling and analysis of complex systems." *State University of New York at Geneseo: Open SUNY Textbooks, Milne Library* (2015).
110. Chen, Min, and Heike Jaenicke. "An information-theoretic framework for visualization." *Visualization and Computer Graphics, IEEE Transactions on* 16.6 (2010): 1206-1215.
111. Bar-Yam, Yavni, Dion Harmon, and Yaneer Bar-Yam. "Computationally tractable pairwise complexity profile." *Complexity* 18.5 (2013): 20-27.
112. Pyster, Art, et al. "Guide to the Systems Engineering Body of Knowledge (SEBoK) v. 1.3." *Guide to the Systems Engineering Body of Knowledge (SEBoK)* (2014).
113. Kearney, Joseph P., et al. "Software complexity measurement." *Communications of the ACM* 29.11 (1986): 1044-1050
114. Newman, Mark. *Networks: an introduction*. OUP Oxford, 2010.
115. Snowden, David J., and Mary E. Boone. "A leader's framework for decision making." *harvard business review* 85.11 (2007): 68.
116. Alderson, David L., and John C. Doyle. "Contrasting views of complexity and their implications for network-centric infrastructures." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 40.4 (2010): 839-852.
117. Poli, Roberto. "A note on the difference between complicated and complex social systems." *Cadum* 2.1 (2013): 142.

118. Rouse, William B. *Modeling and Visualization of Complex Systems and Enterprises: Explorations of Physical, Human, Economic, and Social Phenomena*. John Wiley & Sons, 2015.
119. Solé, Ricard V., and Sergi Valverde. "Information theory of complex networks: on evolution and architectural constraints." *Complex networks*. Springer Berlin Heidelberg, 2004. 189-207.
120. Barabási, Albert-László. "Scale-free networks: a decade and beyond." *science* 325.5939 (2009): 412.
121. Wilensky, U. (2005). NetLogo Preferential Attachment model.
<http://ccl.northwestern.edu/netlogo/models/PreferentialAttachment>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
122. Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
123. Newman, Mark, Albert-László Barabási, and Duncan J. Watts. *The structure and dynamics of networks*. Princeton University Press, 2006.
124. Mehra, Ajay, and Linton C. Freeman. "The Development of Social Network Analysis: A Study in the Sociology of Science." (2005): 148-151.
125. Huberman, Bernardo A., and Lada A. Adamic. "Information dynamics in the networked world." *Complex networks*. Springer Berlin Heidelberg, 2004. 371-398.
126. Easley, David, and Jon Kleinberg. "Networks, crowds, and markets: Reasoning about a highly connected world." *Significance* 9 (2012): 43-44.
127. Mitchell, Melanie. "Complex systems: Network thinking." *Artificial Intelligence* 170.18 (2006): 1194-1212.
128. Scott, John. *Social network analysis*. Sage, 2012.
129. Freeman, Linton C. "Centrality in social networks conceptual clarification." *Social networks* 1.3 (1978): 215-239.
130. Hanneman, Robert A., and Mark Riddle. "Introduction to social network methods." (2005).

131. Ruhnau, Britta. "Eigenvector-centrality—a node-centrality?." *Social networks* 22.4 (2000): 357-365.
132. Murcio, Roberto, et al. "Urban transfer entropy across scales." *arXiv preprint arXiv:1505.02761* (2015).
133. Schreiber, Thomas. "Measuring information transfer." *Physical review letters* 85.2 (2000): 461.
134. Railsback, Steven F., and Volker Grimm. *Agent-based and individual-based modeling: a practical introduction*. Princeton university press, 2011.
135. Barabási, Albert-László, and Réka Albert. "Emergence of scaling in random networks." *science* 286.5439 (1999): 509-512.
136. Watts, Duncan J., and Steven H. Strogatz. "Collective dynamics of ‘small-world’ networks." *nature* 393.6684 (1998): 440-442.
137. Bakshy, E. and Wilensky, U. (2007). NetLogo Team Assembly model.
<http://ccl.northwestern.edu/netlogo/models/TeamAssembly>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL
138. Guimera, Roger, et al. "Team assembly mechanisms determine collaboration network structure and team performance." *Science* 308.5722 (2005): 697-702.
139. Carrington, Peter J., John Scott, and Stanley Wasserman, eds. *Models and methods in social network analysis*. Vol. 28. Cambridge university press, 2005.
140. Ruane, Regina, Richard Chiou, and Tseng Bill. "Creating an Online Community to engage and support undergraduate engineering students in Green Energy Manufacturing." *Society for Information Technology & Teacher Education International Conference*. Vol. 2015. No. 1. 2015.
141. Doerr, Benjamin, Mahmoud Fouz, and Tobias Friedrich. "Why rumors spread so quickly in social networks." *Communications of the ACM* 55.6 (2012): 70-75.
142. Valverde, Sergi, et al. "Structural determinants of criticality in biological networks." *Frontiers in physiology* 6 (2015).

143. Cajueiro, Daniel O., and Roberto FS Andrade. "Controlling self-organized criticality in complex networks." *The European Physical Journal B* 77.2 (2010): 291-296.
144. Per Bak, T. C., and Kurt Wiesenfeld. "Self-organized criticality: and explanation of 1/f noise." *Phys. Rev. Let* 59 (1987): 381-384.
145. Akundi, Aditya, and Eric Smith. "Understanding Possible Biases and Attributions in Mentoring & Teaching." *UNM Mentoring Institute. Proc. of UNM Mentoring Conference- Impact and Effectiveness of Developmental Relationships, University of New Mexico. N.p., 2013.*
146. Smith, Eric David. "Tradeoff studies and cognitive biases." (2006).
147. Akundi, A., Tseng, T. B., & Smith, E. D., "Social Network Platforms in Educational Settings: A Network Analysis Approach to Analyze Online Student Interactions." *Paper presented at 2016 ASEE Annual Conference and Exposition, New Orleans, LA. (2016, June).*
148. Saramäki, Jari, et al. "Generalizations of the clustering coefficient to weighted complex networks." *Physical Review E* 75.2 (2007): 027105.

Web Pages

1. "Spearman's Correlation." Web. 16 June 2014.
<<http://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf>>.
2. "The Design Structure Matrix (DSM)." *DSMweb.org: Introduction to DSM*. Institute of Product Development , Technische Universität München, n.d. Web. 13 May 2014.
<<http://www.dsmweb.org/en/understand-dsm/technical-dsm-tutorial0/introduction-to-dsm.html>>.

Term Definitions

Classroom Setting: A classroom setting is defined as an environment that is organized to stimulate an effective learning behavior among the participants.

Classroom Structure: A structurally defined implication for organizing the notion of a classroom based on effect of such interactions observed.

Fully connected Ring/Mesh Topology: Refers to a circular organization of the nodes in a network where, each node is connected to each other.

Hierarchy: Organization based on ranking of people above one another.

Logarithm: A logarithm of a number is defined as the value related to an exponent to whom a fixed value is raised to produce the number.

One-to-All: Refers to the transfer of data over many nodes from a single transmission medium. In an educational scenario, One-to-All refers to an instructor imparting knowledge to many students in a classroom.

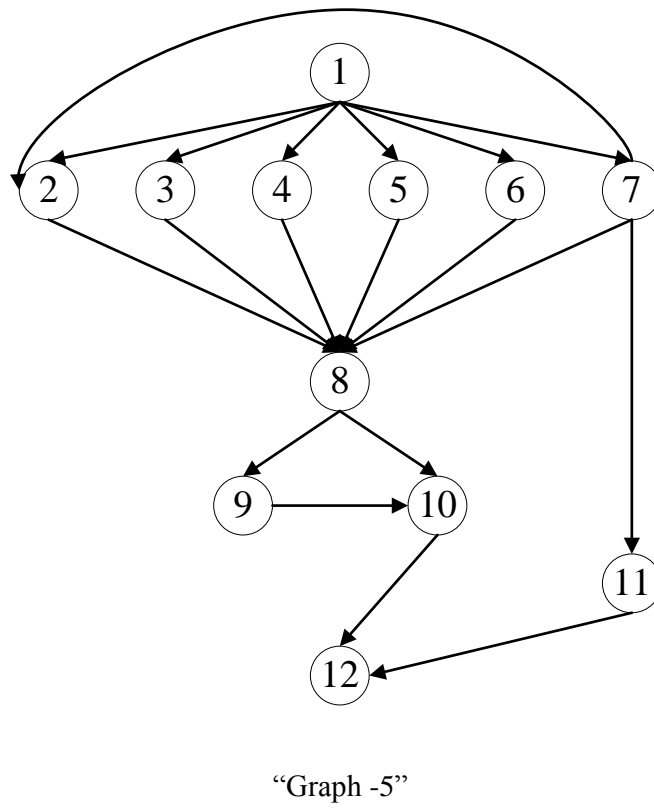
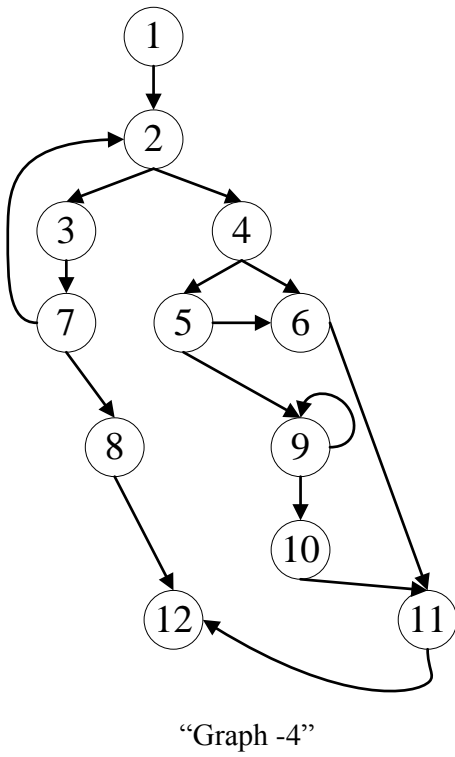
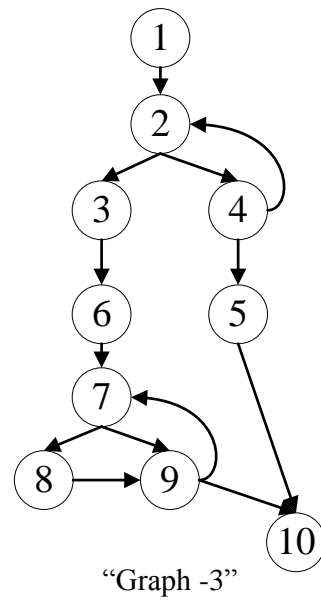
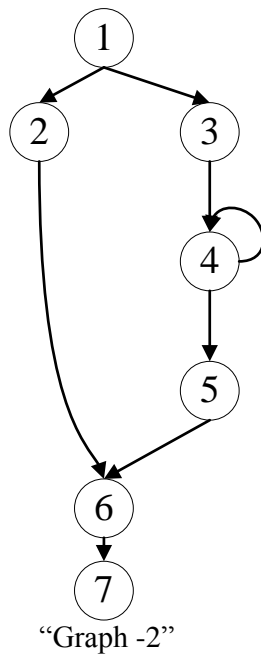
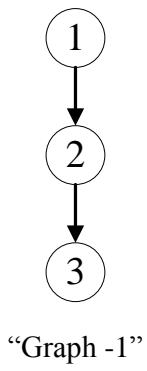
Peers: Peers refer to either senior students mentoring junior students or students from the same year forming partnerships.

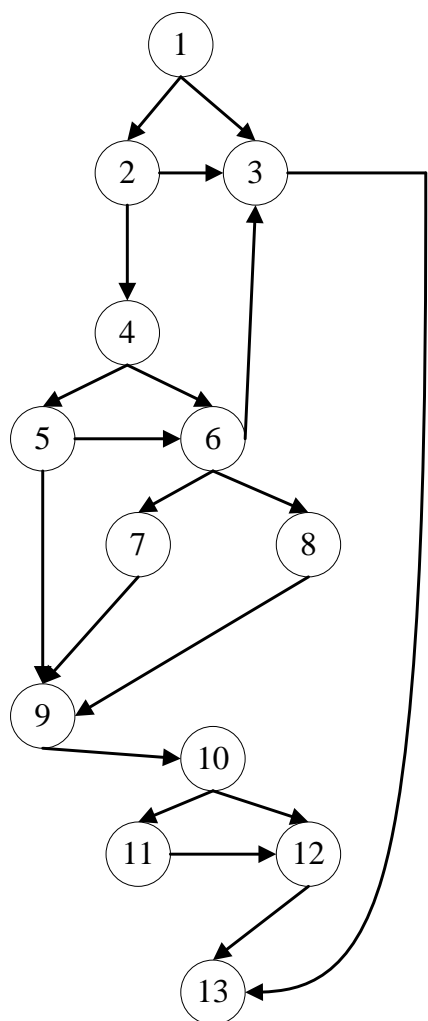
Self-Organizing: Process of achieving order in a network from the coordination of interactions of the nodes.

Star Topology: Refers to connection of every node to a central node.

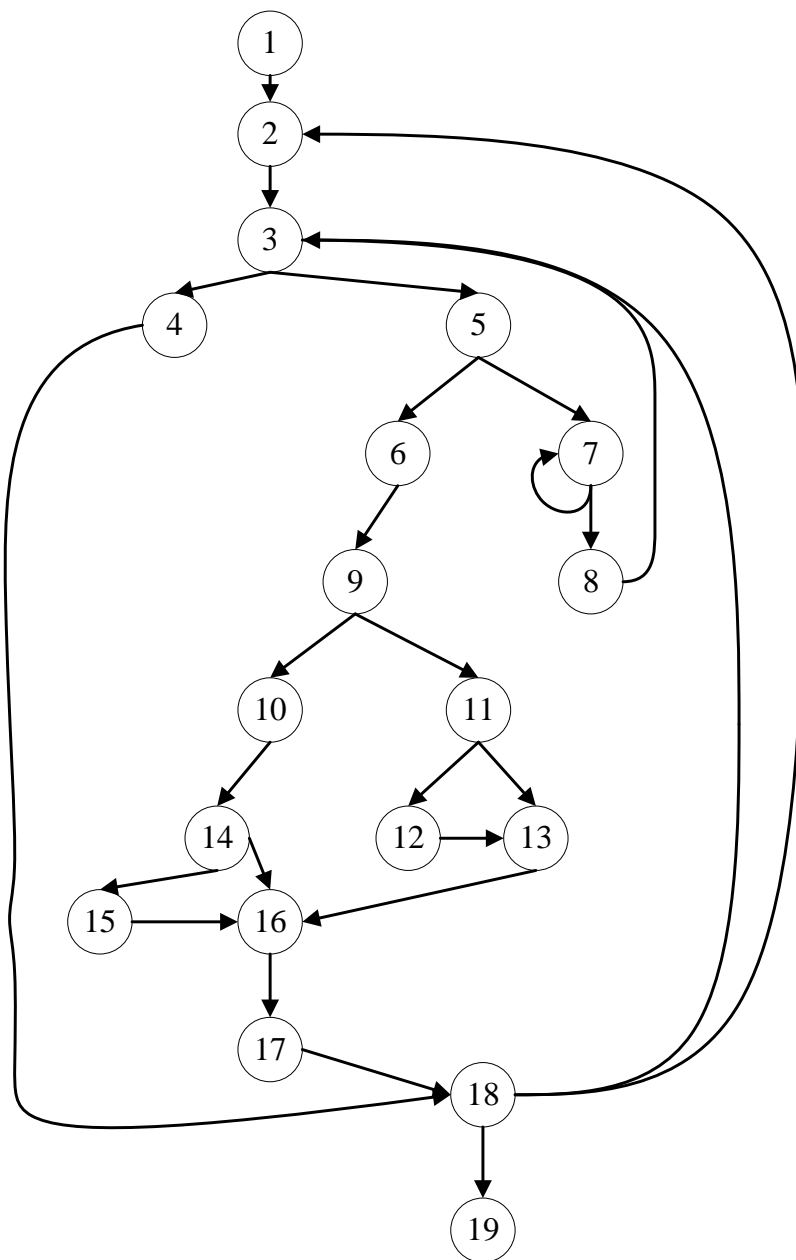
Topology: Arrangement of various elements in a network.

Appendix A

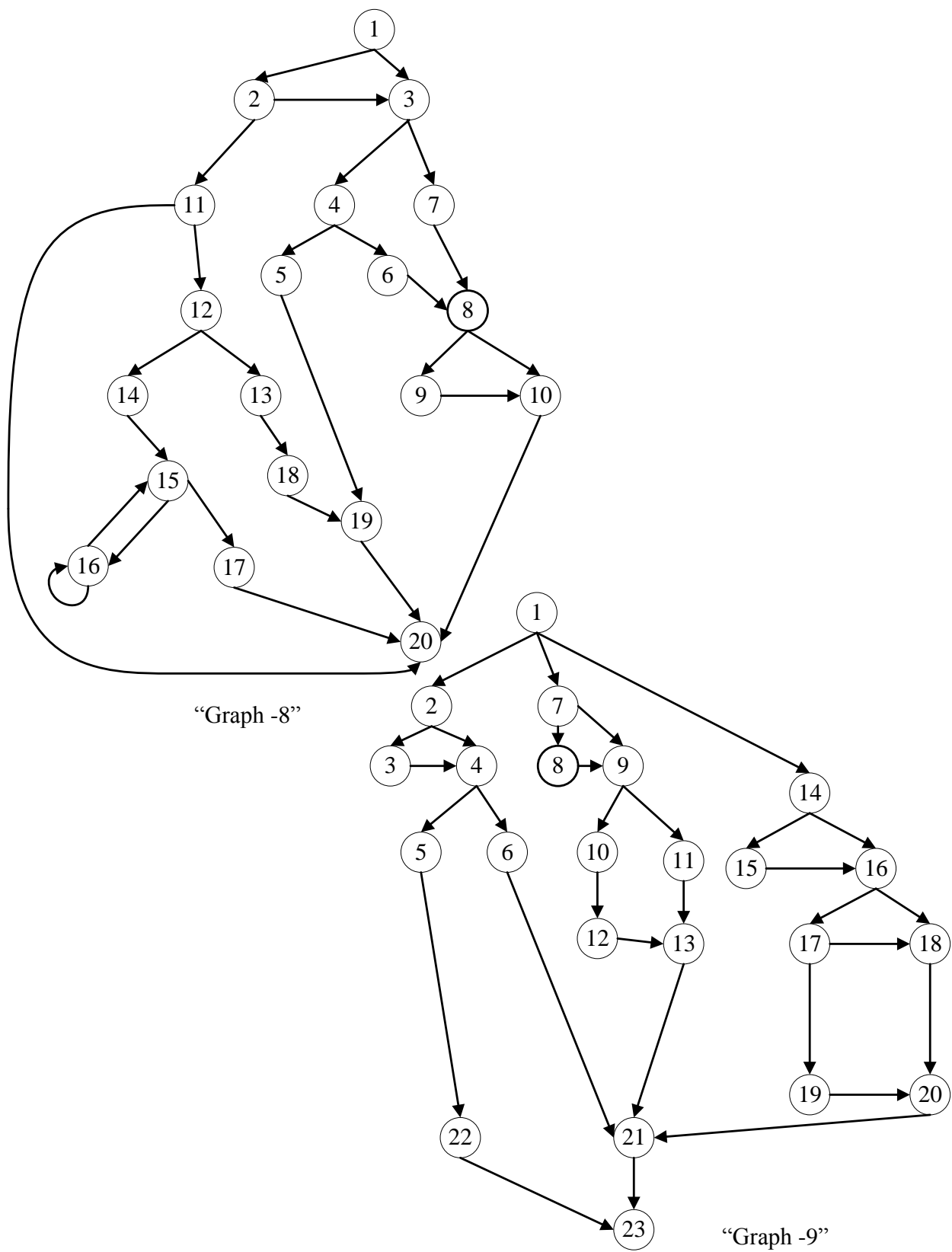


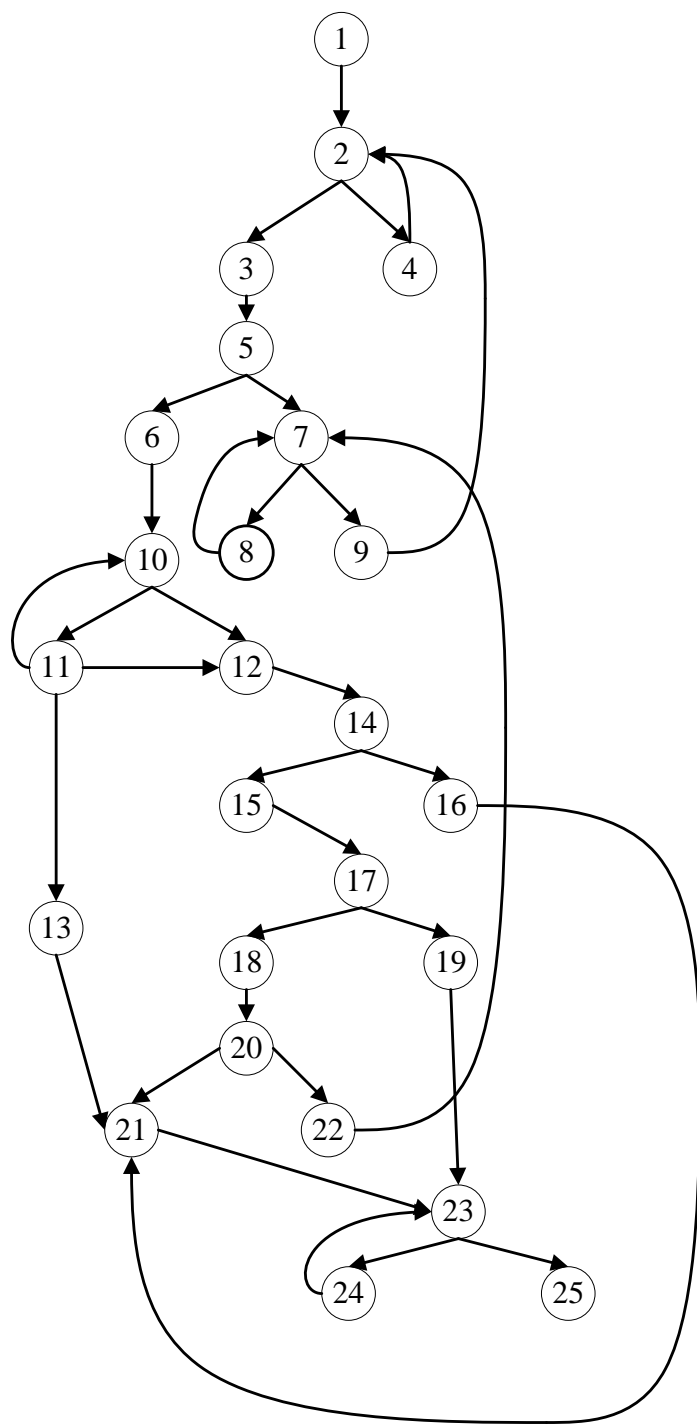


“Graph -6”

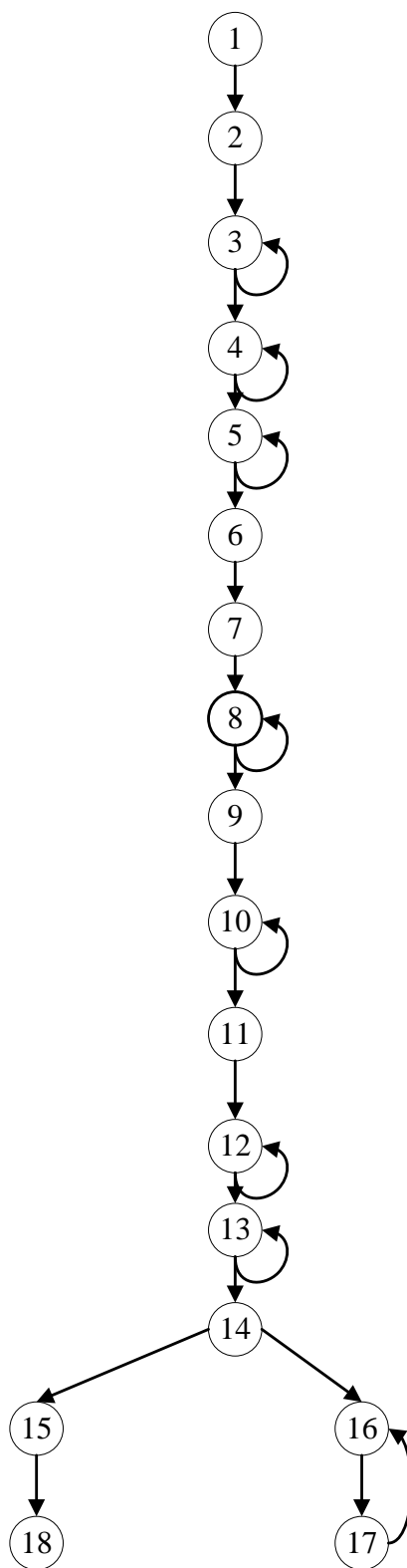


“Graph -7”

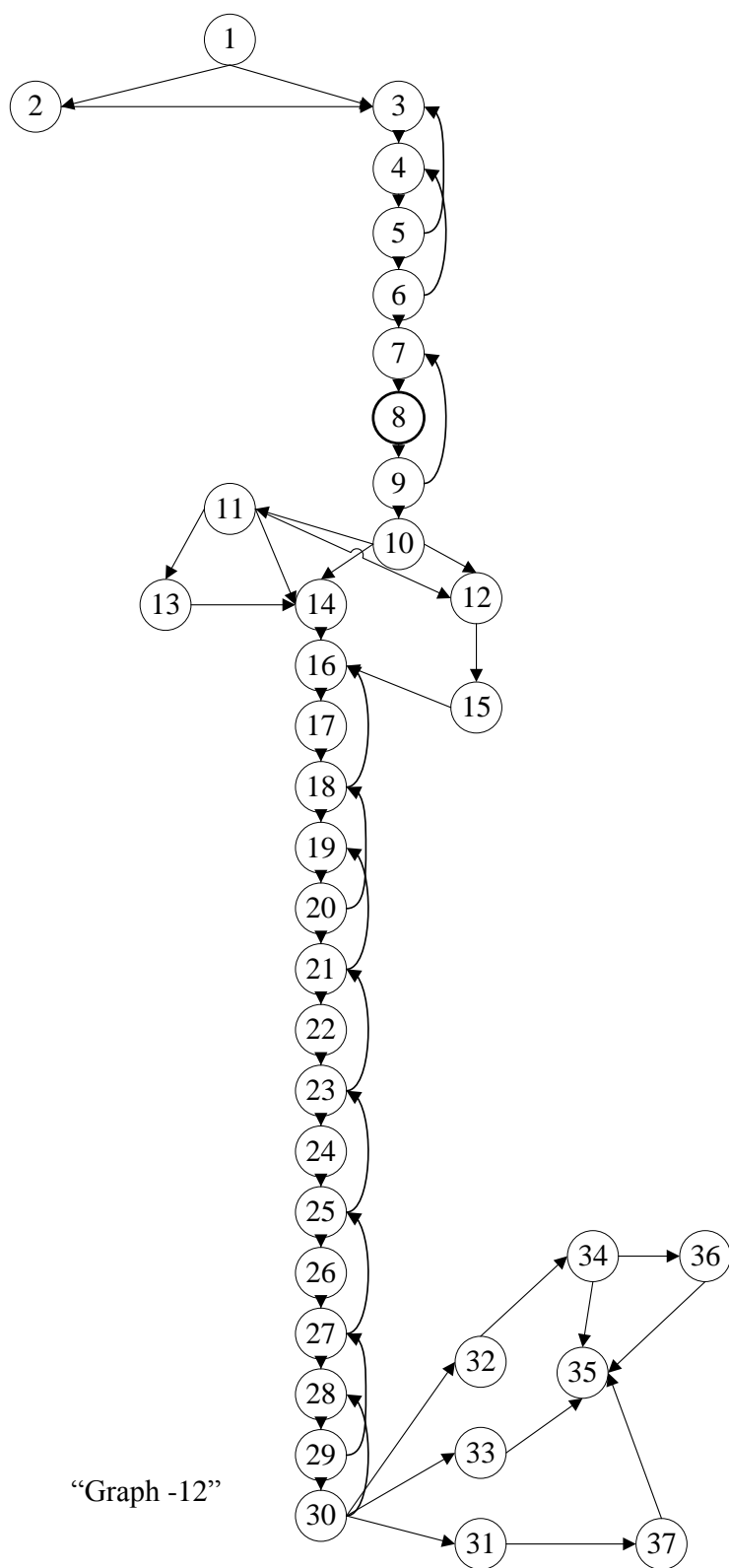




“Graph -10”



“Graph -11”



Appendix B

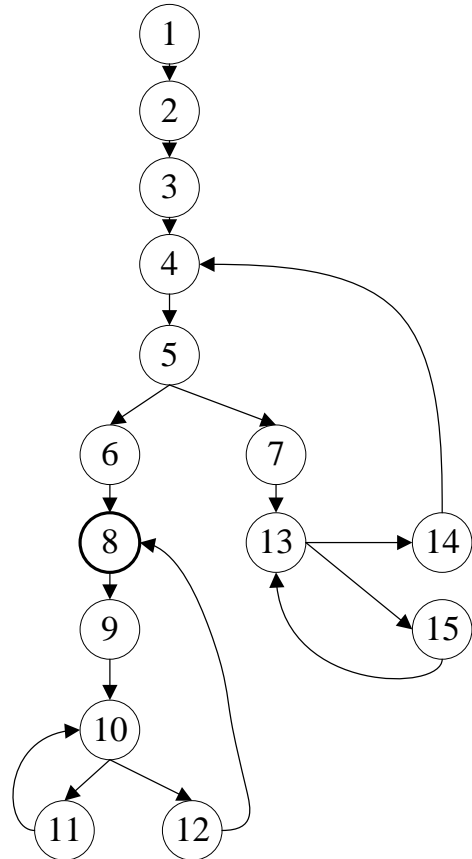
```

function x = splv(A, b)

% splv The solution to a square, invertible
system.
% x = splv(A, b) uses the PA = LU factorization
% computed by splu to solve Ax = b.

[P, L, U] = splu(A);
[n, n] = size(A);
% Permute the right hand side.
b = P*b;
% Forward elimination to solve L*c = b.
c = zeros(n, 1);
for k = 1:n
    s = 0;
    for j = 1:k-1
        s = s + L(k, j)*c(j);
    end
    c(k) = b(k) - s;
end
% Back substitution to solve U*x = c.
x = zeros(n, 1);
for k = n:-1:1
    t = 0;
    for j = k+1:n
        t = t + U(k, j)*x(j);
    end
    x(k) = (c(k) - t) / U(k, k);
end

```

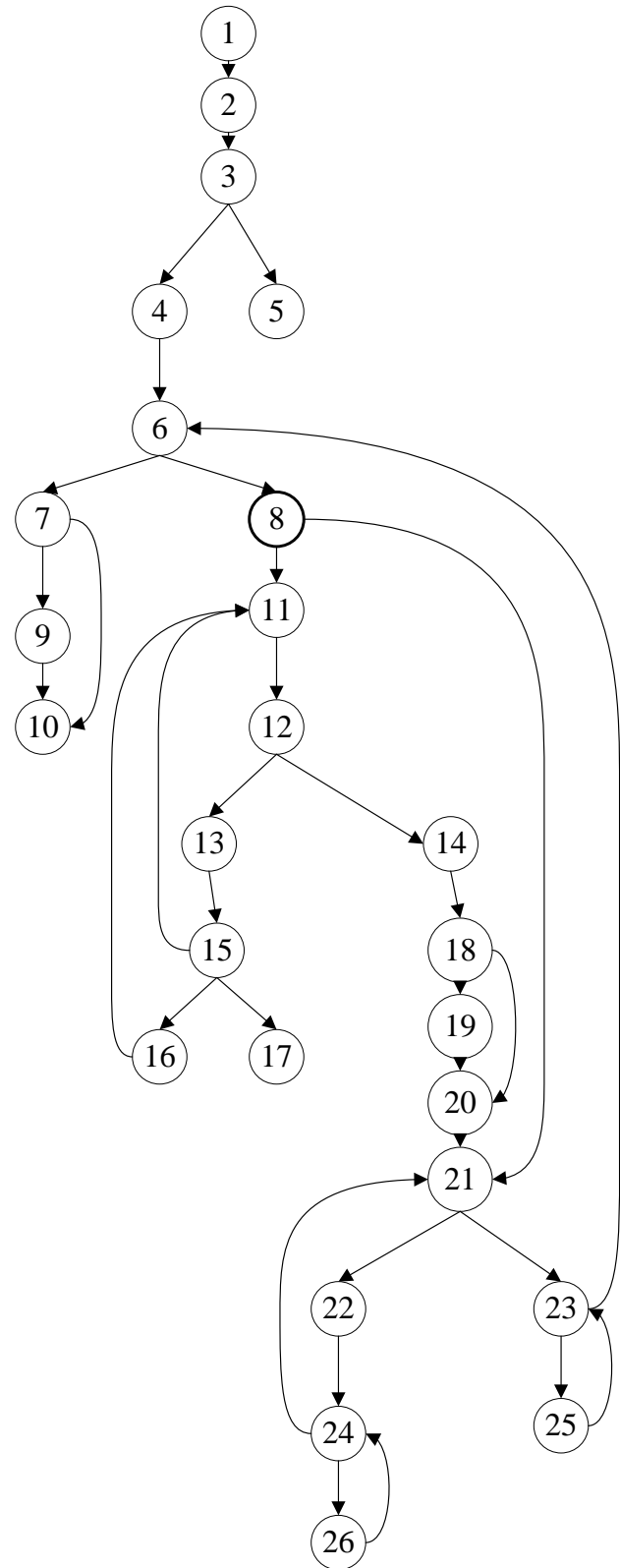


1. “Code 1 and its corresponding Control Flow Graph”

```

function [P, L, U, sign] = splu(A)
% splu Square PA=LU factorization *with row exchanges*.
% [P, L, U] = splu(A), for a square, invertible matrix A,
% uses Gaussian elimination to compute a permutation
% matrix P, a lower triangular matrix L and
% an upper triangular matrix U so that  $P \cdot A = L \cdot U$ .
% P, L and U are the same size as A.
% sign = det(P); it is 1 or -1.
[m, n] = size(A);
if m ~= n
    error('Matrix must be square.')
end
P = eye(n, n);
L = eye(n, n);
U = zeros(n, n);
tol = sqrt(eps);
sign = 1;
for k = 1:n
    if abs(A(k, k)) < tol
        for r = k:n
            if abs(A(r, k)) >= tol
                break
            end
        end
        if r == n
            if nargout == 4
                sign = 0;
                return
            else
                disp('A is singular within tolerance.')
                error(['No pivot in column ' int2str(k) '.'])
            end
        end
    end
    A([r k], 1:n) = A([k r], 1:n);
    if k > 1, L([r k], 1:k-1) = L([k r], 1:k-1); end
    P([r k], 1:n) = P([k r], 1:n);
    sign = -sign;
end
for i = k+1:n
    L(i, k) = A(i, k) / A(k, k);
    for j = k+1:n
        A(i, j) = A(i, j) - L(i, k)*A(k, j);
    end
end
for j = k:n
    U(k, j) = A(k, j) * (abs(A(k, j)) >= tol);
end
if nargout < 4
    roworder = P*(1:n)';
    disp('Pivots in rows:'), disp(roworder); end
end

```



2. “Code 2 and its corresponding Control Flow Graph”

```

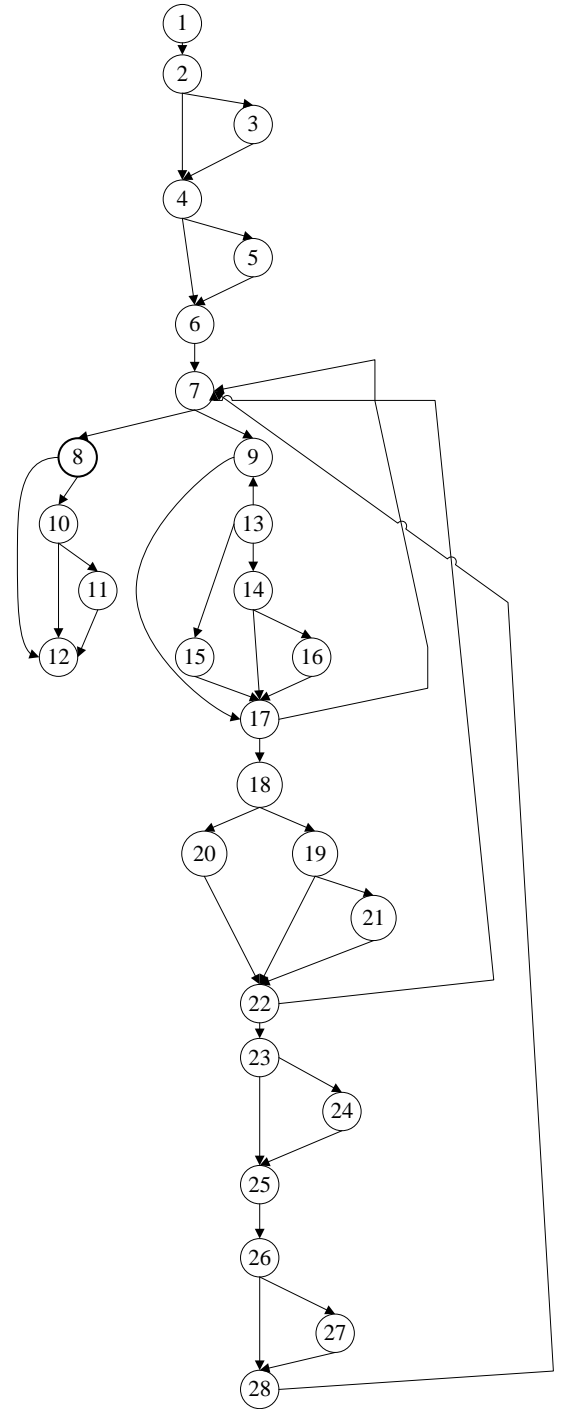
function p = poly2str(c, x)

% poly2str Convert a polynomial coefficient vector to a string.
%
% p = poly2str(c) generates a string representation of the polynomial
% whose coefficients are in the vector c.
% The default variable is 'x', unless otherwise specified by
% p = poly2str(c, 's').
% The coefficients are approximated, if necessary, by the rational
% values obtained from rat.
%
% If x has a numeric value and the elements of c are reproduced
% exactly by rat, then eval(poly2str(c)) will return the same value
% as polyval(c, x).
%
% See also polyval, rat.

if nargin < 2, x = 'x'; end
if all(c == 0), p = '0'; return, end

p = [];
n = length(c);
for d = 0: n-1
    if d > 0
        if c(n-d+1) > 0
            p = [' + ' p];
        elseif c(n-d+1) < 0
            p = [' - ' p];
        end
    end
    if c(n-d) ~= 0
        if d == 1
            p = [x p];
        elseif d > 1
            p = [x '^' int2str(d) p];
        end
        if (abs(c(n-d)) ~= 1) | (d==0)
            if d > 0,
                p = ['*' p];
            end
            [sn, sd] = rat(abs(c(n-d)));
            s = num2str(sn);
            if sd ~= 1, s = [s '/' num2str(sd)]; end
            p = [s p];
        end
    end
end
end
if n > 0
    if c(1) < 0
        p = ['- ' p];
    end
end
end

```

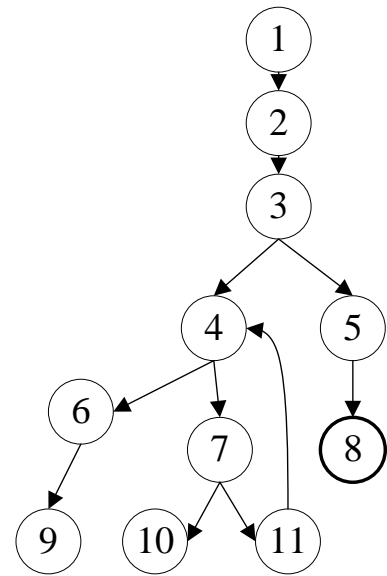


3. “Code 3 and its corresponding Control Flow Graph”

```

function [Q, R] = grams(A)
% grams Gram-Schmidt orthogonalization of the
% columns of A.
% The columns of A are assumed to be linearly
% independent.
% Q = grams(A) returns an m by n matrix Q whose
% columns are
% an orthonormal basis for the column space of A.
% [Q, R] = grams(A) returns a matrix Q with
% orthonormal columns
% and an invertible upper triangular matrix R so that A
% = Q*R.
%
% Warning: For a more stable algorithm, use [Q, R] =
% qr(A, 0) .
[m, n] = size(A);
Asave = A;
for j = 1:n
    for k = 1:j-1
        mult = (A(:, j)'*A(:, k)) / (A(:, k)'*A(:, k));
        A(:, j) = A(:, j) - mult*A(:, k);
    end
end
for j = 1:n
    if norm(A(:, j)) < sqrt(eps)
        error('Columns of A are linearly dependent.')
    end
    Q(:, j) = A(:, j) / norm(A(:, j));
end
R = Q'*Asave;

```

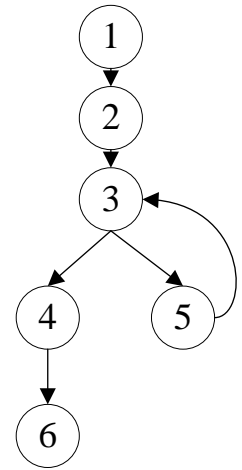


4. “Code 4 and its corresponding Control Flow Graph”

```

function [S, D] = eigvec(A)
% eigvec Eigenvectors and their geometric multiplicity.
% S = eigvec(A) returns the largest possible set of
linearly
% independent eigenvectors of A.
% [S, D] = eigvec(A) also returns the corresponding
eigenvalues
% in the diagonal matrix D.
% Each eigenvalue in D is repeated according to the
number of its
% linearly independent eigenvectors. This is its
geometric multiplicity.
% Always  $A*S = S*D$ . If S is square then A is
diagonalizable and
%  $\text{inv}(S)*A*S = D = \text{LAMBDA}$ .
[m, n] = size(A);
I = eye(n);
[evalues, repeats] = eigval(A);
S = []; d = [];
for k = 1 : length(evalues);
    s = nulbasis(A - evalues(k)*I);
    [ms, ns] = size(s);
    S = [S s];
    temp = ones(ns, 1) * evalues(k);
    d = [d; temp];
end
D = diag(d);

```

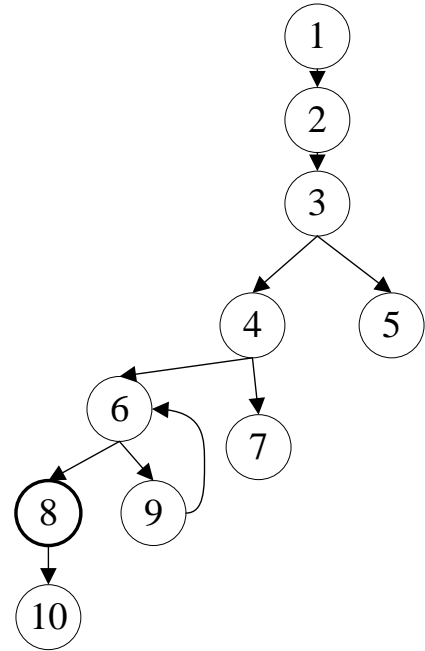


5. “Code 5 and its corresponding Control Flow Graph”

```

function x = cramer(A, b)
% cramer Solve the system Ax=b.
% The matrix A is square and invertible.
%
% x = cramer(A, b) solves the square system Ax = b.
[m, n] = size(A);
if m ~= n
    error('Matrix is not square.')
end
if det(A) == 0
    error('Matrix is singular.')
end
for j = 1:n
    B = A;
    B(:, j) = b;
    x(j) = det(B) / det(A);
end
x = x';

```

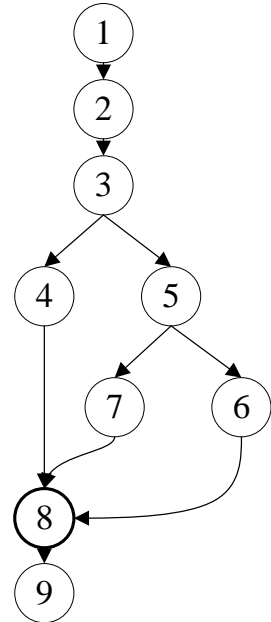


6. “Code 6 and its corresponding Control Flow Graph”

```

function eigen2(A)
% eigen2 Characteristic polynomial, eigenvalues,
% eigenvectors
% of a 2 by 2 matrix.
% eigen2(A) prints the characteristic polynomial det(A-e*I),
% eigenvalues, and eigenvectors of A.
% If A is not diagonalizable, its single eigenvector is
% printed twice.
d = A(1,1)*A(2,2) - A(1,2)*A(2,1);
t = A(1,1) + A(2,2);
e1 = (t + sqrt(t^2 - 4*d))/2;
e2 = (t - sqrt(t^2 - 4*d))/2;
if A(1,2) ~= 0
    x1 = [A(1,2); e1-A(1,1)];
    x2 = [A(1,2); e2-A(1,1)];
elseif A(2,1) ~= 0
    x1 = [e1-A(2,2); A(2,1)];
    x2 = [e2-A(2,2); A(2,1)];
else
    x1 = [1; 0];
    x2 = [0; 1];
end
disp(' ')
disp('For this matrix, the polynomial whose roots are the
eigenvalues is:')
disp([' e^2 - ' num2str(t) '*e + ' num2str(d) ' = 0'])
disp(' ')
disp('The first eigenvalue and eigenvector are:')
e1
x1
disp(' ')
disp('The second eigenvalue and eigenvector are:')
e2
x2

```



7. “Code 7 and its corresponding Control Flow Graph”

Vita

Satya Aditya. Akundi is a doctoral candidate in Electrical and Computer engineering with a concentration in Industrial and Systems engineering at the University of Texas at El Paso (UTEP). He earned a Master of Science in Electrical and Computer Engineering at the University of Texas at El Paso (UTEP) in 2012. He has worked on a number of projects in the field of Systems Engineering, Additive Manufacturing and Green Energy Manufacturing. He held several leadership positions including president of INCOSE UTEP student chapter and board member of UTEP Green Fund committee. His research interests are in Systems Engineering & Architecture, Complex systems, Systems testing and Application of Entropy to Complex Systems.

Address: 411 New York Avenue,

El Paso, TX, 79902.

This dissertation was developed by Satya Aditya. Akundi