Mobilising the Enterprise: A Game Theoretical Trust Framework for Emerging Systems

Andrew Young, M.Math. (Sussex) M.Sc. (Brighton)

College of Business, Victoria University

Submitted in fulfilment of the requirements of the degree of Doctor of Philosophy

May, 2016

“This is a war universe. War all the time. That is its nature. There may be other universes based on all sorts of other principles, but ours seems to be based on war and games. All games are basically hostile. Winners and losers. We see them all around us: the winners and the losers. The losers can oftentimes become winners, and the winners can very easily become losers.” -- William S. Burroughs (Burroughs 1991)

Abstract

Trust frameworks are of importance for increasingly mobile and dynamic enterprise ad-hoc systems to protect privacy, secure information and establish credibility. Trust enables humans and systems to accept risks and manage uncertainty.

While various frameworks have been proposed, a common limitation is that they apply to closed systems where a central trust authority, a known inventory and the fair distribution of resources can be assumed. Open systems such as the Internet, cannot be considered under these assumptions. Enterprises increasingly consist of independently highly reprogrammable nodes and elements that are non-cooperative in nature.

The original contribution of this work is that it identifies the need for and defines *Emerging Systems* as open, mobile ad-hoc systems consisting of highly-reprogrammable nodes within the enterprise, and shows that inherent limitations of these systems can be overcome by supporting proof that a non-cooperative game theoretical model is a suitable foundation for a *Non-cooperative Programmable Open System Trust Framework (NPOST)* for this new class of system.

The framework's underpinning theoretical model is defined by the formulation of mathematical constructs of a trust nomenclature and through rigorous application of non-cooperative game theoretical techniques to establish stability and (Nash) equilibrium.

The framework is experimentally examined, with the results showing robustness under scale (small and large), partitioning (volatile and ephemeral topology) and with changing environmental influence, all conditions characteristic of Emerging Systems.

Student Declaration

I, Andrew Young, declare that the Ph.D. thesis entitled *Mobilising the Enterprise: A Game Theoretical Trust Framework for Emerging Systems* is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

Signature: 

Date: 5th December, 2015

Acknowledgments

I gratefully and respectfully acknowledge Victoria University for sponsoring this work and for demonstrating their commitment through the award of the Dean's Research Scholarship.

For their supervision, I wholeheartedly thank **Jana Polgar** and **Arthur Tatnall**. Without Jana, the topic of this thesis would never have been seeded, and many valuable associations and academic contributions, never made. Without Arthur, academic and personal progress would never have been made through vacillating productivity and confidence. Without the wisdom of both of these individuals, I would remain lost.

For editorial assistance, I recognise the delightfully unexpected and warmly welcome contribution of **David Abrams**. Fastidious in his deliberations, David was often afforded the dubious honour of bearing witness to the innumerable increasingly inappropriately named, "first draft" of the same thing. If this isn't the time to be pedantic, I don't know when is.

Bill Davey, I thank for his assistance, particularly with the mathematics but also philosophically, for his dedication to simplicity and the boundless capacity for trust. Thank you for lunch, too.

For her enhancements to my diagrams, my appreciation extends to **Gemma Wilson**. What is aesthetically pleasing is not always mathematically sound, what is mathematically rigorous however, is always elegant.

To **Edward Champion de Crespigny**, thank you for hanging in there, despite the hypochondria.

Finally, I can only marvel at **Jade Anderson**'s unwavering tolerance and humanity throughout my lengthy obsession. There are very few like her and I am privileged to know one and one so dedicated.

Thank you.

Pivotal to this thesis is the work of John Nash. In advance of the completion of this dissertation, Nash and his wife, Alice, were tragically killed together in New Jersey, USA on 23rd May, 2015 (Goode 2015).

List of Publications and Awards

Publications from this work include:

- G. Adamson and J. Polgar (A. E. Young, “Mobilising the Enterprise”), *Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies* vol. Hershey, PA, USA: IGI Global, 2012 (Young 2012).
- A. E. Young and M. Jessopp, "How Thick Is Your Client?," *International Journal of Web Portals (IJWP)*, vol. 2, pp. 1-11, 2010 (Young and Jessopp 2010).
- A. E. Young, "Service Oriented Architecture Conceptual Landscape PART IV," *International Journal of Web Portals*, vol. 5, 2009 (Young 2009).
- A. E. Young, "Service Oriented Architecture Conceptual Landscape PART III," *International Journal of Web Portals*, vol. 4, 2009 (Young 2009).
- A. E. Young, "Service Oriented Architecture Conceptual Landscape PART II," *International Journal of Web Portals*, vol. 3, April 2009 (Young 2009).
- A. E. Young, "Service Oriented Architecture Conceptual Landscape PART I," *International Journal of Web Portals*, vol. 3, January 2009 (Young 2009).
- A. E. Young, "Every Need to be Alarmed," *International Journal of Web Portals*, vol. 1, pp. 34-49, December 2008 (Young 2008).

This work was supported by the *Victoria University Dean’s Research Scholarship Award*, 2009.

Table of Contents

List of Figures

List of Tables

Dedication

This work is dedicated to my parents, Mary and Martin Young without whom,

I literally would not be here.

For Elsie; I promised you I would – sorry it took so long.

Non sum dignus.

1 Introduction

1.1 Context and Background

The advent of large open, distributed systems of highly programmable participants such as the Internet has extended the boundary of the enterprise beyond the traditional infrastructure and against a centralised governance model. The enterprise is now a collection of highly capable, context-sensitive computational elements all interacting with each other over constantly changing connections (Basole 2008) (Sorensen 2011) (Chakraborti, Acharjya et al. 2015) (Knackmuß and Creutzburg 2015).

The proliferation of transient information and communication technologies has led to a profound change in the way people work, communicate, and collaborate and conduct business. Enterprises recognise the importance and potentially transformative impact of enterprise mobility. While the concept of enterprise mobility continues to mature in management and technology, it is still not well understood (Barnes 2003) (Basole 2008) (Chen 2015).

"The workforce is becoming increasingly dynamic as information demand is everywhere and all the time. Pervasive information is the only way to keep up and the only way to persistently consume this information is high availability through mobility" (Young 2009).

"Mobility" in this sense, is not synonymous with "mobile device" such as a laptop, phone, tablet or any other portable hardware device that is specifically designed to provide access to an enterprise network. It refers more, to the movement of the consumer – human or machine - in that they can interact with assortment of devices or more generically "nodes", within a system from any physical (geographic) or logical (connectivity delimited) location. The nodes themselves can take most any form as the "Internet of Things" (IoT) attests (Ashton 2014) (Yan, Zhang et al.). Further, nodes are becoming increasingly capable, able to be programmed easily to perform multiple and varied tasks (Harter, Pissard-Gibollet et al. 2015) (Young and Jessopp 2012). The hackneyed comparison that there is more computational power in a standard mobile phone now than was available to NASA Apollo moon landing mission in 1969 (Cindy McArthur : Hq 2009), still exhibits pertinence.

Technology has led and proliferated the disruption of established edges of the enterprise. "Cloud" computing allows large amounts of data to be stored and analysed without physical boundaries, allowing significant scale, complex computation to be carried out quickly and inexpensively, remotely from the enterprise (Qian and Andresen 2015), even at the quantum level (Rahaman and Islam 2015). "Fog" computing is an extension of the cloud concept where consumer and infrastructure nodes share the computational responsibility (Yi, Li et al. 2015) (Loke 2015) to

improve efficiency and reduce the amount of data that needs to be transported for processing, analysis and storage. It may also be architected for security or compliance reasons (Stojmenovic, Wen et al. 2015). In parallel, communication speed and capacity has increased massively to support growing demand for access (Eha 2013) (ElDelgawy and La 2015), and is considered as fundamental as water and electricity (Beck 2015) for business and a case for public policy concern (Raja 2015). The variety of communication types has increased with "Bluetooth" (Bluetooth.org 2015) considered the "backbone of IoT" (Palumbo, Barsocchi et al. 2015) and Near Field Communication (NFC) (Forum 2015) for close communication and location services, particular for mobile devices and financial payments (Chae and Hedman 2015) (Imbachi, Jacome et al. 2015) (Pham and Ho 2015) (Cocosila and Trabelsi 2015). Enterprise service consumers, have introduced their own devices into the enterprise. Information Technology (IT) consumerisation has provoked Bring Your Own Device (BYOD) (Weeger, Wang et al. 2015) (French, Guo et al. 2015) (Freedman 2015) (Donaldson, Siegel et al. 2015) whereby policy permits employees to access privileged company information and applications on their personal devices. BYOD is seen significant enterprise adoption, with three quarters of employees in high-growth markets such as Brazil and Russia and almost half in developed markets already using their own technology at work (logicalis 2012). Surveys have indicated that enterprises are unable to stop employees from bringing personal devices into the workplace (Itpro 2015). Research is divided on the benefits (Weeger, Wang et al. 2015) (Pande and Gomes 2015) with the division primarily present between the perceived increased productivity of employees and the effort of regulating and supporting a diversity of consumer platforms.

This technology transformation has brought about seemingly paradoxical business paradigms based on consumer collaboration and self-regulation, which disrupt traditional models; Wikipedia is the largest repository for what is traditionally an encyclopaedia and yet, employs no authors (Holman Rector 2008). Facebook is the largest generator of content in the world but creates no content of its own (Tam 2013). eBay is the one of the largest shops but holds no inventory (Resnick, Zeckhauser et al. 2006). Bitcoin do not mint any currency (Bitcoin Foundation 2015). Similar models are present for Twitter, YouTube, Amazon, amongst others.

"Infrastructure is everywhere but you do not tend to notice it unless it is missing or not functioning properly" (Beck 2015).

The “new” enterprise is characterised by (Barnes 2003) (Basole 2008) (Chen 2015) (Shah, Jan et al. 2012):

- Highly-programmable consumer nodes;
- Decentralisation;
- High distribution;
- Self-configuration;
- Self-regulation;
- Non-cooperation;
- Pervasiveness;
- Dynamic topology;
- No fixed infrastructure;
- Hybrid wireless and wired connectivity, and;
- High scalability.

The enterprise now has to support remote and ad-hoc interactions while still assuring that same level of service to its consumers, both internal and external. Single authority control and cooperation cannot be assumed in contemporary enterprises:

“The increased capability of reprogrammability[sic] of [nodes] offers another threat to this assumption. It is, therefore, important that the issues in networks…should be addressed by using the concepts from non-cooperative game theory” (Shah, Jan et al. 2012).

The scope of the thesis to business-technical and resolutely not business-economic-social.

Throughout this technological transformation, the fundamental concerns of the enterprise have not changed the fundamental concerns of the enterprise to the extent they are reflected in technology architecture frameworks (Smith 2015):

- Effectiveness;
- Efficiency;
- Agility, and;
- Durability.

However, the enterprise can no longer be operated as a “closed” system with centralised technology governance and physical restrictions to access as it no longer fits the business and consumer models. To continue to service the needs of the enterprise by delivering against its fundamental concerns, there needs to be an approach that is responsive to the new technological frontiers.

The enterprise consists of four architectural domains (OpenGroup 2015):

1. *Business Architecture* defines the business strategy, governance, enterprise, and key business processes.
2. *Data Architecture* describes the structure of an enterprise's logical and physical data assets and data management resources.
3. *Application Architecture* provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core enterprise processes of the enterprise.
4. *Technology Architecture* describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services. This includes IT infrastructure, middleware, networks, communications, processing and standards.

Underpinning the domains is the concept of “trust”. Various engineering models such as security, usability, reliability, availability, safety and privacy incorporate some limited aspects of trust with different meanings (Thooyamani, Udayakumar et al. 2014). The concept is derived from Social Science and is defined as the degree of subjective belief about behaviours of a particular entity (Capra 2004). Technology adopts “trust” from the social sciences as a metaphor to describe a relationship between two neighbouring nodes where a trust value expresses the degree that one entity expects another node to offer certain services. The reputation of a node, is the record of the trust values attributed to a node by the consensus of other node.

Thooyamani, Udayakumar et al. (2014) identify the general benefits of establishing trust in the enterprise as follows:

- Trust solves the problem of providing corresponding access control based on judging the quality of the nodes and their services. Significantly, "This problem cannot be solved through the traditional security mechanisms" (Thooyamani, Udayakumar et al. 2014);
- Trust solves the problem of providing reliable communication paths that do not contain malicious, selfish, or faulty node(s), and;
- Trust makes the traditional security services more robust and reliable by ensuring that all the communicating nodes are trusted during authentication and authorisation.

The benefits apply to all domains of the enterprise architecture. There is a case to be answered to isolate trust as a suitable candidate for the fifth domain, since it transcends all the other domains and to reflect its importance in the contemporary enterprise.

To establish trust in a system, it is common to appeal to a framework that underpins how trust is evaluated and communicated. Trust frameworks are important for mobile ad-hoc systems to protect privacy, secure information and establish credibility. Trust enables humans to accept risks and manage uncertainty.

While various frameworks have been proposed (Ziegler and Golbeck 2015) (Sethumadhavan, Waksman et al. 2015) (Noor, Sheng et al. 2015) (Thooyamani, Udayakumar et al. 2014) (Rajesh and Kumar 2014) (Huth and Kuo 2014) (Firdhous, Ghazali et al. 2014) (Xia, Jia et al. 2013) (Gunasekaran and Premalatha 2013), a common limitation is that they apply to closed systems where a central trust authority, a known inventory and the fair distribution of resources can be assumed. Open systems such as the Internet, cannot be considered under these assumptions. They are non-cooperative and increasingly consist of independently highly reprogrammable nodes.

A trust framework to support the enterprise needs to consider the main trust features demanded by the enterprise and are amalgamated as (Golbeck 2006) (Cho and Swami 2009) (Adams and Davis IV 2005):

1. Attribution of a reputation to an entity must be distributed because the existence of a central, trusted authority cannot be assumed;
2. Trust must be established in a highly flexible fashion that captures the complexities of trust relationships between entities.
3. Consideration must be made of the computation and communication overhead of establishing trust relationships;

4. Trust frameworks should not assume that all nodes are cooperative;
5. Trust is dynamic (not static). Reputation changes over time - diachronically;
6. Trust is subjective and based on or influenced by individual entity environmental factors, constraints and opinions;
7. Trust is not transitive. If X trusts Y, and Y trusts Z, it does not mean that X trusts Z;
8. Trust is asymmetrical and cannot be assumed to be reciprocal;
9. Trust is contextual. The circumstances and domain of trust should be defined.

To enforce these features of trust within a system, a mathematical foundation to the framework is required. Game Theoretical techniques provide a rich and flexible approach to describing the system. Game Theory allows the system to be modelled as a "game" where the nodes in a system are rational "players" of the game and the outcome is a consensus trust within the system that guides interactions and decision, and protects the system from malicious or recalcitrant intent. We must explicitly exclude human actors in the common definition of "system" as they cannot be assumed rational.

When the system is modelled as a non-cooperative game, the solution concept of Nash Equilibrium is a system involving two or more nodes, in which each node is assumed to know the equilibrium strategies of the other nodes, and no node has anything to gain by changing only its own strategy.

In this work, trust between machines is a metaphor for human-actors. Experience and observation, can be modelled approximately within a system but concepts such as gullibility and malice are less apparently easy to model. This is directly related to the assumption that machine-actors are always rational unless there is a failure of some kind within the system.

Sufficient conditions to guarantee that a Nash equilibrium game is played are (Aumann and Brandenburger 1995) (Nash 1951):

1. The nodes all will do their utmost to maximise their expected payoff as described by the game;
2. The nodes are flawless in execution;
3. The nodes have sufficient intelligence to deduce the solution;
4. The nodes know the planned equilibrium strategy of all of the other nodes;
5. The nodes believe that a deviation in their own strategy will not cause deviations by any other nodes, and;
6. There is common knowledge that all nodes meet these conditions, including this one. So, not only must each node know the other nodes meet the conditions, but also they

must know that they all know that they meet them, and know that they know that they know that they meet them, and so on.

If each node has chosen a strategy and no node can benefit by changing strategies while the other nodes keep theirs unchanged, then the current set of strategy choices and the corresponding "payoffs" constitutes a Nash Equilibrium. The payoff of the game is the level of trust attributed to each node, or the nodes "reputation" within the system.

1.2 Definitions

Term	**Definition**
Bluetooth	A standard for the short-range wireless interconnection of mobile phones, computers, and other electronic devices.
BYOD	*Bring Your Own Device* - the practice of allowing the employees of an organisation to use their own computers, smartphones, or other devices for work purposes.
Closed System	A system with fixed technology boundaries.
Cloud Computing	A network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.
Distributed System	A software system in which components located on networked computers communicate and coordinate their actions by passing messages
Enterprise	A business, organisation or company, usually of significant scale.
Enterprise Architecture	A conceptual blueprint that defines the structure and operation of an enterprise. The intent of an enterprise architecture is to determine how an organisation can most effectively achieve its current and future objectives.
Fog Computing	Facilitates the operation of compute, storage and networking services between end devices and cloud computing data centres.
Game	A form of competitive activity played according to rules.

Game Theory	The branch of mathematics concerned with the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice of action depends critically on the actions of other participants.
IoT	*Internet of Things* - a proposed development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data.
Nash Equilibrium	(In economics and game theory) a stable state of a system involving the interaction of different participants, in which no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged.
NFC	*Near Field Communication* - a set of short-range wireless technologies, typically requiring a distance of 4cm or less to initiate a connection.
Node	A point in a network or diagram at which lines or pathways intersect or branch. An entity in a game theoretical, game system.
Non-Cooperative Game	In game theory, a non-cooperative game is one in which players make decisions independently. Thus, while players could cooperate, any cooperation must be self-enforcing. A game in which players can enforce contracts through third parties is a cooperative game.
Open System	A system with open, distributed technology boundaries.
Reputation	The record of the trust values attributed to an entity by the consensus of other entities.
System	A set of things working together as parts of a mechanism or an interconnecting network; a complex whole. A set of principles or procedures according to which something is done; an organised scheme or method.
Trust	A relationship between two neighbouring nodes where a trust value expresses the degree that one entity expects another node to offer certain services

1.3 Motivation and Research Gap

The literature attests that there is no comprehensive definition of contemporary enterprise systems. Current comparable definitions do not consider all of the characteristics of an open system with highly programmable nodes that are not beholden to central governance that consequently, cannot be assumed cooperative. Without an established definition, the Community (see 1.6 Significance for Particular Audiences) is unable to distinctly identify systems of this type and develop them accordingly.

1.4 Research Question

Can a comprehensive definition of *Emerging Systems* be established to address the research gap identified for contemporary enterprise system characteristics, and can the suitability of a supporting trust framework be demonstrated theoretically and experimentally, for the definition?

1.5 Contribution and Significance

The original, and overall contribution and significance of this work to knowledge, is that it:

1. **establishes the need for a new definition of a sui generis class of computational system designed to support the nature of the contemporary enterprise and provides it – *Emerging Systems*;**
2. **supports the definition of Emerging Systems with a mathematical underpinning and nomenclature, so that they can be described and explored universally in a well-defined manner;**
3. **validates the need and suitability of a non-cooperative game theoretical trust (Non-cooperative Programmable Open System Trust (NPOST)) framework to support the reliable formation of an Emerging System, and;**
4. **develops and experimentally examines the trust framework to establish its suitability to specifically support the characteristics of Emerging Systems.**

References to “this work” are to the thesis “Mobilising the Enterprise: A Game Theoretical Trust Framework for Emerging Systems” throughout, unless contextually parentally different. In all subsequent sections the convention is that all Contribution and Significance statements are indicated with a **bold** font decoration.

1.6 Significance for Particular Audiences

This work is of particular significance to the fields of:

Audience	**Significance**
Business modelling and analysis	Contribution to the design of business systems that cater for pervasive and distributed business functions and how they could be used in collaboration.
Enterprise business compliance	Establish conditions and terms of use for distributed elements in a system to ensure that suitable constraints are enforced – levels of trust within a system – as environmental factors.
Actor-Network Theory	Application to the social theory and the nature of the interactions as part of social networks, based on trust mechanics.
Architecture	Infrastructure, Solution, Information and Enterprise – design systems that securely support decentralised authority and communicate the design effectively.
Network security	Facilitate a distributed infrastructure without compromise to the enterprise's integrity. Mechanisms for handling security compromise – damage limitation.
Mobile application technology	Promote alternative approaches to the development of applications that adopt distributed systems to establish consensus.
Game theoretical modelling	Framework for modelling games and computing equilibrium results at scale with application to distributed, enterprise business systems.
Parallel and distributed computing	Application and implementation of a framework for distributed and parallel methods, and their applications.

1.7 Roadmap

This work consists of three primary chapters, between this introductory and a final overarching conclusive, chapters:

1.7.1 Literature Review

A review of the literature is conducted under four topic areas to identify the research gap for this work:

1. Game Theory – a brief historical account of the evolution of Game Theory as a systematic method for analysis of strategic problems, primarily in the field of Economics and an exploration of the basic game theoretical principles applied in this work;
2. Emerging Systems – where the need for and a formal definition of, Emerging Systems and their characteristics is established;
3. Trust and Emerging Systems – a discussion of the need for and formal definition of "trust" within the context of Emerging Systems, and;
4. Game Theory and Emerging Systems – an exploration of the application of game theoretical techniques to Emerging Systems, specifically non-cooperative Game Theory and the establishment of trust as a Nash Equilibrium.

1.7.2 A Mathematical Trust Framework for Emerging Systems

This chapter defines Emerging Systems using Graph Theoretical techniques to describe the presence and interactions of members of a system.

Definitions are established for Emerging Systems, culminating in a model that describes how the concepts relate within the framework. The mathematical underpinnings of the framework are established through the fundamental concepts:

- Trust Space;
- Reputation Profile;
- Trust Profile;
- Trust Value, and;
- Environmental Factors.

The chapter identifies the algorithm and game theoretical components that are to be applied in the NPOST simulation:

The concept of a Trust Function is introduced and mathematically explored for stability properties and establishment of consensus trust within the framework. Functions are identified as candidates for experimental analysis.

Iterative methods are discussed in principle, to establish their suitability for use for experimental analysis of the framework simulation. A pseudo-code programmable script is selected as a suitable implementation of the chosen Jacobi OverRelaxation (JOR) algorithm.

1.7.3 Experimental Analysis

Having established the fundamental foundations of the Emerging System Trust Framework, the final primary chapter discusses and determines suitable theoretical research methods for testing the framework simulation. An applicable approach is identified, and the method and approach described in terms of the Emerging System concepts previously established.

The chapter goes on to report the results and interpretations of experiments to support the research objectives, conducted within a simulation of the establishment of consensus trust for an Emerging System. The experiments consider the performance of the simulation when the system is scaled, partitioned and with changing environmental influencing factors, before the conditions for equilibrium are breached. Support for the hypotheses is evaluated and limitations to the experiments explained with a view to further research.

1.8 Conclusion

This work establishes the definition of *Emerging Systems* based on the characteristics of contemporary enterprise systems. Further, it develops a supporting game theoretical trust framework for the definition and demonstrates the suitability of the framework for Emerging Systems theoretically and experimentally.

2 Literature Review

2.1 Introduction

In the related literature, Emerging Systems are explored independently and in relation to trust frameworks and their uses. Following an exposition of the classical results in Game Theory as a foundation to their contemporary application, the exploration is extended to using game theoretical techniques within Emerging Systems and trust frameworks, and how this has been applied previously. **The culmination of this literature review is the identification of the need for a definition of *Emerging Systems* and for a game theoretical trust framework to support them.**

This literature review supports the following contentions:

- A definition of *Emerging Systems* is required;
- Emerging Systems need to be defined and characterised as a specialisation of Mobile Ad-hoc NETworks (MANET) and as open systems;
- There is an absence of an application level trust layer to support the highly reprogrammable nature of nodes in Emerging Systems;
- A game theoretical approach to trust aggregation is appropriate for Emerging Systems and should be strictly non-cooperative to support their characteristics, and;
- A trust framework should be experimentally examined.

The four main sections of this review cover the full scope of this work, supporting the research gap claim and guiding the contribution that it makes.

2.1.1 Roadmap

This literature review covers four main topic areas:

- Game Theory;
- Emerging Systems;
- Trust and Emerging Systems, and;
- Game Theory and Emerging Systems.

2.1.1.1 Game Theory

The *Game Theory* section aims to provide some historical context from the literature to the concepts applied in this work and to identify significant contributions to the field. The main mathematical result from the section is Nash equilibrium which is a solution concept of a non-cooperative game involving two or more agents, in which each agent is assumed to know the equilibrium strategies of the other agents, and no agent has anything to gain by changing only their own strategy.

2.1.1.2 Emerging Systems

As the foundation for the last two sections, the *Emerging Systems* section examines the characteristics of Emerging Systems from the literature, providing a formal definition for Emerging Systems. The section examines the nature of highly reprogrammable nodes and how they distinguish Emerging Systems from other types of the system.

2.1.1.3 Trust and Emerging Systems

Building on the previous section's definition, the section *Trust and Emerging Systems* provides a review of trust concepts in Emerging Systems and a formal definition of "trust". The section explores the implementation of trust-based frameworks in the literature and considers them with respect to this work.

2.1.1.4 Game Theory and Emerging Systems

In the final section, *Game Theory and Emerging Systems* the literature reviewed considers game theoretical approaches to Emerging Systems for various problems including trust. The principle result in this section is the applicability of *non*-cooperative game theory to finding solutions to strategic problems in Emerging Systems. This section is the culmination of the previous sections' concepts that leads to the concluding emphasis fundamental to this work – a trust-based game theoretical framework for Emerging Systems.

2.1.2 Contribution and Significance

This chapter identifies a research gap that exists for:

1. **the definition of *Emerging Systems* for the contemporary enterprise, and;**
2. **a Trust Framework that:**
    a. **is suitable for Emerging Systems**
    b. **can be implementable to support the application layer, and;**
    c. **is specifically, non-cooperative.**

2.1.3 Non-cooperative Programmable Open System Trust (NPOST) Framework

From the literature, this work posits a Non-cooperative Programmable Open System Trust Framework - *NPOST Framework* – for Emerging Systems.

2.2 Game Theory

2.2.1 Introduction

This section briefly introduces the history and significant developments of Game Theory (see Eatwell, Milgate et al. (1989), Schwalbe and Walker (2001), Weintraub (1992), Başar, Olsder et al. (1995), Kuhn (1997) and Smith (1993)) to chronologically (Walker 2013) frame the significant contributions and figures in the field, pertinent to this work.

With its foundations in Economics (Eatwell, Milgate et al. 1989), Game Theory provides a systematic way of analysing problems of strategy. Wherever it is applied, Game Theory develops methodologies applicable in principle to all interactive situations – "Interactive Decision Theory" as an alternative name for it (French 1986).

2.2.1.1 Definitions

This discussion is based on the following definitions:

2.2.1.1.1 Game Theory

1. "Game theory can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers." (Myerson 1997).
2. "Game theory is the science of rational decision making in interactive situations." (Dixit and Skeath 1999).
3. "One of the principle aims of game theory is to provide a mathematical framework for modelling interactions among autonomous and interdependent decision-makers with possibly conflicting objectives and selfish behaviour." (Michalopoulou and Mahonen 2012).

Game Theory then, is a mathematical modelling framework for rational decision making behaviour.

The following definitions are an amalgamation of and elaboration on definitions and descriptions from a variety of sources including Eatwell, Milgate et al. (1989), Schwalbe and Walker (2001), Weintraub (1992), Başar, Olsder et al. (1995), Kuhn (1997) and Smith (1993).

2.2.1.1.2 Agent or Player

Game theory deals with strategic interactions among multiple rational decision makers, called *players* or *agents*.

2.2.1.1.3 (Strategic) Game

A *game* is the predefined structure which defined the bounds and outcomes within which agents interact with one another. Alternatively, a game is conflict involving gains and losses between two

or more opponents who follow formal rules. Games can consist of two (a two-agent game) or many agents (an n-agent game).

These games can be:

2.2.1.1.4 Cooperative

A *cooperative* game is one where the agents of the game are able to act in some agreed coordinated or collaborative, collusive fashion. Agents in a cooperative game attempt to maximise a group of agents' gain, as well as their own.

2.2.1.1.5 Non-cooperative

A *non-cooperative* game is the one where agents act independently, without collusion or the creation of coalitions between agents. Agents attempt to maximise their own gain in isolation through competitive strategic advantage. These types of game are sometimes deemed *selfish*.

2.2.1.1.6 Zero / Constant-Sum

Zero or *constant-sum* is the benefit gained by one agent in a game which is inversely and directly proportional to the loss of others. Agents make payment only to each other.

2.2.1.1.7 Finite

In a *finite* game, each agent has a finite number strategic choices at any stage.

2.2.1.1.8 Single Play

If each agent acts only once, the game is *single play*; otherwise it is a *repeated* or *dynamic* game.

2.2.1.1.9 Static or Simultaneous

Games of this type are where agents make decisions or select a strategy *simultaneously*, without knowledge of the strategies that are being chosen by other agents.

2.2.1.1.10 Dynamic or Repeated

When agents interact by playing a similar stage game numerous times, the game is a *dynamic*, or *repeated* game. Unlike simultaneous games, agents have at least some information about the strategies chosen by others and thus may contingent stratagem on relative decisions.

2.2.1.1.11 Stochastic

A *stochastic* game is a dynamic game with probabilistic transitions. The game is played in a sequence of stages. At the beginning of each stage the game is in some state. The agents select actions and each agent receives a payoff that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the agents. The procedure is repeated at the new state and play continues for a

finite or infinite number of stages. The total payoff to an agent is often taken to be the discounted sum of the stage payoffs or the limit inferior of the averages of the stage payoffs.

Within any game, there can be:

2.2.1.1.12 Information – Complete / Perfect or Imperfect / Incomplete

A game has *perfect* or *complete* information if it is a sequential game where each agent acts in turn, and every agent knows the strategies chosen by the previous agent. If there is strategic information in the game that is not shared amongst all agents, the game has *incomplete* or *imperfect* information.

2.2.1.1.13 Strategy – Pure / Mixed

A *pure strategy* provides a complete definition of how an agent will play a game. In particular, it determines the move an agent will make for any situation.

A *mixed strategy* is an assignment of a probability to each pure strategy. This allows for an agent to randomly select a pure strategy. Since probabilities are continuous, there are infinitely many mixed strategies available to an agent, even if their strategy set is finite.

Pure strategy can be regarded as a degenerate case of a mixed strategy, in which that particular pure strategy is selected with probability one and every other strategy with probability zero.

2.2.1.1.14 Dominant Strategy

Strategic dominance occurs when one strategy is better than another strategy for one agent, no matter what decisions other agents make. The opposite, intransitivity, occurs in games where one strategy may be better or worse than another strategy for one agent, depending on other agent's decisions.

2.2.1.1.15 Payoff

The benefit or utility that an agent receives from playing a game, is called *payoff*.

2.2.1.1.16 Objective / Utility / Cost / Modelling Function or Matrix

Each agent's ordered strategic preference among multiple alternatives is captured (equivalently) in a *utility, payoff, cost, modelling or objective function or matrix*, which the agent attempts to maximise or minimise during a game. These functions map an agent's choices into a real number.

The objective function of an agent depends on the choices of at least one other agent, and generally of all the agents, and hence agents cannot simply optimise their own objective functions independent of the choices of the other agents. There is a coupling between the actions of the agents which binds them together in decision making.

2.2.1.1.17 Nash Equilibria

A stable state of a system involving the interaction of different participants, in which no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged, is said to be in *Nash Equilibrium*.

Alternatively, the optimal outcome of a game is one where no agent has an incentive to deviate from a chosen strategy after considering an opponent's choice. Overall, an agent can receive no incremental benefit from changing actions, assuming other agents retain constant strategies. A game may have multiple *Nash equilibria* or none at all.

2.2.1.1.18 Bargaining Problems

Bargaining problems or games refer to situations where two or more agents must reach agreement regarding how to distribute a utility. Each agent prefers to reach an agreement in these games, rather than abstain from doing so; however, each prefers that agreement which most favours his interests.

Agents in a bargaining problem can bargain for the objective as a whole at a precise moment in time. The problem can also be divided so that parts of the whole objective become subject to bargaining during different stages.

2.2.2 Discussion

2.2.2.1 Bargaining Economics

Game Theory's empirically testable approach formalised and extended the optimisation of economic strategy. Comparable and repeatable results were realisable in large communities of *agents* (or *players*), beyond relatively simple oligopolies. Seemingly impossibly complex problems become solvable (Eatwell, Milgate et al. 1989).

> *"By bargaining we mean negotiations between two or more parties about the terms of possible cooperation, which may involve trade, employment (collective bargaining), a joint business venture, etc." (Eatwell, Milgate et al. 1989).*

Prior to the advent of Game Theory, strategic economic theory amounted to two bargaining rationality postulates between two agents; individual and joint (Harsanyi 1963). In essence in the first case, rational agents consider if the outcome of a bargain is of greater benefit than conflict, and in the second case, rational agents agree on a bargain where there can be perceived no greater benefit to either agent. All perceived possible points of possible agreement or "final settlements", was called the "range of practicable bargains" by Pigou (1905) (Pigou 1924) and later by Luce and Raiffa (1957) (Raiffa 1982), the "negotiation set".

Neoclassical fin de siècle economics offered only what may be deemed a weak bargaining theory because it tells us no more than that the point of agreement between two rational agents lies within some bounds ("range of practicable bargains" or "negotiation set") but it does not tell us where exactly it is or about the economic forces that might influence it (Harsanyi 1963).

It was Zeuthen (1930) who ideated the need for a strong bargaining theory that could predict a unique bargaining point and that it should make consideration for agents' appetite for risk, particularly from conflict rather than accept unfavourable terms.

2.2.2.2 Structure

French philosopher and mathematician Antoine Cournot pioneered the departure from less structured methods for solving these problems by setting out models for describing market duopolies, "…explicitly and with mathematical precision." (Cournot 1838). Cournot described the profit potential of producers or "firms" as functions and used (partial-differential) equations to represent their best responses to given exogenous output levels of the other producer(s) by way of "Cournot Competition" (Van den Berg, Bos et al. 2012).

The simultaneous solutions to these systems of equations reposes in a stable *equilibrium* state where it is not beneficial for any producer to change its output level decision (Morrison 1998).

Cournot's model simplified the producer entities by the fundamental non-conjecture that the producers are economically rational and act strategically, seeking to maximise profit given their competitors' decisions and the assumptions that (Varian 2010):

1. There is fixed number of producers, greater than one and all producers produce a homogeneous product - there is no product differentiation;
2. Producers do not cooperate - there is no collusion;
3. Producers have market power - each producer's output decision affects the good's price;
4. Producers compete in quantities and choose quantities simultaneously.

The advent of *imperfect competition theory* - the situation prevailing in a market in which elements of monopoly allow individual producers or consumers to exercise some control over market prices (Schumpeter and Nichol 1934) (more recently considered by Bonanno (1990)) - in the 1930s, gave rise to many oligopoly theories established on plausible assumptions but with few readily testable (Eatwell, Milgate et al. 1989).

A still inchoate Game Theory benefitted from complete formal descriptions of a game in extensive or "tree" form, and matrix form (Myerson 2013), and was preoccupied with two-person zero-sum

games – games consisting of only two agents where the gain of one agent is to the direct and inversely proportional detriment of the other (Kuhn 1997).

2.2.2.2.1 Strategy – Pure / Mixed

A *pure strategy* provides a complete definition of how an agent will play a game. In particular, it determines the move an agent will make for any situation.

A *mixed strategy* is an assignment of a probability to each pure strategy. This allows for an agent to randomly select a pure strategy. Since probabilities are continuous, there are infinitely many mixed strategies available to an agent, even if their strategy set is finite.

Pure strategy can be regarded as a degenerate case of a mixed strategy, in which that particular pure strategy is selected with probability one and every other strategy with probability zero.

Pure strategies where explicit reactions to situations evolved to consider more rational, non-deterministic *mixed strategies* where there are multiple reactions, each with a likelihood of being carried out (Smith 1993):

- A pure strategy states: “in situation A, always do X”;
- A mixed strategy states: “in situation A, do X with probability P and Y with probability Q” (Smith 1993).

Mixed strategies lead naturally to the concept of *expected utility*. When randomised strategies are used in a strategic game, strict payoff must be replaced by an *expected* return since the game is no longer expected to exhibit discrete outcomes. This payoff utility is considered expected since it cannot be determined for certain since the strategies of the game have probabilistic components – utility can only be determined according to likelihood.

2.2.2.3 Zermelo and Minimax

Two significant theorems were formulated during the early 1900s (Eatwell, Milgate et al. 1989):

Zermelo’s Theorem (Zermelo 1913) – considered the first theorem of Game Theory - (Eatwell, Milgate et al. 1989) asserts that, “in any finite two-person game of perfect information in which the players move alternatively and in which chance does not affect the decision making process, if the game cannot end in a draw, then one of the two players must have a winning strategy.”

Neumann established the *Minimax Theorem* (Neumann 1928) (for contemporary proofs see Schwalbe and Walker (2001) and Van Benthem (2001)) that asserts, “every two-person zero-sum game with infinitely many pure strategies for each player is determined; that is, when mixed

strategies are admitted, it has precisely one individual rational payoff [for each player]" (Eatwell, Milgate et al. 1989).

All of these concepts have endured significantly beyond this period and are cornerstones of game theoretic thought (Eatwell, Milgate et al. 1989).

Von Neumann and Morgenstern's 1944 seminal work (Neumann 1928) marked a return to more empirical methods for economic stratagem conducive to rigorous interpretation. The work is based on prior research by von Neumann, published in 1928 under the German title "Zur Theorie der Gesellschaftsspiele" (Neumann 1928) ("On the Theory of Parlor Games") and exhibits four axioms of rationality – "completeness", "transitivity", "continuity", and "independence" - such that any agent satisfying the axioms has a *utility function*. That is, they proved that an agent is "Von Neumann–Morgenstern [(VNM)] rational" if and only if there exists a (real-valued) function defined by possible outcomes such that every preference of the agent is characterised by maximising the expected value of the function, which can then be defined as the agent's "Von Neumann–Morgenstern-utility". Conspicuously, no claim is made that the agent has a conscious desire to maximise the function, only that it exists.

2.2.2.3.1 Objective / Utility / Cost / Modelling Function or Matrix

Each agent's ordered strategic preference among multiple alternatives is captured (equivalently) in a *utility, payoff, cost, modelling or objective function or matrix*, which the agent attempts to maximise or minimise during the game.

The objective function of an agent depends on the choices of at least one other agent, and generally of all the agents. Hence, agents cannot simply optimise their own objective functions independent of the choices of the other agents. This brings in a coupling between the actions of the agents and binds them together in decision making.

The expected utility hypothesis is that rationality can be modelled as maximising an expected value and can be summarised as "rationality is VNM-rationality" (Neumann 1928).

Applied to Economics, the axiomatisation for subjective expected utility has limited predictive accuracy simply because in practice, humans do not always behave VNM-rationally. This is manifest in the experimental outcomes of the Allais Paradox (Allais 1979) who observes the inconsistency of observed human agent choices with the predictions of theoretical expectations. Allais argues that it is not a realistic reflection of human decision making to consider parts of a decision or gamble in isolation, that all elements of the decision as a whole have to be considered as "complementarities" such that each part-decision has a dependency on all others. This is all part of a "bounded

rationality" (Gigerenzer and Selten 2002) where the rationality of human agents is limited by the information they have, the cognitive limitations of their minds, and the finite amount of time they have to make a decision.

As recently as 2000 Rabin, proved that a VNM-rationally approach to the utility of wealth cannot explain the human agent tendency to avoid loss in preference to acquiring gain, or "loss aversion" (Rabin 2000).

2.2.2.4 Cooperation

2.2.2.4.1 Cooperative Games

Where the agents of the game are able to act in some agreed coordinated or collaborative, collusive fashion.

2.2.2.4.2 Non-cooperative Games

The case where agents act independently, without collusion or the creation of coalitions between agents. Sometimes deemed selfish.

Von Neumann and Morgenstern established the notion of *Cooperative Games* and its coalitional form (Weber 1994) where commitments – agreements, promises, threats – are fully binding and enforceable (Harsanyi 1966). Conversely, if these commitments are not enforceable, and even if pre-play communication between agents is possible, then the game is considered *non-cooperative*.

The entirety of von Neumann and Morgenstern's analysis was applied to *single-play* games. These games are played once and then the agents disperse. Games with multiple rounds or iterations need to be analysed holistically with consideration for all future interactions of the same agents (McCabe, Rigdon et al. 2002).

In the introduction of its 60th anniversary, commemorative edition of "Zur Theorie der Gesellschaftsspiele", the book is described as "the classic work upon which modern-day game theory is based." (John von Neumann 2013).

Eatwell, Milgate et al. (1989) speculate that for the twenty years after 1930, Game Theory failed to realise its promise in Economics because the theory of two-agent constant-sum games - the most advanced area of the field at the time - was the least applicable to Economics, while the better suited games of two or (usually many) more agents (commonly referred to as *n-agent* where "n" is the potentially large number of agents in the game), approaches were yet to be applied.

2.2.2.5 Mathematics and Fixed Points

The mathematics of Game Theory did progress however, particularly through the work of Abraham Wald in the area of Statistical Inference (Wald 1950) (Wald 1942). Wald introduced the process of drawing conclusions from data subject to random variation.

Further significant work was done by Arrow and Hurwicz (1958) (Arrow and Debreu 1954) contributing to the rigorous, axiomatic, and formal analysis of producer behaviour, consumer behaviour, general equilibrium, and the optimality of the market mechanism for resource allocation (Debreu 1959).

The Arrow–Debreu model (also referred to as the Arrow–Debreu–McKenzie model (McKenzie 1959)) suggests that under certain economic assumptions (convex preferences, perfect competition and demand independence) there must be a set of prices such that aggregate supplies will equal aggregate demands for every commodity in the economy. This proved the existence of perfectly competitive equilibrium (Arrow and Debreu 1954). The model is central to the theory of general economic equilibrium and it is often used as a foundational reference for many other microeconomic models (Nicholson and Snyder 2011).

Instrumental in many game theoretical mathematical analysis proofs is the *Fixed Point Theorem* from Kakutani (1941). The theorem makes use of the concept of "upper hemicontinuity" (Börgers 1991) (Rath 1996) (Bianchi and Pini 2003) (Ausubel and Deneckere 1993) to provide sufficient conditions for a set-valued function defined on a convex, compact subset of a Euclidean space to have a fixed point - a point which is mapped to a set containing it. Stated informally, the theorem implies the existence of a Nash equilibrium in every finite game with mixed strategies for any number of agents.

2.2.2.6 Maturity

Game Theory matured significantly during the 1950s (Eatwell, Milgate et al. 1989) continuing to endorse formal methods and develop programmatic approaches to finding Economic strategy solutions. Mathematical developments were made in algebra and in particular, convexity theory (Fenchel 1949).

2.2.2.6.1 John Nash

John Nash, Jr.'s 1950 28-page doctoral dissertation "Non-Cooperative Games" (Nash 1951) contained the definition and properties of what would later be called the "Nash Equilibrium" (NE). It is a crucial concept in non-cooperative games, and won Nash the Nobel prize in economics in 1994 (Nash 1996). Nash is the Game Theory, Founder of Discursivity (Foucault 2013).

An Equilibrium (Nash, 1951) of a strategic game is a (pure or mixed) strategy profile in which each player's strategy maximises his payoff given that others are using their strategies (Eatwell, Milgate et al. 1989).

Nash's major publications relating to this concept can be found in the following papers (Eatwell, Milgate et al. 1989):

- Nash, JF (1950). "Equilibrium Points in N-person Games". Proceedings of the National Academy of Sciences 36 (36): 48–9. doi:10.1073/pnas.36.1.48. PMC 1063129. PMID 16588946., MR0031701 (Nash 1950);
- "The Bargaining Problem". Econometrica (18): 155–62. 1950. MR0035977. (Nash Jr 1950)
- Nash, J. (1951). "Non-cooperative Games". Annals of Mathematics 54 (54): 286–95. doi:10.2307/1969529. JSTOR 1969529 (Nash 1951), and;
- "Two-person Cooperative Games". Econometrica (21): 128–40. 1953., MR0053471 (Nash 1953).

Nash considered the theory of games in the n-agent non-cooperative case in which multiple agents make decisions independently. Thus, while agents could cooperate, any cooperation must be self-enforcing. Nash equilibrium is considered one of the most important and elegant ideas in Game Theory (Myerson 1978) and is fundamental to this work.

Nash Equilibrium is a solution concept of a non-cooperative game involving two or more agents, in which each agent is assumed to know the equilibrium strategies of the other agents, and no agent has anything to gain by changing only their own strategy (Osborne and Rubinstein 1994). If each agent has chosen a stratagem and no agent can benefit by changing stratagem while the other agents keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash Equilibrium.

Removing the two-agent zero-sum game restriction from Zermelo's Theorem (Zermelo 1913) leads to the Nash notion of *Strategic Equilibrium* (Nash 1951).

Moreover, "The fundamental concept of non-cooperative [(in which agents make decisions independently)] n-person Game Theory – the strategic equilibrium of Nash – is an outgrowth of [von Neumann's] minimax, and the proof of its existence is modelled on a previously known proof of the minimax theorem." (Eatwell, Milgate et al. 1989).

> *"Stated simply, [agents] A and B are in Nash equilibrium if A is making the best decision she can, taking into account B's decision, and B is making the best*

decision he can, taking into account A's decision. Likewise, a group of agents are in Nash equilibrium if each one is making the best decision that he or she can, taking into account the decisions of the others." (Osborne and Rubinstein 1994).

2.2.2.6.2 Prisoners' Dilemma

The Prisoners' Dilemma (Poundstone and Metropolis 1992) is a classical example of a non-cooperative game that shows why two agents might not cooperate with an external entity, even if it appears that it is in their best interests to do so. It was originally framed by Flood and Dresher in 1950 (Surhone, Timpledon et al. 2010). Tucker formalised the game with prison sentence rewards and gave it the name "Prisoner's Dilemma", presenting it as (Poundstone 1992):

"Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of speaking to or exchanging messages with the other. The police admit they don't have enough evidence to convict the pair on the principal charge. They plan to sentence both to a year in prison on a lesser charge. Simultaneously, the police offer each prisoner a Faustian bargain [(Peters and Pierre 2004)]. [Faustian Bargaining means:] Each prisoner is given the opportunity either to betray the other, by testifying that the other committed the crime, or to cooperate with the other by remaining silent. Here's how it goes:

If A and B both betray the other, each of them serves 2 years in prison.

If A betrays but B remains silent, A will be set free and B will serve 3 years in prison [(and vice versa)]...

If A and B both remain silent, both of them will only serve 1 year in prison (on the lesser charge)."

It is implied that the prisoners will have no opportunity to reward or punish their partner other than the prison sentences they get, and that their decision will not affect their reputation in future, which completes the setup of the game.

Strictly "cooperation" in this sense, should not be confused with the formal game theoretical concept of a *Cooperative Game* where agents in a game make collaborative strategic decisions to determine allocation of some payoff. In the case of the Prisoners' Dilemma, the agents are able to cooperate *with their captors* (or repudiate responsibility) but not with each other in accordance with the setup of the game – "Each prisoner is in solitary confinement with no means of speaking to or

exchanging messages with the other." The decisions made by each agent are not collaborative and therefore, the game is non-cooperative.

Because betrayal offers a greater reward than cooperation, all purely rational self-interested agents in this situation would betray the other, and so the only possible outcome for two purely rational agents is for them to betray each other (Rapoport 1965).

Pursuing individual reward logically leads both of the agents to betray, when they would get a better reward if they both cooperated. In reality, humans display a systematic bias towards cooperative behaviour in this and similar games, much more so than predicted by simple models of rational self-interested action as analysed by Oosterbeek, Sloof et al. (2004), Ahn, Ostrom et al. (2003), Tversky (2004) and Fehr and Fischbacher (2003). Machines in contrast, can be programmatically directed to act strictly rationally and without regard for others. They can be instructed to act only to maximise individual gain or minimise individual loss, and exhibit purely selfish behaviour, in accordance with strict programmatic procedures. The concerns of Allais (1979) and Rabin (2000) for instance, can be allayed in the machine-to-machine (M2M) case as the mathematical model *defines* the behaviour of the agents in the game and is not an attempt to emulate an external community of agents. The bounds of rationality in the M2M case, are strictly defined.

Presented in canonical form, a generalised Prisoners' Dilemma game exhibits a clear unique (such that there is only one) Nash equilibrium solution. Suppose that the two agents are represented by the colours, red and blue, and that each agent chooses to either "cooperate" or "defect":

- If both agents cooperate, they both receive the reward, R, for cooperating;
- If Blue defects while Red cooperates, then Blue receives the payoff, T while Red receives the payoff, S and vice versa;
- If both agents defect, they both receive the punishment payoff, P.

This can be expressed in normal, canonical form as a *payoff matrix*:

	Cooperate	Defect
Cooperate	R, R	S, T
Defect	T, S	P, P

Figure 1 Normal, canonical form, payoff matrix

and to be a Prisoners' Dilemma game in the strong sense, the following condition must hold for the payoffs:

$$T > R > P > S$$

The payoff relationship R > P implies that mutual cooperation is superior to mutual defection, while the payoff relationships T > R and P > S imply that defection is the *dominant strategy* (where one strategy is better than another strategy for one agent, no matter how that agent's opponents may play) for both agents. That is, mutual defection is the only strong Nash equilibrium in the game - the only outcome from which each agent could only do worse by unilaterally changing strategy. The dilemma then is that mutual cooperation yields a better outcome than mutual defection but it is not the rational outcome because the choice to cooperate, at the individual level, is not rational from a self-interested point of view. Variations of this explanation can be found by Eatwell, Milgate et al. (1989), Schwalbe and Walker (2001), Weintraub (1992) and Başar, Olsder et al. (1995).

2.2.2.7 Repetition

Stochastic and *dynamic* (Başar, Olsder et al. 1995) principles developed games beyond single-play. Games played within some stationary time structure are considered dynamic with a subset of these games – stochastic – where a strategic game is played with the payoff calculated at each time interval and the game to be played at the next interval is determined. The games considered in this work are *repeated* where equilibrium is determined in consecutive rounds of the same game, each round inheriting the results of the previous one to establish a bounded consensus of *trust* (Pakes, Ostrovsky et al. 2007) (Mitchell, Bayen et al. 2005).

Repeated games model the psychological, informational side of ongoing relationships. "Phenomena like cooperation, altruism, trust, punishment and revenge are predicted by the theory" (Eatwell, Milgate et al. 1989).

Employing the ideas from genetics to strategies in games, Axelrod (1987) staged a strategically adaptive Prisoners' Dilemma computer tournament social setting, concluding that the, "…results of the evolutionary process show that the generic [Prisoners' Dilemma] algorithm has a remarkable ability to evolve sophisticated and effective strategies in a complex environment." (Axelrod 1987).

For *cooperative games*, agents can coordinate their strategies and share the payoff in both zero and non-zero sum games. In particular, sets of agents formed as *coalitions* can (Ma, Chiu et al. 2011):

- make binding agreements about joint strategies;
- pool their individual agreements, and;
- redistribute the total payoff in a specified way.

2.2.2.7.1 Nash's Bargaining Problem

Nash's *bargaining problem* (Nash Jr 1950) is an approach to understanding how two agents should cooperate when non-cooperation leads to Pareto-inefficient (a state of allocation of resources in which it is impossible to make any one individual better off without making at least one individual worse off (Barr 2012)) results. It is in essence an equilibrium selection problem.

Unlike the Prisoners' Dilemma which has a unique Nash equilibrium (Kreps, Milgrom et al. 1982), many games have multiple equilibria with varying payoffs for each agent, forcing the agents to negotiate on which equilibrium to target. For any game, it is critical to establish the uniqueness (or otherwise) of an equilibrium state (Nash 1950) so that the ideal strategic solution can be established with certainty, or the bounds of multiple solutions can be understood (Başar, Olsder et al. 1995).

Often the setup of a game includes the definition of the characteristics of the modelling function – concave (Rosen 1965) or (inversely) convex (Gairing, Lücking et al. 2004) for instance, where the uniqueness of a Nash equilibrium can be demonstrated with mathematical certainty (Tan, Yu et al. 1995).

The underlying assumption of bargaining theory is that the resulting solution should be the same solution an impartial arbitrator would recommend. Solutions to bargaining games take two forms (Kalai and Smorodinsky 1975):

1. an axiomatic approach where desired properties of a solution are satisfied, and;
2. a strategic approach where the bargaining procedure is modelled in detail as a sequential game.

The Nash bargaining game is a simple two-agent game used to model bargaining interactions. Two agents demand a portion of some good. If the total amount requested by the agents is less than that available, both agents get their request. If their total request is greater than that available, neither agent gets their request. A Nash bargaining solution is a (Pareto efficient) solution to a Nash bargaining game (Nash Jr 1950).

Nash proposed that a solution should satisfy the axioms (Nash Jr 1950):

1. Invariant to affine transformations [(Begelfor and Werman 2006)] or invariant to equivalent utility representations;
2. Pareto optimality;
3. Independence of irrelevant alternatives, and;
4. Symmetry.

Nash suffused economic and mathematical thinking with his approach to principles of equilibrium and bargaining.

2.2.2.8 Information

Significant work by Harsanyi and Selten (1988), Harsanyi and Selten (1988) and Selten (1965) in the mid-1960s (Eatwell, Milgate et al. 1989) marked the beginning of a very fertile period in the evolution of Economics theory, still very much based on the classic theories of Nash and von Neumann.

In 1957, Luce and Raiffa submitted that the assumption that each agent, "…is fully aware of the rules of the game and the utility functions of each other [agent]… is a serious idealization which only rarely is met in actual situations." (Luce and Raiffa 1957). To address this, Harsanyi (2004) postulated that agents could be categorised by *type* where each type represented a subset of the complete information in the game.

2.2.2.8.1 Information – Complete / Perfect or Incomplete / Imperfect

A game has perfect or complete information if it is a sequential game where each agent acts in turn, and every agent knows the strategies chosen by the previous agent. If there is strategic information in the game that is not shared amongst all agents, the game has incomplete or imperfect information.

This lead to the formal analysis of games of *incomplete information* as a more realistic representation of particularly, large more complex games (Eatwell, Milgate et al. 1989). Further, it is not enough for complete information, for the agents in a game to be fully aware of the rules of the game and the utility functions of other agents. Each agent must be aware of this fact – of the awareness of other agents. Moreover, each agent must be aware of each other agent's awareness. This is known as *common knowledge* and was formulated by Lewis (2008).

The extent of knowledge and information in a game determines how it is to be played. In human situations, this can be difficult to model (Allais (1979) and Rabin (2000)) while in machine (M2M) simulations, it is possible to ensure that all information is made available and to structurally define the types of the information available to agents. The game to be simulated can be setup so that it can be rigorously mathematically described and that all agents share the information as needed. The game is not intended to model an external (human) scenario but is valid of itself.

2.2.3 Summary

The classic work from the beginning of the 1900s of Cournot, Zermelo, Wald, Debreu and Arrow, predominately in Economics, lead to the Nash fundamental theories of games in the 1950s (strategic equilibrium and bargaining problems), with enhancements made by Kakutani and Harsanyi, and significant contributions from von Neumann and Morgenstern to the field. Approaches to contemporary computing of Başar amongst others, led to the recent application of Game Theory to the area of optimal control and resource solutions for distributed computing systems and networks.

More recently, there has been a resurgence of Game Theory because of its application in the field of Computer Science with the advent of Emerging Systems making use of refined fundamental theories to examine the efficient routing of network traffic and security (Eatwell, Milgate et al. 1989). All the information pertinent to the game can be well defined within the framework of a strategy problem. Game Theory is used to dictate the behaviour of a system rather than try to model the behaviour of some far more complex, exogenous system.

Contemporary approaches to resource allocation and security are underpinned by Nash's principles of cooperative bargaining to determine the optimal topology and routing strategy within computing systems and networks. "Nodes" (network elements and the like) are treated as agents in a structured game, and equilibrium solutions determined to identify the most efficient and fault tolerant way to make use of resources available to the system. The possibility of collaborative solutions to these problems is when systems are "closed" such that there is some central authority with jurisdiction over the system. This approach is also common when trying to identify acts of malice within a system and potential security breaches from outside. Exploration by Başar, referenced extensively in this work, (Başar, Olsder et al. 1995) in the fields of distributed computing, wireless and communication networks (Han, Niyato et al. 2012), and control systems (Başar and Bernhard 2008), has extended the application of Game Theory.

The next section reviews recent literature concerned with Emerging Systems. The section establishes a definition of an Emerging System as an extension of mobile ad hoc networks for the benefit of this work, and considers the characteristics of Emerging Systems and in particular the highly reprogrammable nature nodes that make them up.

2.3 Emerging Systems

2.3.1 Introduction

The previous section provided an historical context for the development of game theory and introduced the significant contributors and results in the field.

This section reviews current literature in order to establish a definition of Emerging Systems. The definition is derived from the concept of Mobile Ad-hoc NETworks (MANET) with the significant differentiating feature being the nature of nodes within the system. This section describes the characteristics of Emerging Systems and how they differ from "closed" systems with an emphasis on the highly reprogrammable capabilities of the nodes.

2.3.1.1 Definitions

This discussion is based on the following definitions.

2.3.1.1.1 Mobile Ad hoc Network

The concept of a MANET (Mobile Ad-hoc NETwork) (Murthy and Manoj 2004) has attracted assorted definitions with a high degree of commonality, not limited to:

> *"… a collection of nodes with no infrastructure while its nodes are connected with wireless links. Nodes in the network are capable to sense and determine nearby nodes. They communicate by forwarding packets hop by hop in the network." (Gowthami and Buvaneswari 2013).*

> *"… is the cooperative engagement of a collection of Mobile Hosts without the required intervention of any centralized [sic] Access Point." (Perkins and Bhagwat 1994).*

> *"…is a collection of autonomous, self-organized [sic], mobile wireless nodes." (Jiang and Baras 2004) (Jiang and Baras 2004) (Jiang and Baras 2004) .*

> *"…[is a] multi-hop system comprised by multiple mobile wireless nodes with peer-to-peer relationships." (Xia, Jia et al. 2011).*

Characteristics of ad-hoc networks according to Weimerskirch and Thonet (2002) include:

- Communication links are wireless to guarantee mobility. Accordingly, their dependability and capacity have to be carefully scrutinised;
- Ad-hoc networks act independently from any provider. However, access points to a fixed backbone network are expected to be available if required;

- Because they do not rely on a fixed infrastructure mobile hosts have to be somehow cooperative. This ranges from simple schemes for short range networks to high cooperative strategies in the case of multi-hop wide-area networks;
- The network topology may be very dynamic, making the links and routes very unstable.
- Power management is an important system design criterion. Hosts have to be power-aware when performing such tasks as routing and mobility management;
- Finally, security is a critical issue because of the weak connectivity and of the limited physical protection of the mobile hosts.

In such a self-organised network each node relies on its neighbour nodes to keep the network connected. Furthermore each node might take advantage of the services offered by other nodes. (Weimerskirch and Thonet 2002).

Eissa, Razak et al. (2013) note the unsuitability of fixed infrastructure for mobile node collaboration:

> *"Infrastructure networks are not suitable in environments where limited resources devices are connected through weak wireless links. In this case, the network should be able to setup on-the-fly without the aid of any administrator or manager. … [MANET] is a self-organizing and self-configuring network. It is established on a temporary basis and nodes can join or leave the network at any time." (Eissa, Razak et al. 2013).*

> *"[MANET] does not require any fixed infrastructure to be configured which makes it more suitable to be used in environments that require on-the-fly setup." (Eissa, Razak et al. 2013).*

2.3.1.1.2 Emerging Systems

Having explored the definitions of MANET and established their characteristics, we are now able to define Emerging Systems formally.

Emerging Systems **are characterised by:**

- **Decentralisation;**
- **High distribution;**
- **Self-configuration;**
- **Self-regulation;**
- **Non-cooperation;**
- **Pervasiveness;**

- **Dynamic open ad-hoc ("for this" purpose) topology – non-generalisable;**
- **No fixed infrastructure;**
- **Wireless connectivity;**
- **High scalability, and;**
- **Consisting of highly reprogrammable nodes.**

The motivation for defining a "system" rather than a "network" is that the nodes in Emerging Systems are by definition, programmatically advanced lending themselves more to identification as the components of a system than traditionally, more simple data conveying elements of a network. A high level of computation can take place on these nodes and complex relationships can be established with other component nodes within the system.

Static nodes are trivial cases of mobile nodes and will be treated implicitly as such.

The use of the term "emerging" in this work, deliberately elicits a threefold implication:

1. that the systems are formed from groups of previously unrelated nodes without structure;
2. that the systems are evolving into something to be established as an "Internet of Things" (IoT) (Ashton 2014) (Yan, Zhang et al.) referring to uniquely identifiable objects and their virtual representations in an Internet-like structure, beyond a "network", and;
3. that the diversity of the manifestations of these systems such as social media and distributed semantic knowledge-bases where, as active participation is encouraged beyond passive consumption, diverse unstructured collaboration develops (Berners-Lee 1989) (Tim, James et al.).

Snowden and Boone (2015) describe "emergence" where a complex system, "…is dynamic, the whole is greater than the sum of its parts, and solutions can't be imposed; rather, they arise from the circumstances" within the context of understanding "complexity" in a scientific business context. This description resonates well with the concept of an Emerging System in this work.

2.3.2 Discussion

Closed systems can have central trust and identity mechanisms that control, track and monitor node associations and interactions. Corporate networks are a prime example of where system control through a single authority is possible since all nodes share the same services and network infrastructure, and therefore, can have similar trust policies enforced universally for all identifiable members of the system to control access levels. If a node is not identified, it can be denied fully or permitted restricted access to the system.

Further, in closed systems, nodes are often configured to function harmoniously. Restrictions can be enforced within the system that ensure that there is a balanced allocation of resources and that no single node can act selfishly or to the determent of any other. The sum total of resources available to the system is known and can be allocated suitably. Nodes can cooperate with each other under a shared agreement of service to the system. Emerging Systems exhibit dissimilar characteristics; they transcend networks and, mobile and fixed systems. Not only are nodes present on different networks and systems, they also roam between them often during transactional sessions, "[d]ue to the openness in network topology and the absence of centralized [*sic*] administration management [of Emerging Systems]." (Xia, Jia et al. 2011).

Resources available to the Emerging System are unknown and usually highly variable. No central mechanism of authority or control for trust and identity is possible. Nodes cannot be assumed cooperative and should be assumed selfish.

Emerging Systems have two primary advantages over closed systems; they are highly scalable and inherently fault tolerant. Shastri, Patil et al. (2013) consider cloud computing essentially as an Emerging System and its use as part of mobile networks to reduce mobile device limitations of data storage and processing power (Qian and Andresen 2015). In this infrastructure paradigm, larger or more complex tasks are "offloaded" to larger servers that carry out the work and then return the result to the device. Resources are not strictly delimitated or allocated so that they can be shared between mobile devices as required. Cloud and trust are further considered by Firdhous, Ghazali et al. (2014) in an effort to evaluate the Quality of Service (QoS) of a cloud system. They propose a robust multi-level computing mechanism that can be used to track the performance of cloud systems using multiple QoS attributes, based on trust. Similar analysis is carried out by Sanchez (2013) in defining a risk and trust framework for pervasive mobile environments like Emerging Systems.

Emerging Systems are able to flexibly vary their capacity depending on the resources that are available to them. Often considered as the ability of a system to support resource growth without impact to performance, an Emerging System is also able to cater for a reduction in resources.

Rather than assuming that failures and disasters will be the exception, these systems are designed assuming the worst will happen. The principles and protocols assume that failures are the rule rather than the exception. A key consideration of distributed systems is the need to maintain consistency, availability and reliability (Trifunovic, Kurant et al. 2014).

CAP Theorem postulated by Brewer (2000) states that a system can only provide any two of consistency, availability and partition tolerance fully, where:

- Consistency means all nodes see the same data at the same time;
- Availability is a guarantee that every request receives a response about whether it was successful or failed, and;
- Partition tolerance is such that the system continues to operate despite failure of part of the system.

Emerging Systems do not have a Single Point of Failure (SPoF). For a closed system, if the central identity and trust authority fails, the whole system is rendered inoperable. With a distributed system, there is no single dependency point; all nodes have to fail for the system to cease to function completely. This makes the system highly fault tolerant and resilient to changes in the topology and fluctuating node membership within it. Nodes can function in clusters and even in isolation.

The success of an Emerging System on these terms depends on the reliability of the members that make it up and their ability to function harmoniously. Nodes that are members of the system have to be reliable themselves and have compatible incentives.

Emerging Systems have nodes that are very prone to outages, high latency and quickly diminishing resources. The system itself must be resilient enough to recover from situations that arise from these limitations gracefully.

The nature of Emerging Systems, requires consideration be made that trust needs to be managed in a complementary fashion. This means that a trust mechanism for an Emerging System needs to be:

1. distributed in nature such that it does not hinder the scalability or fault tolerance of the system it supports, and;
2. it should assume that nodes within the system are strictly non-cooperative.

2.3.2.1 Highly Reprogrammable Mobile Nodes

When discussing mobile devices as highly reprogrammable nodes, their nature needs to be understood. According to Kovacs, Robrie et al. (2006) and Young (2009) mobile nodes are characterised by:

- Small screen sizes (limited screen real-estate);
- Restrictive input mechanisms;
- Limited processor power;

- Limited storage;
- Limited power capacity;
- Fluctuating network connectivity – unreliable network connectivity, connection loss and service termination;
- Narrow bandwidth – slow transfer of data to the device from the network;
- Expensive and unpredictable data traffic cost;
- Vastly differing software between devices.

Many of these characteristics improve as devices become more powerful and capable over time, while the cost of use is mainly a business concern. Despite continuing advances in infrastructure, "...mobile communication will remain costly, unreliable, and different from communication over fixed networks" (Kovacs, Robrie et al. 2006).

The primary differentiating capability of mobile devices is pervasiveness (Saha and Mukherjee 2003). It facilitates roaming communication (data and voice) and location services. Retaining this capability, necessitates the "negative" characteristics; for example, making a device's screen larger makes it easier to read but increases the space required to transport it and reduces mobility.

Taking CPU (the Central Processing Unit is responsible for the calculations carried out by a computing device) speed as indicative of the progression of computing devices, Intel CPUs achieve clock speeds of near 4GHz with "Extreme", "Xeon" and poly-core varieties for wired devices. Intel Corporation (Intel 2013) introduced their 386 SL processor specifically to support portable devices in 1990. Currently they produce "Atom" processors for Mobile Internet Devices (MID) that reach speeds approaching 2GHz, the same speed common in desktop machines in 2001.

Conceding the application for top-end CPUs is server machines (large computers designed to provide a service over a network) and that current poly-core CPUs support parallel processing instead of just increased clock speed, mobile device CPUs are becoming increasingly comparable to desktop computers (Wang, Lin et al. 2014) (Rodriguez, Mateos et al. 2014) (Chou, Liu et al. 2014). Assuming a similar convergence in other facets of mobile and desktop capabilities (storage and RAM) (Shiraz, Ahmed et al. 2014), the outstanding mobile node limiting characteristics will be screen size, input mechanisms (Lai and Wu 2014), network connectivity and power consumption (Malm, Jani et al. 2003). This work makes consideration for the last two as significant factors that contribute to the strategic problems of Emerging Systems. Because wireless connectivity is unreliable and topological volatility without a central authority, robust approaches need to be developed to ensure the integrity of these systems (Murthy and Manoj 2004). Similarly, power consumption and resource

allocation in general, need to be distributed effectively within an Emerging System (Conti and Giordano 2014).

The characteristics of mobile devices present unique challenges as constituents of Emerging Systems. Kanoc (1999) identifies the main challenges as threefold:

1. The wide variety of wired and wireless networks available, many of which have nonstandard, complex protocols;
2. The variety of devices, which incorporate numerous mobile operating platforms, across which an application must run;
3. The need to communicate with roaming workers who move in and out of network coverage, who switch between different devices/networks to meet different needs and who operate in a disconnected fashion.

To meet these challenges, Kanoc (1999) also identifies that the technological considerations to extend the enterprise into the mobile field would have to address:

- Security – information transported over many networks with different ownership as it is for mobile devices to permit them to move freely are difficult to regulate. Consequently, sending potentially sensitive business information poses security concerns about the integrity of the information and the potential for unauthorised access to the enterprise;
- Scalability – there are ever increasing numbers of mobile consumers. According to the Australian Communications and Media Authority (Acma 2014), there are now more mobile device services in Australia than people, steadily increasing each year. While the enterprise business user is more esoteric, Emerging Systems should be capable of supporting ever increasing numbers of consumers over time;
- Reliability – particularly in regard to network connections, the networks would be used for business critical functions that could prove highly detrimental to business processes should they be interrupted or fail;
- Ease of integration – to adapt current most static technologies to mobility should be designed to be as easy as possible to promote adoption;
- Multiple network and platform support – static consumers usually connect to a single network and with now common standards and software. This is not the case for mobile devices. There are many different devices with different software and manufacturers, and to facilitate perverseness, they connect over many varied networks that can change even during a network transaction.

Mobile nodes are capable of supporting fully reprogrammable applications (Shah, Jan et al. 2012). Program developers are able to create fully functional network enabled, database driven, graphically rich applications that form Emerging Systems over the shared resources of which, some computational work can be carried out. The work could be social sharing (human distribution), map-reduce calculations (Dean and Ghemawat 2004) (machine distribution) or any other arbitrary computational load.

The applications exist in a closed environment with controlled access to the device node's resources and services. This constitutes an application layer supported by services surfaced by the operating system. Permission (human agreement or certification) is usually required for access to services that manage sensitive information – contacts, call logs, access credentials – or to resources that have a potential fiscal cost associated with them – network access, messaging, application updates and upgrades (Google 2014), (Apple 2014).

Single authority control and node cooperation cannot be assumed between highly programmable mobile nodes in Emerging Systems and:

> *"[t]he increased capability of reprogrammability[sic] of wireless devices offers another threat to this assumption. It is, therefore, important that the issues in networks like … MANET's should be addressed by using the concepts from non-cooperative game theory." (Shah, Jan et al. 2012).*

2.3.3 Summary

This section established the characteristics and a definition of Emerging Systems. It explored the nature of the highly reprogrammable nodes that make up Emerging Systems as the differentiating characteristic from MANET's, their characteristics and problems they introduce into a system.

Shah, Jan et al. (2012) clearly acknowledge the need to consider the capabilities of highly reprogrammable nodes and their capacity for selfish or malicious behaviour, beyond the network layer. It is impossible to determine what software is being used on a node or how it has been devised to behave within an Emerging System. Shah, Jan et al. (2012) go on to advocate a specifically *non*-cooperative game theoretical approach to the distributed management of Emerging Systems, as adopted in this work and applied to trust. A *trust layer*, conceptually located beneath the application layer incorporated into the application stack is proposed to manage trust.

The next section explores the concept of "trust" and its role in Emerging Systems. It provides a definition of trust for the purposes of this work. Through the literature, trust frameworks are

examined and compared to the framework proposed here. The section expands on the principle of a "trust layer" for highly reprogrammable nodes in an Emerging System.

2.4 Trust and Emerging Systems

2.4.1 Introduction

The previous section established a definition of Emerging Systems and described their characteristics.

This section extends the previous discussion to consider "trust" and how it is applied in Emerging Systems. A formal definition of trust is established, a trust layer for highly reprogrammable nodes is described and "trust frameworks" are explored through related literature.

The "trust" concept is derived from Social Science and is defined as the degree of subjective belief about behaviours of a particular entity (Capra 2004). Interpersonal trust is an expectation about a future behaviour of another person and an accompanying feeling of calmness, confidence, and security depending on the degree of trust and the extent of the associated risk. That other person shall behave as agreed, not agreed but loyal, or at least according to subjective expectations, although she has the freedom and choice to act differently, because it is impossible or voluntarily unwanted to control her. That other person may also be perceived as a representative of a certain group (Bamberger 2010).

Trust is a fundamental factor that influences decisions pertaining to human interactions, be they social or economic in nature. Trust enables humans to accept risks and deal with uncertainty (Tavakolifard and Almeroth 2012). Mayer, Davis et al. (1995) offer a definition of trust as "...the willingness to be vulnerable, based on positive expectation about the behaviour of others." These expectations of the trustor would be informed by trust signals exchanged with the trustee of a planned interaction. Trust has an economic incentive, it avoids the use of costly measures that guarantee assurance in the absence of trust-enabled interaction (Huth and Kuo 2014).

> *"Trust is a general level of confidence in a person or thing." (Thooyamani, Udayakumar et al. 2014).*

Defined by Golbeck (2006), the properties of trust are transitivity, asymmetry and personalisation in a MANET:

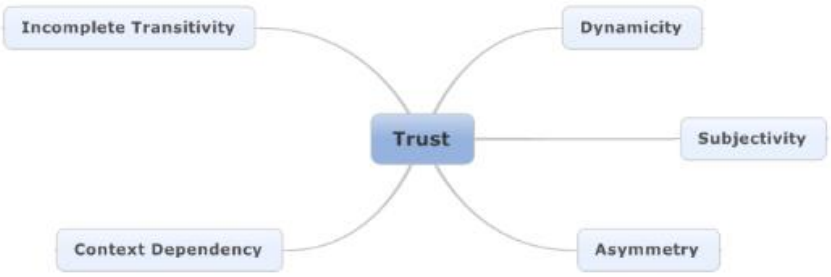


Figure 2 Golbeck (2006) properties of trust

Eschenauer, Gligor et al. (2004) define trust as, "…a set of relations among entities that participate in a protocol. Such relations are based on evidence created by earlier interactions within protocol entities. Generally, if the interactions are faithful to the protocol, then the trust is more between such entities." The matter considered here, is how much trust should we have (Lucy 2014)?

In regard to the application of trust for Emerging Systems: various engineering models such as security, usability, reliability, availability, safety and privacy incorporate some limited aspects of trust with different meanings (Thooyamani, Udayakumar et al. 2014).

The approach is based on the way that human agents establish relationships and establish levels of confidence about the relationship between themselves. When a person wants to verify another person, he usually asks his friends about this person. He also asks this person to provide him with the list of referenced people who will be asked if he is to be trusted (Eissa, Razak et al. 2013).

The competent provision of a service by a node excludes malice. Malice cannot be considered a desirable constituent of a competent service from the perspective of the consumer, by definition.

Specifically for Emerging System reliability, "…trust is used as a measure of node's competence in providing the required service." (Thooyamani, Udayakumar et al. 2014).

Resource allocation is based on trust which is dependent on reputation. A reputation level is computed for each node by other nodes and shared within the system. There can be further external influential factors such as incentive and the environment. These combined elements, constitute a community trust level for any participating mobile node. The most reliable and efficient way to carry out some distributed work is assumed to be the collaboration of the most trusted and reputable nodes in the system.

2.4.1.1 Definition

This discussion is based on the following definition.

2.4.1.1.1 Trust

For the purposes of this work, we take the definition of trust to be a relationship between two neighbouring entities where a trust value expresses the degree that one entity expects another entity to offer certain services. The reputation of an entity, is the record of the trust values attributed to an entity by consensus of other entities.

2.4.2 Discussion

Blaze, Feigenbaum et al. (1996) introduced the term "Trust Management" identifying it as a separate component of a system. Trust management is required in Emerging Systems when participating nodes form a relationship among themselves, potentially without earlier interactions.

Thooyamani, Udayakumar et al. (2014) identify the general benefits of establishing trust in a system as follows:

- Trust solves the problem of providing corresponding access control based on judging the quality of the [nodes] and their services. This problem cannot be solved through the traditional security mechanisms.
- Trust solves the problem of providing reliable routing paths that do not contain malicious, selfish, or faulty node(s).
- Trust makes the traditional security services more robust and reliable by ensuring that all the communicating nodes are trusted during authentication, authorization, or key management.

To establish trust in a system, in this work we appeal to a framework that underpins how trust is evaluated and communicated.

The deployment of a global computing infrastructure raises new and difficult security and privacy issues. Traditional security mechanisms are of questionable effectiveness in the new global computing era. Part of the reason is that no common infrastructure can be assumed to enforce any notion of correct behaviour, in part because even defining a common and acceptable standard is impossible. No single authority can define and enforce rules, and therefore, online interactions cannot be governed by common rules as before (Tavakolifard and Almeroth 2012).

2.4.2.1 Layered Models of Interconnection

To give context to our discussion of a proposed "trust layer", we consider two related layer models that describe how machines are interconnected and sets of standards for doing so. They are often called the "5-layer" or "Internet TCP" and "7-layer" or "Open Systems Interconnection (OSI)" (Day and Zimmermann 1983) (William, Desmond et al. 2007) (Zimmermann 1980) models.

Comparatively, the models exhibit complementary strengths and weaknesses. The OSI model has an emphasis on layering but its protocols are apparently weakly defined. The TCP protocols are well defined, while the TCP conceptual model appears unsubstantial. The origin of these models goes some way to explaining this.

From the literature, there are numerous alternative approaches to these models, varying between three and seven layers forming the basis for or derived from, the TCP and OSI models, including:

- RFC 112, Internet STD 3 (1989) (Braden 1989);
- Cisco Academy (Dye, McDonald et al. 2007);
- Kurose (Kurose and Ross 2007), Forouzan (Forouzan and Fegan 2003);
- Comer (Comer 2006), Kozierok (Kozierok 2005);
- Stallings (Stallings 2007), and;
- Tanembaum (Tanenbaum 2003).

Some of these are secondary sources that may conflict with the intent of RFC 112 (O'Sullivan 1971) primary sources.

2.4.2.1.1 5-Layer Internet TCP Model

Internet TCP Model		
Layer	**Name**	**Function**
5	Process and Applications	Provides application services to users and programs.
4	Transport	Handles data-consistency functions.
3	Internet / Network	Provides network addressing and routing.
2	Network / Data Link	Internet protocol.
1	Physical	Hardware.

The TCP model has five layers with its origins in American military. The protocols preceded the model and is considered a "bottom-up" approach which does little to distinguish the concepts of protocol, interface, and service. These weaknesses are partially based on the Advanced Research Projects Agency Network (ARPANET) (Salus and Vinton 1995) assumptions that network users are technical experts with great programming sophistication.

2.4.2.1.2 7-Layer OSI Model

OSI Model				
Layer			**Protocol Data Unit (PDU)**	**Function**
Host Layers	7	Application	Data	High-level API's, including resource sharing, remote file access, directory services and virtual terminals.
	6	Presentation		Translation of data between a networking service and an application.
	5	Session		Managing communication sessions.
	4	Transport	Segment (TCP) / Datagram (UDP)	Reliable transmission of data segments between points on a network.
Media Layers	3	Network	Packet	Structuring and managing a multi-node network.
	2	Data Link	Frame	Reliable transmission of data between two nodes connected by a physical layer.
	1	Physical	Bit	Transmission and reception of raw bit streams over a physical medium.

The genesis of the OSI networking model is European telephony. It has seven layers and is the result of international deliberation that formulated the model before there were protocols to support it. This approach emphasised existing proprietary software protocols and relied on telephony for its conceptual foundation with little consideration for computing.

2.4.2.1.3 Model Comparison

Despite using a different concept for layering from the OSI model, a comparison can be made with the TCP model's layers in the following way (Goralski 2009):

- The Internet application layer includes the OSI application layer, presentation layer, and most of the session layer;
- Its end-to-end transport layer includes the graceful close function of the OSI session layer as well as the OSI transport layer;
- The internetworking layer (Internet layer) is a subset of the OSI network layer, and;
- The link layer includes the OSI data link layer and sometimes the physical layers, as well as some protocols of the OSI's network layer.

2.4.2.1.4 Trust Layer

It is beyond the scope of this work to propose a complete alternative layering standard. The definition of these standards is varied with strong affinity to the domain to which the standard is applied, attested by the origins of the OSI and TCP layer models. While there are similarities between these models, they are not consistent and are the product of different approaches. There are numerous other models proposed with a range of layers and variable assimilation with standards authorities. All of this considered, a new layer or potentially the explicit extension of responsibility

of an existing layer in the OSI or TCP layer models, is proposed that can be configured to manage trust for applications installed on mobile nodes within a distributed system.

For the OSI model, the Trust Layer would exist in the Data PDU:

- While it does not service the high-level API and resource sharing function directly, it contributes to the decisions concerning consumers that should be serviced from the Application Layer. In the OSI model, applications conceptually sit above the Application Layer as the physical material resides below the Physical Layer.
- It could be argued that trust should be a consideration when translating data between network services and the application, ensuring that the service is trusted. The basis of this thesis is that an application-centric Trust Layer is needed therefore, this should be higher than the network level.
- Lower layer levels are beyond the intended application of the Trust Layer proposed here.

For TCP, Layer 5 is broader in responsibility than for OSI, incorporating Process and Applications, both of which, it can be argued, that the Trust Layer should be directly supporting.

In both cases and despite the inconsistency between the definitions of layers for OSI and TCP, the Trust Layer proposed exists immediately below the layer responsible for application support and above the transport layer.

The Trust Layer manages the distribution of trust and reputation profiles between nodes, as well as resolving conflicts and outages. It is “just-in-time” consistent and available, meaning that this type of distribution and synchronisation does not subscribe to the strict integrity constraints of traditional closed data storage and can be considered, “relaxed”. It makes use of open standards and protocols ensure device node agnosticism to overcome the differing device software support (Sanchez 2013).

An implementation of the layer is present on every node in the system ensuring that the application remains partition tolerant and scalable. A node can continue to hold an opinion in isolation.

This layer is based on a non-cooperative Game Theoretical (Eatwell, Milgate et al. 1989) linear algebraic (Strang 2003) mathematical framework that garners the opinions of nodes within the system to establish a community consensus that constitutes a node’s reputation. This reputation can then be used by a mobile application to make decisions about which nodes are most trustworthy and therefore, suitable for collaborative interaction.

Non-cooperative means that it is assumed that nodes are not privy to each other’s motivations and that each node will make decisions in isolation that best serve its purposes.

2.4.2.2 Trust Frameworks

Rajesh and Kumar (2014), propose a trust architecture for MANET's. The architecture explicitly supports three levels of trust – low, medium and high – each adopting a more stringent sensitivity to context-derived "functional unit" (Rajesh and Kumar 2014) observations than the last. Each trust level, incurs an increased computational overhead and cost as a result. Rajesh and Kumar (2014), propose applications for the different levels of trust:

Trust level	Functional units	Computational cost	Overhead	Application
Low	Direct observation.	Low	Low	Local level (Home networks).
Medium	Direct observation and recommendation.	Medium	Medium	Collaborative work (Office environment).
High	Direct observation, recommendation and reputation (second hand opinions).	High	High	Military application.

Table 1 Applications for different levels of trust

The architecture is integrated with a geographic routing layer, Position-based Optimistic Routing (POR) (Yang, Yeo et al. 2012) protocol, "…which takes advantage of the stateless property of geographic routing and the broadcast nature of wireless medium." (Yang, Yeo et al. 2012). This work aims to establish a similar framework but with the emphasis on the application layer of the stack or as a separate "trust layer" directly below the application layer. The trust layer supports the application layer by removing the responsibility for trust decisions from the application.

For the high trust level case, Rajesh and Kumar (2014) make use of a weighted-average technique which derives a value from "one hop" locality opinions to establish the reputation of a node – "…reputation (second hand opinions)" (Rajesh and Kumar 2014). Reputation tables are held by each node and thresholds are defined to determine bounds for the suitability of a node as the "next hop", to be sent a data packet. This is a simplistic approach to determining a community opinion consensus as a reputation. It does not cater for complex negotiation (cooperation for instance) or stratagem (selfishness). If we establish the consensus of reputation as a game-theoretical "game", we can apply techniques and tools to determine more insightful reputation values. The Rajesh and Kumar (2014) systems would have to be modelled as trivial complete information and non-cooperative. Further, the framework proposed in this work allows levels of trust within the system to be more granularly enforced beyond three tiers and with consideration for more environmental

and context factors, and with a greater descriptive model of a node's interests and motivation. This more complex model is much better suited to describing the highly-programmable nodes that make up Emerging Systems because of the complexity of the programs that can be run on them.

As Rajesh and Kumar (2014) observe, the more complex the mechanism is for determining the reputation of nodes, the higher the computational overhead is. This is absolutely the case for a game theoretical approach to determining reputation and should be consider when deciding on an implementation of the approach. Large, complex systems that establish diverse relationships between nodes, as Emerging Systems tend to be due to their characteristics, can incur very high computational overheads. It is imperative to determine the capacity of a trust framework to support systems experimentally as Rajesh and Kumar (2014) do to establish that their proposed architecture increases packet delivery ratio and throughput.

In an effort to improve resource consumption (power and memory) and mitigate the threat of malicious nodes in a Wireless Sensor Network (WSN) – "…spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location" (Yang 2014) - Thooyamani, Udayakumar et al. (2014) propose a trust management schema based on direct and indirect interactions with neighbouring nodes. The military application of the schema makes use of the Ad-Hoc On-Demand Vector (AODV) communication protocol where nodes are self-organising and gather event information from their surroundings while unattended.

Thooyamani, Udayakumar et al. (2014) advocate the use of trust to establish the quality of the information relayed by a node in the system represented by a trust value within a range determined by the interaction events between nodes. There is a "direct" trust value determined by the events directly encountered between nodes, and an "indirect" trust value which is derived from the experience of neighbouring nodes. Nodes with the lowest trust value are considered at least, inefficient and potentially malicious and removed from any routing path for packets of data within the system.

Related to the Bellmann-Ford (Perkins and Bhagwat 1994) distant vector algorithm, AODV determines a route to a destination only when a node wants to send a packet to the destination. Routes are maintained as long as they are needed by the source. The AODV protocol is used by the Thooyamani, Udayakumar et al. (2014) trust framework to exchange trust values between nodes in the system efficiently.

The Thooyamani, Udayakumar et al. (2014) framework is designed to work at the protocol level of the system stack where a broadcast approach is used to maintain routing tables for nodes in a system. Trust values are determined by the success or failure of a node to forward a packet of data without modification. This approach is similar to the one proposed in this work in that a broadcast is used to establish individual opinions of nodes in close proximity and a record is kept of (a consensus of) opinion, but we do not consider the protocol level itself. The mechanism for the dissemination of trust values is not considered explicitly and is left to the implementation.

Xia, Jia et al. (2013) propose a "Trust-based Source Routing" (TSR) protocol to provide a flexible approach to choosing the shortest route through a system while adhering to the predictions of their fuzzy logic rules prediction method (Xia, Jia et al. 2013). Again, an adaptation of Ad-Hoc On-Demand Vector (AODV).

While it is not apparent how trust values are determined in the Thooyamani, Udayakumar et al. (2014) framework, more how they are distributed, they make some noteworthy observations regarding Bayesian game-based (Harsanyi 1967) reputation calculations to determine a stable consensus state:

Reputation-based Framework for Sensor Networks (Huang, Kanhere et al. 2014) assumes that the node has enough interactions with the neighbours so that the reputation can reach a stationary state. However, if the rate of node mobility is higher, the reputation information will not stabilise (Thooyamani, Udayakumar et al. 2014).

This is an important consideration for this work as we intend to use a game theoretical approach to establish a stable consensus of trust values to determine a reputation for nodes. It is necessary to prove that the approach to the calculation of these values is well suited to the characteristics of Emerging Systems.

Contrary to the title's implication, it appears that the nodes within the systems considered by Thooyamani, Udayakumar et al. (2014) are not actually collaborating to provide a suitable solution to the routing problem. It can be argued that each node is assumed to be selfish in its behaviour in some cases so much so, that a node can be deemed malicious. As with the work proposed here, considered as a game-theoretical "game", the systems considered are non-cooperative and of complete information. Not necessarily for trust information but for the sensor information that is being gathered, the systems exhibit a central point to which information reporting is directed. This could potentially be considered a centralised authority, moving us away from our pure definition of an Emerging System.

Trust-based security mechanisms have emerged as a solution that expands the scope of traditional security models (Tavakolifard and Almeroth 2012).

Some of the challenges for Emerging Systems have their roots in the subjective nature of feedback and some are related to the ease with which online identities can be attacked. “Before online reputation systems will be accepted as legitimate trust solutions, a better understanding is needed of how such systems can be compromised and how these problems can be solved.” (Tavakolifard and Almeroth 2012).

These mechanisms provide weaker security guarantees, but serve greater application areas. Online environments such as the Internet, search engines, peer-to-peer networks, and new applications built on highly complex social networks introduce several challenges in the interpretation and use of online trust and reputation systems (Tavakolifard and Almeroth 2012).

Eissa, Razak et al. (2013) address the security challenges for MANET’s, adopting and testing a trust and identification-based schema over the Ad-Hoc On-Demand Vector (AODV) communication routing protocol, called “Friendship-based” AODV (FrAODV).

The implementation of this schema is proposed as a complementary layer to traditional security approaches: the traditional cryptography schemes that provide authentication and data privacy do not detect when an internal node provides false routing information, or where a node does not cooperate with the other nodes to save its resources. There should be a layer of security that detects such misbehaviour. This layer is based on the trust concept (Eissa, Razak et al. 2013).

This is a similar principle to the work proposed here but is concerned with network elements and protocol. It describes the implementation as metaphorical layer in the same way as the trust layer is proposed in this work, below the application layer for highly-programmable nodes in Emerging Systems. The approach was first proposed by Weimerskirch and Thonet (2002) who proposed a security model for low-value transactions in ad-hoc networks with the emphasis on authentication since they consider it to be the, “…core requirement for commercial transactions.” (Weimerskirch and Thonet 2002).

Eissa, Razak et al. (2013) propose that each node in the system retain a list of “friends” and “friendship” values for these friends represented numerically within some value range with thresholds that identify “unfriendly” nodes - analogous to the concept of trust and trust values. The consideration of each node as a suitable inclusion in a network route to some destination is determined based on the cumulative friendship values attributed to it, by its neighbouring nodes. Friendship values are established via two algorithms using an averaging formula. The simplicity of

the approach is appealing and lends itself well to fast, “light-weight” situations but is not so appropriate for more complex situations potentially strategic ones, more often seen in applications (where this work is concerned).

Subramanian and Ramachandran (2012) augment the Ad-Hoc On-Demand Vector (AODV) communication routing protocol, with a trust management framework to produce Trust Based Reliable AODV (TBRAODV). The approach is similar to Eissa, Razak et al. (2013) in that the it is design to enhance routing efficiency in Emerging Systems by identifying and removing malicious or misbehaving system components, but from a consideration of Quality of Service (QoS) perspective. The approach is decentralised, non-cooperative and mathematically simple in its network layer implementation.

No form of collaboration is assumed in the Eissa, Razak et al. (2013) approach and none of the system holds privileged knowledge of the system therefore, the schema could be considered as a trivial non-cooperative game of full information.

The efficiency of the approach is explored experimentally concluding that the results are “promising” (Eissa, Razak et al. 2013), with further experimentation proposed to explore the suitability of the schema in a wider area range and with a higher volume of mobile devices. This emphasises the importance of testing these systems under circumstances that exaggerate their possible constitutional characteristics (volume and distribution in this case) to ensure that the approach remains viable.

Weimerskirch and Thonet (2002) make use if the “Distributed Trust Model” as a decentralised approach to trust management which uses a recommendation protocol to exchange trust-related information: “The model assumes that trust relationships are unidirectional and taking place between two entities. The entities make judgements about the quality of the recommendation based on their policies, i.e. they have values for the trust relationships. Also trust is not absolute, e.g. [*sic*] an entity can change the trust value is received as a recommendation.” (Weimerskirch and Thonet 2002).

The recommendation protocol works by requesting a trust value in a trust target with respect to a particular classification. After getting an answer, an evaluation function is used to obtain an overall trust value in the target. The protocol also allows recommendation refreshing and revocation. To do so the recommender sends the same recommendation with another recommendation value, or a neutral value to revoke (Weimerskirch and Thonet 2002).

Weimerskirch and Thonet (2002) extend the model by adding a request for references as a way to validate identity and authenticate.

The model is consistent with the Emerging System paradigm and scalable through the use of recommendations and references to derive trust values, and has a low operational demand on physical nodes to implement. "Using our model, the nodes in an ad-hoc network can set up a secure [communication] channel." (Weimerskirch and Thonet 2002). While there are some potential concerns such that entities or nodes need to reveal their transactional relationships to each other and the need for a centralised feedback mechanism, the model incorporates all the essential qualities of a trust framework needed to support Emerging Systems. This work makes use of all of these attributes - scale, agility, decentralisation and efficiency - in experimentally testing a mathematically rigorously defined trust framework. It is not apparent how Weimerskirch and Thonet (2002) derive a consensus of trust opinion which is the focus of this work, but the application and concerns are shared.

Nodes within an Emerging System establish relationships with other nodes directly connected to them. From theses nodes, this work demonstrates how trust can be established from various nodes that all share a relationship with a particular node through a consensus of trust - modelled game theoretically. A node will determine a trust value from the reputation of an adjacent node. Gowthami and Buvaneswari (2013) propose a clustered-node approach as a way to reduce the computation overhead of the trust and reputation determination process. Neighbouring nodes establish an internal reputation for the cluster within which they exist, based on some threshold. New relationships can be established at a less granular, cluster-based level reducing the need for individual nodes to determine trust relationships with large numbers of individual nodes and reducing the computational overhead of determining the trust values.

This approach potentially lends itself to a cooperative game theoretic analysis to the establishment of the clusters. This is not the intent. Each node remains self-interested as does any cluster of which a node may be a member. The criteria underpinning membership is determined dynamically, without a central authority (in accordance with our definition of the characteristics of an Emerging System) and published to external nodes. The game remains non-cooperative and of complete information.

The selection of a node for a potential trust relationship is made more efficient when there are fewer to consider directly. Nodes are members of clusters by virtue of the trust attributes that they share. Should a node exhibit trustworthy credentials determined by some threshold, it may be included in a cluster of similarly qualified nodes. Moreover, should a cluster deem a node

untrustworthy, the offending node will be removed from the cluster. Each cluster becomes self-regulating and is able to assure and declare its trustworthiness to other clusters.

Experimentally, we can consider the representation of a cluster of nodes to be the same as a single node. The criteria for membership to a cluster is the consensus trust value of every node within in the cluster and can be present within the system as a single node (potentially with some variance threshold). While there is no significant practical experimental difference between the consideration of single nodes and clusters of nodes in this work, though the potential for increased efficiency by reducing the volume of computation required to establish trust values and how well suited the approach is to emerging networks in that it could provide more stability for highly dynamic topologies - a trusted node could be a pool of similar nodes, taking turns to represent the cluster as the “head” (Gowthami and Buvaneswari 2013) node - is worthy of note here.

As for the efficiency of the approach, Gowthami and Buvaneswari (2013) concede, "[i]n this system the [trust] messages passed through the cluster head may overload, creating a bottleneck due to the additional message exchanges." (Gowthami and Buvaneswari 2013). Future effort is proposed to address this concern by increasing the efficiency and granularity of the approach, which is primarily applicable to cryptographic security.

Buchegger and Le Boudec (2002) propose “Confidant” (Cooperation Of Nodes, Fairness in Dynamic Ad Hoc Networks) protocol for the detecting and isolating malicious nodes and guiding efficient information flow through a system. Confidant consists of four components which monitor, maintain, report and manage the state of the system:

Component	**Role**
Monitor	Monitors neighbouring nodes for signs of malicious behaviour – packet dropping for instance.
Reputation System	Records notifications from the Monitor component and collates a reputation for each node. Nodes behaving outside of thresholds of behaviour, are added to a “blacklist” and deemed malicious.
Trust Manager	Broadcasts alerts to the system that identify malicious nodes.
Path Manager	Ranks and maintains routing information based on the reputation of nodes on a path through the system.

Table 2 Cooperation of Nodes, Fairness in Dynamic Ad hoc Networks

This approach is predicated on a centralised authority which is in contradiction to the characteristics of an Emerging System as defined here. In this work, monitoring, reputation and trust is managed in a distributed fashion, by individual nodes in the system to corroborate Brewer's (2000) theorem. There is a level of cooperation between nodes that is unreasonable to assume with highly reprogrammable nodes in a system and is potentially open to abuse that would not be detected outside of a closed system (Shah, Jan et al. 2012). The approach does however, serve to highlight more traditional monitoring-based approaches to managing ad hoc networks and how this approach is no longer appropriate for Emerging Systems.

Wang, Wu et al. (2008) carry out further experiments with the Confidant system, extending it to consider more closely the possibility that nodes provide spurious information and how this can be managed. They propose a Trust Scaling Factor (TSF) to reduce the impact of potentially misleading information shared between nodes. Integral to the trust framework proposed in this work is the configurability of the environmental and individual criteria for trust assessment. The reputation of a node can determine how much influence its opinion is considered in the consensus opinion of another node. The experimental cases explore how the framework performs; if consensus is reached and how quickly.

With a monitoring approach similar to Buchegger and Le Boudec's (2002), Marti, Giuli et al. (2000) propose a "watchdog" implemented on every node in the system which is able to identify misbehaving nodes by comparing data communicated across the system with that contained in a buffer. Discrepencies between beyond a threshold deem the node untrustworthy, and a "path rater" updates the ranking of routing paths through the system accordingly to identify the most reliable paths. Because the watchdog and rater elements of the approach are deployed on every node, the approach complements the distributed characteristics of an Emerging System well. There is an overhead to carrying out large amounts of computational work on indiviual nodes the efficiency of which is in principle experimentally considered for the trust framework explored in this work (Eissa, Razak et al. 2013). The focus for Marti, Giuli et al. (2000) is the reliability of a path through a system not trustworthiness though it is trivial enough to observer the analogy. For highly reprogrammable nodes in a system, there is potential for the elements to be modified. While not directly address in this work, this is a concern. Traditional network elements tend to have more fixed, firm and hard (read-only) implementations of their routing behaviour that are less likely to be abused because they are far less easily modified (Schuett, Butts et al. 2014).

Composed of three components – generation, distribution and evaluation – used to determine trust evidence in a system, Eschenauer, Gligor et al. (2004)'s framework considers trust as a set of

relationships established according to evidence. Any node can generate (and revoke with counter evidence) trust evidence about any other node. Evidence may be an identity, a public key, a location, an independent security assessment, or any other information required by the policy and the evaluation metric used to establish trust. Evidence can be replicated across various nodes to guarantee availability. The distributed nature of the approach – that evidence is persisted in the system beyond the presence of the node that generated the evidence in the fist instance – accords well with Emerging System characteristics. Experimentally, this work examines the behaviour of the framework under topologically volatile circumstances such as this, particularly when nodes are intermittently abscent from the system. It is assumed that the last opinion provided by a node before it failed to respond is enduring until it re-establishes itself within the system, or it is expelled permanently for breaching tolerance thresholds.

Eschenauer, Gligor et al. (2004)'s approach is designed to be implemented at what OSI would consider a level 2, Datalink layer, while this work is concerned with supporting application layers but they share conceptual approaches, not least because there is no assumption of cooperation between nodes. The approach does however, implicitly rely on a central authentication authority to assure each node in the first instance. It is not clear quite how this would be carried out in a pure distributed fashion as necessary to support an Emerging System.

Building the trust relationship between entities is a fundamental problem in ad hoc networks, since the availability of servers, which distribute trust certificates, is not guaranteed. Furthermore, the existence of any trusted server might not be assumed either (Jiang and Baras 2004).

Sen, Chowdhury et al. (2007) and Sen (2010) propose a scheme for the establishment of trust and a framework based on it. The approach is in keeping with most discussed here in that each node in the systems maintains a record of its interactions with other nodes in the system to establish a reputation against which malice and referential paths can be identified and shared. Of particular interest here are the scenarios under which reputation is established and how it is maintained. (Sen 2010) elaborate on three scenarios:

1. Reputation computing during system establishment;
2. Combining previous and current reputation values, and;
3. Establishing reputation when exchanging reputation information within a neighbourhood.

A reputation is an integer value inclusively between zero and one, computed with consideration for the ratio of correctly forwarded data packets and those sent over time. A newly established relationship between two nodes is attributed a value of one. Should the perceived trustworthiness

of a node change, the maintained reputation value undergoes a correction derived from a sum combination of the previous and current values, and a weighting factor used to dampen volatile fluctuations in opinion to prevent the system's path topology from becoming chaotic.

The last scenario concerned with exchanging reputation information within a neighbourhood, can either be proactive or "as required". In the proactive case, a node broadcasts its reputation evaluation to its neighbours. A reputation value of zero is indicative of malice or failure, and could be broadcast as a warning to other nodes in the system to avoid incorporating the node in routing paths. In the second case, the reputation evaluation of node is actively requested by another node.

The credibility of the reputation evaluation provided by each node in both cases, is also weighted according to the internally held view of each of the contributing nodes. The less trusted a node is from whom a request for opinion has been made, the lower the importance attributed to its opinion of other nodes.

This work is a generalised case of Sen's (2010) implementation. The underpinning mathematical framework extends trust evaluation to higher dimensional spaces with a greater modelling complexity and flexibility to the relationships between nodes – game theoretically. The experimental analysis in this work makes use of Sen's (2010) weighting factor approach but extends it to higher dimensional environmental factors as part of a more abstract framework with more varied application beyond ad hoc network packet routing, and particularly at the application layer (OSI) on highly reprogrammable nodes. Hostile nodes consisting of malicious opportunistic code has not be specifically considered in this work.

Xia, Jia et al. (2011) propose a subjective trust management model called "AFStrust" which utilises multiple decision factors based on the Analytical Hierarchy Process (AHP) (Saaty 1980) (Saaty 1988) theory and fuzzy logic (Klir and Yuan 1995) rules prediction method, in a bid to reduce the hazards from malicious nodes within a system.

When the factors of decision-making are given, though we know that different factors have different weights, the precise weights are difficult to determine. Existing methods in these models for weight determination lack rationality and practicability. As a result, they cannot calculate an accurate trust value for each node. Hence, these models are ineffective in MANET trust management and their applications are very simple (Xia, Jia et al. 2011).

Xia, Jia et al. (2011) recognise the need for advanced tools and techniques to determine trust and reputation accurately, beyond what has been previously proposed. Being derived from Game Theory means there is a richness to the mathematical approach in this work capable of supporting

the characteristics of Emerging Systems under complex conditions (Alpcan, Rencik et al. 2010). With complexity however, comes computational overhead, the impact of which this work explores experimentally. Xia, Jia et al. (2011) explore the effectiveness of their approach experimentally (concluding that is it highly effective as a result of the capabilities of the mathematical approach) but do not directly address how practical deployment of the approach into an Emerging System where resource contrained nodes constitue the majority of the members, would be.

This work proposes a trust framework based on Game Theory that is able to find consensus of opinion within an Emerging System. Consensus is reached by establishing a mathematical equilibrium within the system, modelled as a "game". The experimental analysis carried out here relies on computational capability of the underlying simulation environment (Matlab) and assumes that the parameters for establishing trust are the same between nodes because they originate from the same system. From our definition of an Emerging System, this is a precarious assumption. Huth and Kuo (2014) (Huth and Kuo 2014) propose a tool, "PEALT" based on Pluggable Evidence Aggregation Language (PEAL) and the Z3 constraint solver (Bjørner and Moura 2014), for the understanding and validation of mechanisms that numerically aggregate trust evidence from heterogeneous sources. The tool determines the compatibility of trust constraints between domains. While this work does not explicitly define trust parameters, more proposes how consensus can be establish in abstraction once the trust domain is established, PEALT (Huth and Kuo 2014) could serve to distribute the framework's implementation across variable domains. It would be possible to establish the suitability of trust evaluation amongst one faction of an Emerging System with another to establish a consensus of trust between domains based on common mapped criteria. Used in conjunction with Gowthami and Buvaneswari's (2013) clustering approach, PEALT could be used to distribute trust consensus rapidly within an Emerging System and conjoin disparate heterogeneous systems. A major consideration would be how to implement the approach without a centralised authority, where the computational effort would reside and how the implementation approach could be abused.

Gong, You et al. (2010) take a different approach to increasing the performance of an Emerging System, by incentivising non-malicious behaviour of nodes rather than trying to isolate malicious ones. Nodes consider the behaviour of neighbouring nodes and their performance from previous interactions to establish a vector of trust model (Ray and Chakraborty 2004). This work attempts to establish a consensus of trust be it positive or negative.

Venkataraman, Pushpalatha et al. (2012) propose a regression-based, proactive and reactive trust model Vector Auto Regression (VAR) over Ad-hoc On-demand Distance Vector (AODV) and

Optimised Link Static Routing (OLSR) MANET protocols to identify malicious nodes and improve performance across the system. The proposal is significant to this work as it emphasises the importance of establishing suitable trust criteria and thresholds.

From experiments carried out in a test-bed, "[t]he performance evaluation shows that by carefully setting the trust parameters, substantial benefit in terms of throughput can be obtained with minimal overhead." (Venkataraman, M et al. 2012).

Trust metrics are established for all contiguous nodes in the system and stored locally as trust vectors. This work shares this approach and generalises it as a framework for implementation in highly reprogrammable nodes. The specifics of the trust metrics are left to the implementation as is the interpretation of the suitability of imposed thresholds. Venkataraman, Pushpalatha et al. (2012) define a suitable trust metric specifically for MANET's and explore how effective they are for detecting malice in a system.

Venkataraman, Pushpalatha et al. (2012) approach incorporates a confidence vector that weights the perceived significance of an individual node's opinion on that held by another node. In this work, influencing factors are modelled through environmental variables that can be used to reflect system-wide concerns or individually to reflect the bias of a particular node. These highly configurable parameters mean that an approach such as (Venkataraman, Pushpalatha et al. 2012)'s could be modelled within the framework.

Applied specifically to battlefield group military mission communication and based on a stochastic Petri nets (Haas 2002), Cho, Swami et al. (2012) attempt to identify the optimal length of a trust chain among peers in a "trust web", that generates the most accurate trust levels. The approach attempts to do this without increasing risk of compromise to the system (identification of nodes for instance - Gunasekaran and Premalatha (2013) consider an approach to anonymity while reporting malicious nodes), based on a trade-off between trust availability and path reliability. Cho, Swami et al. (2012) define a trust metric for mission-driven group communication systems in MANET's to accurately reflect unique characteristics of trust concepts and demonstrate that an optimal trust chain length exists for generating the most accurate trust levels for trust-based collaboration among nodes in MANET's while meeting trust availability and path reliability requirements. They consider "social trust" (Golbeck 2006) and "Quality of Service (QoS) trust" as their trust domains. This is an example of the application of this work where trust criteria can be modelled for a specific situation. There are specific security and risk factors that affect the application of the framework in this case. Through the establishment of suitable trust criteria and thresholds, and fitting environmental factors, the framework can reflect the implementation accurately.

In this work, we take a graph theoretical approach to modelling the systems of nodes while Cho, Swami et al. (2012) chose Petri nets. These are similar concepts in that a Petri net is a directed bipartite graph in Graph Theory. Stocastic random transitional variables in Peti nets model the topological volatility of Emerging Systems – where nodes join and leave the system regularly and unpredicatably. A random variable is used to introduce a "delay" into the "transitions" between nodes in the Petri net (Marsan 1990). In this work, the behaviour of nodes in the system is modelled experimentally in adherence to a statistical distribution or random generation. The algorithm used to test the framework is parameterised so that different behaviour can be modelled, from a completely stable system where every node response to every request (potentially an arbitrary static case) to extremely high volatility (more consistent with Emerging Systems).

The security issues for ad-hoc networks are different than the ones for fixed networks. While the security requirements are common, namely availability confidentiality, integrity, authentication, and non-repudiation, they are considered differently for ad-hoc networks. This is due to system constraints in mobile devices and frequent topology changes in the network. System constraints include low-power microprocessor, small memory and bandwidth, and limited battery power. (Weimerskirch and Thonet 2002).

Menaka and Ranganathan (2013) summaries the results of their survey of trust-related protocols for mobile ad hoc networks as follows:

Approach	**Routing Technique**	**Methodology**	**Performance Metric**
Context aware inference	Dynamic Source Routing (DSR) - similar to AODV in that it forms a route on-demand when a transmitting computer requests one. However, it allows the sender of a packet to partially or completely specify the route the packet takes through the network instead of relying on the routing table at each intermediate network element.	Add digital signatures to route request ordered lists of nodes that a request has traversed. Routes will not be included where their signatures cannot be verified.	Throughput and data delivery success.
Trust scheme	Dynamic Source Routing (DSR).	Misbehaving nodes are detected and isolated.	Throughput.

Misbehaviour detection	On-demand routing.	Redundancy of routing information is used to identify misbehaving nodes.	False positives, successful detection rate, total convergence time (the time taken to distribute trust certificates to all non-malicious nodes in the system) and communication overhead.
AFStrust	On-demand routing	Analytic hierarchy process and rules to weigh each node.	Dynamic adaptability, network interaction quality and identification of malicious nodes.
Trust-based AODV	AODV	Trust values are used to identify misbehaving nodes.	Throughput and delay.
Auto-regression trust model	AODV and OLSR	Trust parameters are set up to minimise the overhead of finding malicious nodes.	Data losses and end-to-end packet delivery.
Trust enhancement	On-demand routing	Misbehaving nodes are identified based on multiple claims by neighbouring nodes.	Anonymity.

Table 3 Survey of trust-related protocols for mobile ad hoc networks

This work does not concentrate on using trust to increase security in a system, more to define an abstracted framework that can be tested in principle, against the requirements for supporting Emerging Systems. It is in essence a "trust enhancement" approach better tested for integrity under volume and topological volatility.

2.4.3 Summary

Menaka and Ranganathan (2013) demand that the unique characteristics of Emerging Systems require careful consideration in matters of trust. The main trust features shared with Emerging Systems are amalgamated as (Golbeck 2006) (Cho and Swami 2009) (Adams and Davis IV 2005):

1. Attribution of a reputation to an entity must be distributed because the existence of a central, trusted authority cannot be assumed;
2. Trust must be established in a highly flexible fashion that captures the complexities of trust relationships between entities.

3. Consideration must be made of the computation and communication overhead of establishing trust relationships;
4. Trust frameworks should not assume that all nodes are cooperative;
5. Trust is dynamic (not static). Reputation changes over time - diachronically;
6. Trust is subjective and based on or influenced by individual entity environmental factors, constraints and opinions;
7. Trust is not transitive. If X trusts Y, and Y trusts Z, it does not mean that X trusts Z;
8. Trust is asymmetrical and cannot be assumed to be reciprocal;
9. Trust is contextual. The circumstances and domain of trust should be defined.

The trust framework proposed in this work demonstrably exhibits these features:

- the mathematical framework algebraically defines and constrains the association between nodes (asymmetry, non-transitivity and context);
- non-cooperative game theoretical tools describe relationships between nodes (complex, flexible and subjective), and;
- experimental testing establishes the suitability of the framework for application in Emerging Systems (distributed, topologically volatile, resource scarce and dynamic).

The next section considers game theory and its application to Emerging Systems, building on the previous two sections to establish the complete literature foundations to this work.

2.5 Game Theory and Emerging Systems

2.5.1 Introduction

The previous section explored trust, trust frameworks and a trust layer in Emerging Systems. It provided a formal definition of trust for use throughout this work and reviewed current literature on the applications of the concepts of trust to assist Emerging Systems with security and efficiency problems.

This section examines game theoretical approaches to Emerging Systems, through the literature. It considers selfishness and malice, and how game theory is applied to solve equilibrium problems. The suitability of game theory techniques that do not assume cooperation to Emerging Systems is established.

2.5.2 Discussion

Emerging Systems follow standards and protocols so that entities in the system are able to communicate effectively. Internet architecture for instance, follows the TCP/IP (Transmission Control Protocol / Internet Protocol) (Stevens and Wright 1995). Entities are assumed to follow the rules of the protocol in exact order (Shah, Jan et al. 2012). Full agreement to cooperate and follow the rules cannot however, be guaranteed. Entities in an Emerging System are manufactured by different vendors and configured by different administrators, are prone to malicious attack, and most importantly here, are highly reprogrammable by different programmers with inconsistent agendas. There can be no assumption that the assorted entities that constitute the Emerging System are following the protocol and not behaving selfishly in some way to serve their own purposes, potentially at the expense of others (Urpi, Bonuccelli et al. 2003). This selfish behaviour enables individual entities to maximum their own performance by unfairly consuming shared resources within a system.

The concept of "selfishness" (formally defined by Urpi, Bonuccelli et al. (2003)) is one directly considered in Game Theory. It is modelled using "non-cooperative" techniques that assume that there is no collaboration between entities in a game and that all entities are purely self-interested.

In fact, Urpi, Bonuccelli et al. (2003) use a Bayesian (Harsanyi 1967) game-based (where information is incomplete), cooperative general model where nodes in an Emerging System make decisions on how to route traffic based on historical experience of the behaviour of neighbouring nodes, trading off energy consumption and throughput. Modelling Emerging Systems as games is the theoretical basis of this work though the approach is to use Nash equilibrium solutions for non-cooperative games, argued to be better suited to Emerging Systems, and is applied to the application layer of highly reprogrammable nodes.

There are two approaches to managing selfish behaviour in Emerging Systems (Shah, Jan et al. 2012). The first is to establish an incentive and punishment mechanism to promote collaboration and discourage misbehaviour. This can take the form of a trust mechanism (Han and Liu 2008) (Alpcan and Başar 2005) that attempts to regulate behaviour through self-moderation by entities of other entities within their purview. The system only serves the members who cooperate. The second and complementary approach in this work, is non-cooperative Game Theory. Modelled as a game, non-cooperative game theory provides a rich set of the mathematical tools that suit the nature of Emerging Systems – members cannot be universally discerned as not acting in their own interests. By assuming the behaviour of all entities is selfish, non-cooperative theory can be used to

solve strategic problems in Emerging Systems to ensure no entity can unfairly take advantage of the systems resources or service relationships between entities.

As well as trust frameworks, the approach is most commonly applied to the access mechanisms to shared resource (Chen, Low et al. 2010), the OSI Model Media Layers of the stack (Sun, Ding et al. 2014), and then to network routing (Altman, Boulogne et al. 2006).

Chen, Low et al. (2010) propose a game-theoretic model, dubbed a "random access game", for resource contention control. They characterise Nash equilibria of random access games and propose distributed algorithms to achieve Nash equilibria. This provides a general analytical framework that is capable of modelling a large class of system-wide Quality of Service (QoS) models through the specification of per-node cost functions, in which system-wide fairness or service differentiation can be achieved in a distributed manner as long as each node executes a contention resolution algorithm that is designed to achieve the Nash equilibrium. In this work, nodes with an Emerging System resolve trust value games represented by systems of cost functions, the solutions to which, provide a Nash equilibrium. However it is applied, the trust framework serves to abstract (applicable to any resource), formulise and resolve some form of contention situation. The experimental analysis uses a computational algorithm to find the solutions on each node.

In their survey on networking games in telecommunications, Altman, Boulogne et al. (2006) consider primarily non-cooperative games. They discuss different equilibrium concepts, in terms both of their qualitative and quantitative properties. In particular, they consider in depth the issue of uniqueness of an equilibrium, the Braess (Braess 1965) paradox (Korilis, Lazar et al. 1999), controlling equilibria through design parameters or pricing, as well as the Stackelberg (Von Stackelberg 1932) (Von Stackelberg 1934) (Cui, Zhou et al. 2014) framework for hierarchical, or leader–follower, equilibrium. They also provide a brief summary of some work on equilibria in cooperative games that are related to resource allocation, pricing and to the Stackelberg framework. The uniqueness of Nash equilibria assures that the solution to an Emerging System game is the only one. That is, the identification of a solution assures that the nodes in the Emerging System are truly in a strategic position where none of them can gain advantage by deviating from it. By limiting the cost functions each node can use to represent itself to quadratic functions, we assure that the Nash equilibria that are identified are unique. There is no reason why the mathematical framework in this work could not be used to underpin Bayesian, Stackleberg (which uses Nash equilibrium as a solution), Stochastic or any other game theoretic technique for finding strategic solutions. It can easily be extended to consider the avoidance of capacity inhibitions (such as adding extra capacity to a network when the moving entities selfishly choose a route that compromises overall system performance since the Nash

equilibrium solution of the system is not optimal) or applied to cooperative games. Non-cooperative games have been initially considered based on how well they suit the characteristics of Emerging Systems, particularly, the highly reprogrammable nature of the constituent nodes.

Non-cooperative game theoretical models are often used to solve path-finding problems to ensure that the most efficient route is determined for transferring data from one entity to another within the system (Han, Niyato et al. 2012). A similar approach is used to determine how resources should be allocated within system. At the OSI Transport Layer, TCP for instance has been analysed using game theory to reduce congestion over the network (Alpcan and Başar 2005) (Başar and Bernhard 2008). With the goal of maximising the spectrum utilisation between primary and secondary users, cognitive radio networks have been modelled game theoretically (Niyato and Hossain 2008) (Nie and Comaniciu 2005) (Wang, Wu et al. 2010). Shah, Jan et al. (2010) Shah, Jan et al. (2011) adopt game theory to assign Medium Access Control (MAC) (Demirkol, Ersoy et al. 2006) layer channels (He, Ma et al. 2013).

Significant throughout this work and in their general contribution to the field figuring often in contemporary literature, are Alpcan and Başar. In Alpcan and Başar (2005), they develop a congestion control scheme in a non-cooperative game framework, where each node's cost function is composed of a pricing function proportional to the queueing delay experienced by the user, and a cost function which captures the user demand for bandwidth. Using a network model based on fluid approximations and through a realistic modeling of queues, they establish the existence of a unique equilibrium as well as its global asymptotic stability for a general network topology, where boundary effects are also taken into account. They provide sufficient conditions for system stability when there is a bottleneck link shared by multiple users experiencing nonnegligible communication delays. While applied specifically to congestion control, this work shares its approach with Alpcan and Başar (2005) in establishing the existence and stability of equilibrium of cost functions. The cost functions in this work are used to model concensus of trust between nodes but the approach is very much the same. Taken into account in this work is the volatility of Emerging Systems' topology with the framework experimentally tested for stability. Futher, boundary effects are considered with the inclusion of environmental factors that influence and bias the cost functions of the system.

Cooperative Game Theory applied to Emerging Systems, on the other hand, where entities within a system are encouraged to collaborate directly has also been adopted at the OSI Physical Layer by Han and Liu (2008) Alpcan and Başar (2005), at the network layer by Alpcan and Başar (2005) while congestion control modelled as a cooperative game has been posited by Floyd and Fall (1999), and Kelly (2003).

Comparatively, designing cooperative games in a large system like the Internet and other scalable networks faces many challenges ranging from efficiency, complexity and fairness amongst the individual users (Shah, Jan et al. 2012).

The effective role of cooperation in Emerging Systems was introduced as relay channel cooperative games by Başar and Olsder (1999). Cooperative strategies adopted by network elements have been proposed by Alpcan, Başar et al. (2001) MacKenzie and DaSilva (2006).

Altman, Boulogne et al. (2006) prove that multi-hop forwarding achieves optimal capacity scaling in systems of large populations. In these large systems, cooperation has also been proved to improve energy efficiency in MANET's (Thrall and Lucas 1963) (Başar and Olsder 1999). These approaches all assume cooperation between entities within the system, that entities perform selfless acts to their own detriment to service the needs of other entities in the system. Further, they assume the existence of a central authority that organises the entities and mediates their behaviour. It is clear that these are not reasonable assumptions for Emerging Systems and contradict the definition. Cooperative theory then, is not suitable as a mathematical underpinning for the framework proposed in this work. As with MANET's (Basagni, Conti et al. 2004) and Wireless Mesh Networks (WMN) (Akyildiz, Wang et al. 2005), Emerging Systems are highly distributed, decentralised and automatically configure themselves.

Mejia, Peña et al. (2011) claim that cooperation between nodes within an Emerging System is fundamental to its operation. They propose a mechanism for enforcing the cooperation between nodes in the system based on non-cooperative game theory that encourages cooperative behaviour through a learning "bacterial" algorithm. Mejia, Peña et al. (2011) recognise the need for an effective adaptable distributed model optimised to support the high volatility of Emerging Systems.

Baras and Jiang (2004), and Baras and Jiang (2005) attempt to establish, propagate and manage trust within an Emerging System with cooperative game theory. Significant to this work, is their approach to the propagation of trust using local interactions that proliferate to full distribution. They prove that trust mechanisms can establish collaboration, even without negotiations between the nodes. This work concentrates on the establishment of trust consensus between local nodes. Baras and Jiang (2005) demonstrate how this is extended to systems of potentially boundless size which is crucial to support Emerging Systems.

An Emerging System is the underpinning structure of social networks. Etesami and Basar (2014) consider game theoretical "diffusion games" that aim to strategically solve the "best placement" problem of how to "seed" a social network most efficiently. A seed is an initial node within the

system that distributes information to its adjacent nodes. Different seeds are assessed for capacity for efficient proliferation of the information, modelled by a game theoretical game. The equilibrium of the systems is measured in terms of the "social welfare" of each seeded node.

2.5.3 Summary

This section explored the application of game theory to Emerging Systems and identified the common types of problems it is used to solve.

From the literature, while much has been considered to address trust and control problems in physical system entities and protocols, little has been done to approach the highly re-programmable entities typical in Emerging Systems. This work specifically addresses how a layer beneath the application layer can be used to mediate trust within Emerging Systems using a non-cooperative game theoretical trust framework.

> *"In such environment [sic] [as Emerging Systems], the assumption of cooperation may not be valid. The increased capability of reprogrammability [sic] of wireless devices offers another threat to this assumption. It is therefore, important that the issues in networks like WMNs and MANET's should be addressed by using the concepts from non-cooperative game theory." (Shah, Jan et al. 2012).*

2.6 Conclusion

From the literature, we have been able to identify a research gap that exists for:

1. **the definition of *Emerging Systems* for the contemporary enterprise, and;**
2. **a Trust Framework that:**
    a. **is suitable for Emerging Systems**
    b. **can be implementable to support the application layer, and;**
    c. **is specifically, non-cooperative.**

In this chapter, we have established the definitions within Game Theory and Trust to be used throughout this work.

- *Game Theory* is a mathematical modelling framework for rational decision making behaviour.

- *Trust* is a relationship between two neighbouring entities where a trust value expresses the degree that one entity expects another entity to offer certain services. The reputation of an entity, is the record of the trust values attributed to an entity by other entities.

Most significantly, we have identified the need for and the definition of *Emerging Systems*. Further, we have established that Emerging Systems are imbued with the characteristics:

- **Decentralisation;**
- **High distribution;**
- **Self-configuration;**
- **Self-regulation;**
- **Non-cooperation;**
- **Pervasiveness;**
- **Dynamic open ad-hoc ("for this" purpose) topology – non-generalisable;**
- **No fixed infrastructure;**
- **Wireless connectivity;**
- **High scalability, and;**
- **Consisting of highly reprogrammable nodes.**

We have established that the overwhelming research that has been carried out in trust frameworks has been at the physical and protocol layers of the stack. Virtually nothing has been done at the application layer. Similarly, the application of game theory to trust or resource allocation or routing problems is at these layers.

A differentiating characteristic of Emerging Systems from other types of systems (MANET's for instance) is the reprogrammability of the constituent nodes. This characteristic makes them prone to selfish and potentially, malicious behaviour. A framework for establishing consensus trust sufficiently abstracted from the specific purpose of any application would serve to address this, and could be implemented as a "trust layer".

The literature supports the claim that a non-cooperative approach is best suited to modelling Emerging Systems. The potential selfishness of the nodes in the system means that no assumption of collaboration can be made and by definition, there is no central authority in an Emerging System to mediate and assure the credibility of nodes. While other game models exist, stable and unique Nash equilibrium provide a suitable solution to Emerging Systems modelled as a game.

It will be explored experimentally that the trust framework proposed in this work is able to support the scale, high distribution and topological volatility characteristics of Emerging Systems. Different

extreme circumstances will be tested to determine their effect on the ability of the framework to exhibit equilibrium where the nodes in the system manage to reach a consensus of trust and reputation.

2.6.1 Experimental Analysis

Michalopoulou and Mahonen (2012) suggest that it is clear why there is a trend towards designing distributed and self-organizing wireless networks being analysed with game theoretical models, "…usually the nodes of the wireless network are considered as the players of the game that have to take their own decisions in order to optimize their performance."

However, Michalopoulou and Mahonen (2012) question how desirable a Nash Equilibrium solution is for wireless networking problems. They posit that the highly variable topological nature of wireless networks can cause instability around Nash Equilibria. Their approach is to establish the connection between Game Theory and *Statistical Physics*, specifically *Statistical Mechanics*, and explore the equilibrium problems in wireless networks through an analogous game theoretic framework.

Statistical Mechanics is a branch of physics concerned with the macroscopic properties of large population systems (McQuarrie 2000) (Huang 1963). As game theory models the interactions among players, statistical mechanics studies the interactions among molecules, atoms, or particles in a physical system (Michalopoulou and Mahonen 2012).

Michalopoulou and Mahonen (2012) caution that while not an inherent problem with game theory, Nash Equilibrium as always a desirable outcome for game theoretic frameworks, can be unreasonably taken for granted as a suitable stability solution for wireless networks.

External factors play a significant role in influencing the final stability solution. In a fixed game without external influences, a Nash Equilibrium can be well established. This is not the case in highly dynamic situations such as Emerging Systems where, as the stability solution is in the process of being established, the factors that determine it are changing. The system can "drift" (Michalopoulou and Mahonen 2012) in proximity to the Nash Equilibrium solution. In statistical mechanics, even a small drift from equilibrium can cause the system to undergo a *phase transition*. In game theoretical terms, this would be a significant change to the outcome of a game. This fluctuation could have a large degrading effect on the performance of the system.

Michalopoulou and Mahonen (2012) conclude that a game theoretical analysis of systems that are characterised by persistent environmental change, aiming to provide an understanding of a particular mechanism, needs to examine the behaviour of the game around the equilibrium solution. They support this conclusion experimentally in the case of a simplified resource allocation game

using an Isling ferromagnetism model (McDonald 1985) (Srivastava and Ashok 2005) for CSMA/CA Medium Access Control (MAC) (Demirkol, Ersoy et al. 2006) wireless protocol. The experimental results indicate that for critical numbers of users (nodes), stability can be lost around equilibrium solutions in the system.

The Michalopoulou and Mahonen (2012) findings while not specific to an application layer framework as proposed here, call attention to the importance of an experimental analysis of the trust framework. It is vital to establish the theoretical mathematical concepts underpinning this framework, but it is not enough to assume that these results can be extended to practice and it should be tested, particularly when scaled.

The application of statistical physics to wireless networks lacks the richness of mathematical models and tools available to a game theoretical approach. The exploration and extension of what is currently available is the future direction of the Michalopoulou and Mahonen (2012) enquiry.

3 A Mathematical Trust Framework for Emerging Systems

3.1 Introduction

The trust framework in this work is predicated on a mathematical structure that requires formal definition and rigorous investigation to establish its validity and suitability as the basis for the framework.

This section collates classical results primarily from five sources: J. A. Bondy and Murty (2008), Strang (2003), Bertsekas and Tsitsiklis (1997) and Bertsekas, Nedić et al. (2003), and Başar and Olsder (1999), and adapts many of the approaches taken by Alpcan, Rencik et al. (2010), to establish a mathematical trust framework for Emerging Systems. It draws on established fields of mathematics including: Graph Theory, Game Theory, Linear Algebra, Convex Analysis and Optimisation, Numerical Analysis, and Matrix Analysis.

"Every discourse, even a poetic or oracular sentence, carries with it a system of rules for producing analogous things and thus an outline of methodology" (Derrida 1986).

3.1.1 Contribution and Significance

While many of the results are unremarkable as they are well established, the contribution and significance of this chapter lies in the application of the results to the construction of the mathematical underpinnings integral to the trust framework.

The contribution and significance of this chapter is to support:

1. **demonstration that the formulation of mathematical constructs can define a trust nomenclature as a foundation for a trust framework;**
2. **proof of the suitability of rigorous applications of non-cooperative game theoretical techniques to establish stability and equilibrium applied to the constructs;**
3. **proof of the suitability of iterative methods and algorithms as the computational mechanics of these techniques for a trust framework, and;**
4. **derivation of well-constructed cost function as a candidate for the experimental analysis of the trust framework.**

The statements of these are indicated in the text by a mathematical tomb-stone or *quod erat demonstrandum* notation, "□".

3.1.2 Conceptual Model

The mathematical structure can be conceptually modelled:

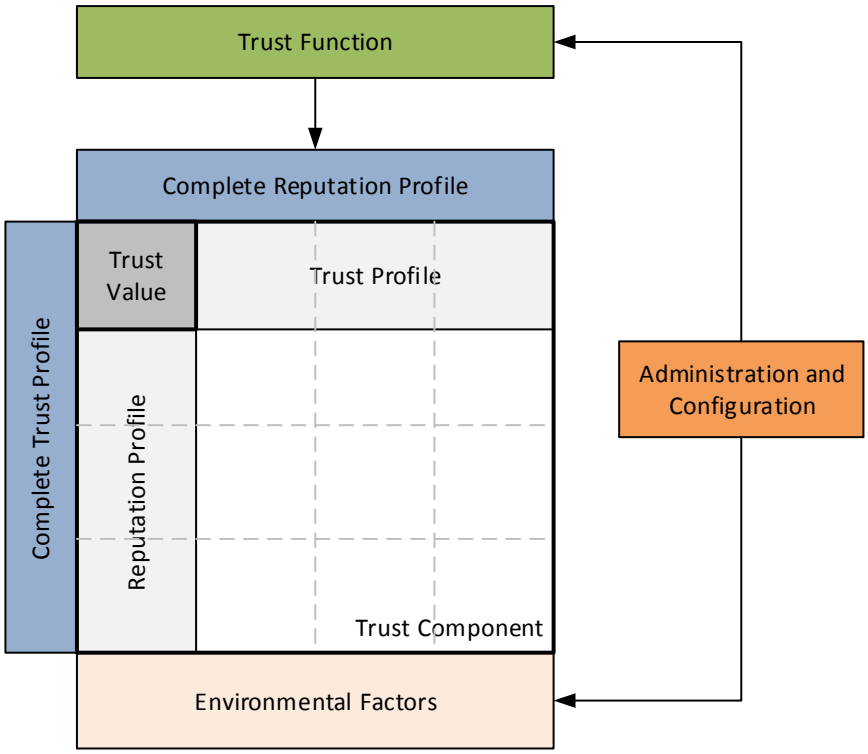


Figure 3 Trust framework conceptual model

Each one of the components of the model is explored and rigorously examined to establish it as fundamental to the framework.

3.1.3 Roadmap

This chapter covers eight main topic areas:

- Graphs of Emerging Systems;
- Final Reputation Profiles;
- Multi-Component Trust Spaces;
- Environmental Factors;
- Convex Functions;
- Game Theory;
- Iterative Computation for Trust Spaces, and;
- A Game Theoretical Trust Framework for Emerging Systems.

3.1.3.1 Graphs of Emerging Systems

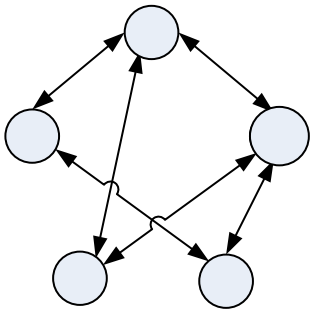


Figure 4 A simple graph node System representation

Emerging Systems can be formally described in this fashion and analysed through matrix representation of the node associations, with each vertex representing a member node of the System topology, and directed arc weights representing opinions held between nodes. This approach to representing Systems is continued throughout the section for illustrative purposes.

The mathematical underpinnings of the framework are established through the fundamental concepts:

- Trust Space;
- Reputation Profile;
- Trust Profile;
- Trust Value, and;
- Environmental Factors.

These concepts define the complete mathematical model for the trust framework and establish a nomenclature against which all analysis and computation is carried out. They are represented formally by linear algebraic constructs.

3.1.3.2 Final Reputation Profiles

Reputation and *trust profiles* consist of *Trust Value* components. A set of *Trust Values* constitutes a *profile*.

To establish Trust Values, nodes derive initial values based on the frameworks configured criteria, then determine final Trust Values after consultation with other nodes in the System, through the solution of a *Trust Function*.

Considering Trust Values in relation to other nodes in the System, forms a consensus that is of the local System and manages potential outliers that may unreasonably skew opinion.

The set of all profiles for a configured trust criteria forms a *trust space*.

3.1.3.3 Multi-Component Trust Spaces

Trust spaces represent a single component of the trust criteria configured for the framework. Hence, multiple trust spaces are required to support multiple criteria.

Trust spaces typically consist of many components, each component representing a trust criteria valued by the System.

The complete set of all component trust spaces forms the *complete trust space*.

3.1.3.4 Environmental Factors

A System administrator might control *Environmental Factors* to alter the influence of the various facets of the Trust Function. This could be in response to events outside the System such as a security breach or inside such as a change to the nature of an application within the Emerging System.

Environmental Factors are not derived from the experiences of the nodes in the System or influenced by a consensus of nodes, they are dictated at a System level.

3.1.3.5 Convex Functions

Convex functions form a suitable class of function that exhibits desirable qualities for the formulation of Trust Functions.

In particular, they emit unique minimal solutions and they adhere to classic associativity and commutatively properties, while maintaining convexity.

3.1.3.6 Game Theory

Game Theory contributes a wealth of rich mathematical tools and approaches to the analysis of trust. *Trust games* (game theoretical games used to model trust) can be played out over time and under changing circumstances to determine different outcomes and states of agreement between participants.

Emerging Systems can be modelled in Game Theory and trust evaluation can be analysed based on some simulated behaviour, influences and initial states, within a trust game.

The trust game allows each node to re-evaluate and update its individual opinion of another node based on some criteria, determined by a Trust Function.

The solution to a trust game is established by the minimisation of a well-defined *cost function* (in game theoretical nomenclature, specialised to a Trust Function here). This solution is a *Nash equilibrium* and is the fundamental mathematical concept underpinning the trust framework.

The type of game that the framework should use is established according to game theoretical characteristics of *non-cooperative games*.

These are key results that ensure that we are able to determine final Trust Values for reputation and trust profiles, as long as our Trust Functions exhibit the necessary characteristics and we constrain the trust game type suitably.

3.1.3.7 Iterative Computation

We consider how we can compute Nash equilibrium for Trust Functions with suitable iterative methods, their algorithmic implementation, and the stability of these solutions.

The computation method for the framework must be able to support the distributed nature of an Emerging System, and under rapidly fluctuating conditions and topology. Emerging Systems are potentially very large too.

We also consider the main methods in terms of ease of computation, convergence rate and factors that assure convergence.

We assess the methods' suitability for use within the trust framework for Emerging Systems and explicitly model the framework undergoing iterative changes.

3.1.3.8 A Game Theoretical Trust Framework for Emerging Systems

We formulate a well-constructed Trust Function as a candidate for experimental analysis of the framework.

The Trust Function is defined by its component terms and Environmental Factors, and is a member of the main class of function identified through the theoretical mathematical discussion.

The Trust Function is interrogated rigorously through techniques in the previous analysis in the chapter to ensure it exhibits the characteristics necessary to converge to an equilibrium solution of the trust game.

3.2 Graphs of Emerging Systems

3.2.1 Introduction

In this section, we formally describe an Emerging System in terms of graph theoretical graphs with directed and weighted arcs to denote an *opinion* value. This is the basis for the mathematical concepts underpinning the framework. All future sections, appeal to these foundational concepts.

We consider *reputation* as the set of weights of directed edges towards a node and *opinion* as the set of weights of directed edges away from a node. A directed edge indicates that an opinion is held between two nodes, and by and of whom. We go on to represent these graphs as association System matrices.

The matrix form of these sets determines the *trust space* of a System. We formally define *trust space* as the set of zero-diagonal node System matrices. That is, the complete space of all reputations and opinion combinations.

We formally define *trust profile* and *Reputation Profile* as set of opinions held of other nodes and as record of the opinions of others of a node respectively, in the *trust space*.

3.2.2 Graphs

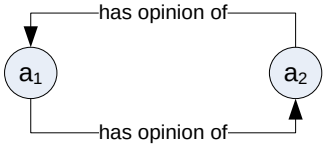
A node in an Emerging System is evaluated by the other nodes for its trustworthiness based on the *opinion* of the other nodes in the System based on some criteria it has derived from its experiences interacting with other nodes, and gains a *reputation* based on the local collective opinion. This is an aggregated trust amongst the System of nodes. It is extended to every node such that each node can hold a Trust Value for other nodes in the System.

Definition: We can consider the nodes in an Emerging System as:

$$A = \{a_1, \dots, a_i, \dots, a_N\}$$

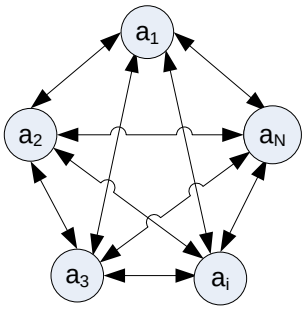
where A is the local System set of all nodes, N is the total number of nodes and i denotes the identity of a specific node.

In a simple System A of two nodes, a_1 and a_2, we have the graph, G:



where the connecting directed edges indicate which opinions exist, which node holds an opinion, and of which other node it is held.

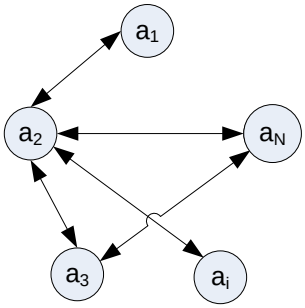
More generally, we have the graph, G:



In the case where $N = 5$ (alluded to here), there are $4 + 3 + 2 + 1 + 0 = 10$ mutual (bi-directional) opinion associations considering each node in order, generalising the sum of the arithmetic series is a result commonly attributed to Gauss (Cox 2004):

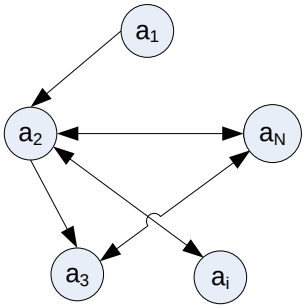
$$\frac{N(N-1)}{2} = \frac{5(5-1)}{2} = 10$$

In an ideal situation, every node holds an opinion of every other node. It is possible that not every node holds an opinion of all other nodes. In this case, G could be then:

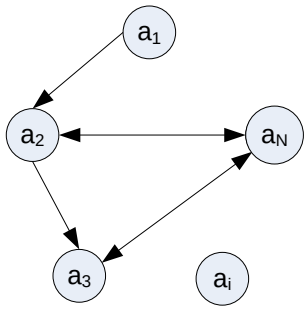


There are $1 + 3 + 1 + 0 + 0 = 10$ mutual opinion associations and there can be no simple generalised formulation for the sum outside of a numerical analytical algorithmic analysis (J. A. Bondy and Murty 2008) (Bertsekas and Tsitsiklis 1997).

It is not necessary that opinions are held mutually. G might be:

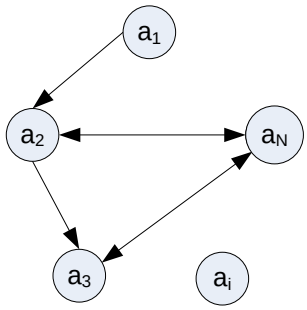


In this case, a_1 has formed an opinion of a_2, but a_2 does not hold a reciprocal opinion of a_1. Nodes a_2 and a_i maintain a mutual opinion association. It is also possible that a node does not hold any opinions of other nodes, as with some node a_i in this graph, G:



Node a_i may have only recently been introduced to the System and is yet to interact with other nodes and so, has not formed any opinions or gained a reputation.

3.2.3 Association Matrices

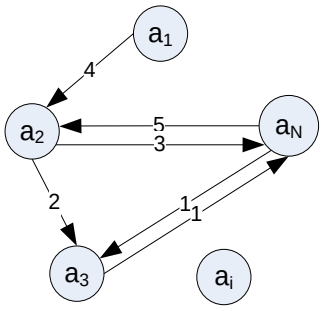


All these associations can be represented in *graph theoretical* matrix form. For graph G above, M is a square hence N by N dimensional, diagonally zero (since a node does not hold an opinion of itself), binary association matrix:

$$M = \begin{bmatrix} & a_1 & a_2 & a_3 & a_i & \dots & a_N \\ a_1 & 0 & 1 & 0 & 0 & \dots & 0 \\ a_2 & 0 & 0 & 1 & 0 & \dots & 1 \\ a_3 & 0 & 0 & 0 & 0 & \dots & 1 \\ a_i & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_N & 0 & 1 & 1 & 0 & \dots & 0 \end{bmatrix}$$

Where each row represents the outbound associations of each node and a '1' indicates an association between nodes and '0' indicates no association (J. A. Bondy and Murty 2008).

Further, for a single arbitrarily determined component of one node's opinion of another node, we can give weight to an opinion beyond a simple binary signifier of its existence. Consider the case where each node forms an opinion of other nodes with a rating from the closed set $\{1,2,3,4,5\}$ (of arbitrary significance). We can represent this graphically by the graph, G:



With a corresponding association matrix, M:

$$M = \begin{bmatrix} 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 \end{bmatrix}$$

Mutually associated opinions do not have to share weight as in the case of a_2 with a_N. Nodes may not share a mutual respect for each other.

These graphs take a similar form to the representation of parallel algorithms by *Directed Acyclic Graphs (DAG)* and are loosely, System topologies (Bertsekas and Tsitsiklis 1997) (Schneider 1977). The representation here does not enforce the acyclic property.

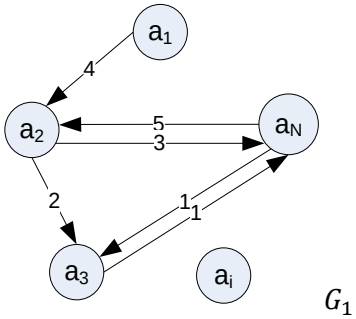
Definition: Each row of the matrix M defines the opinions of a particular node towards other nodes in the System and is an *opinion vector (or profile)*.

We can observe the opinion of node a_1, as the vector $[0 \quad 4 \quad 0 \quad 0 \quad 0]$ and a_2 similarly, $[0 \quad 0 \quad 2 \quad 0 \quad 3]$ for this component of opinion.

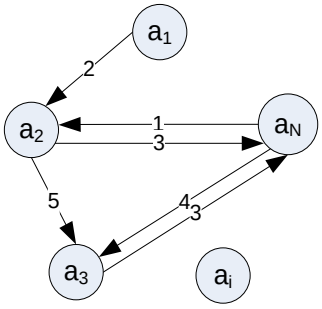
Definition: Each column of the matrix M represents a node's reputation and is a *reputation vector (or profile)*.

We can observe the reputation of node a_1, as the vector $[0 \quad 0 \quad 0 \quad 0 \quad 0]^T$ (no opinion of a_1 is held) and a_2 similarly, $[4 \quad 0 \quad 0 \quad 0 \quad 5]^T$ (where $[.]^T$ is the transpose of a vector).

We can then naturally extend this approach to opinions of multiple components. For each opinion component, we can draw a graph G_i and derive a corresponding association matrix M_i (where $i \in \mathbb{N}$).



G_1



G_2

$$M_1 = \begin{bmatrix} 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 3 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 \end{bmatrix}$$

Each node holds a *complete opinion* of another node or not at all. That is, an opinion must consist of all components – zeros are present in all the same places in M_1 and M_2 while none, some or all of the opinion values differ. In this case, opinion has two components. It does not have to be the case the zero represents a 'null' opinion. This could be represented in any way best suited to the implementation of the framework.

$M_i$is formally defined as:

$$M_i = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 0 & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & 0 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & a_{in} \\ a_{n1} & a_{n2} & a_{n3} & a_{nj} & 0 \end{bmatrix}$$

where $a_{11}, a_{22}, a_{33}, \dots, a_{ij}, \dots, a_{nn} = 0,\ i = j\ \forall\ i,j\ \leq n \in \mathbb{N}$ which is the definition of a zero diagonal matrix (Horn and Johnson 1990), and $a_{ij} \in \mathbb{R}$.

3.2.4 M, R and T

Definition: The set $\boldsymbol{M}$of all matrices $M_j$is a complete *trust space* for the System:

$$\boldsymbol{M} = \{M_1, \dots, M_i, \dots, M_n | \ j, n \in\ \mathbb{N}, j \leq n\}$$

Definition: The complete set of reputation vectors of a node in the trust space is defined as the *Reputation Profile*:

$$R(a_i) = \{M_1(col\ a_i), \dots, M_i(col\ a_i), \dots, M_n(col\ a_i) |\ i, n \in \mathbb{N}, i \leq n\}$$

where $M_i(col\ a_i)$ is the column vector a_i of the matrix M_i.

Definition: The complete set of opinion vectors of a node in the trust space is defined as the *trust profile*:

$$T(a_i) = \{M_1(row\ a_i)^T, \dots, M_i(row\ a_i)^T, \dots, M_n(row\ a_i)^T |\ i, n \in \mathbb{N}, i \leq n\}$$

where $M_i(row\ a_i)^T$ is the transpose row vector a_i of the matrix M_i.

3.2.5 Numerical Example

Considering the following two matrices:

$$M_1 = \begin{bmatrix} 0 & \mathbf{4} & 0 & 0 & 0 \\ 0 & \mathbf{0} & 2 & 0 & 3 \\ \mathbf{0} & \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & \mathbf{0} & 0 & 0 & 0 \\ 0 & \mathbf{5} & 1 & 0 & 0 \end{bmatrix} \qquad M_2 = \begin{bmatrix} 0 & \mathbf{2} & 0 & 0 & 0 \\ 0 & \mathbf{0} & 5 & 0 & 3 \\ \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{3} \\ 0 & \mathbf{0} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 4 & 0 & 0 \end{bmatrix}$$

$\boldsymbol{M} = \{M_1, M_2$ is the complete trust space of the System while the Reputation Profile of a_2 is:

$$R(a_2) = \begin{bmatrix} 4 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 3 \\ 0 \\ 1 \end{bmatrix}$$

and trust profile of a_3 is:

$$T(a_3) = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \\ 0 \\ 0 \\ 3 \end{bmatrix}$$

Here, a_3 has two opinions of a_2 for M_1 and M_2, with values 2 and 3. These elements are therefore present as elements of Reputation Profile of a_2. From $R(a_2)$ and $T(a_3)$, we cannot know the opinion a_2 holds of a_3.

□

3.2.6 Summary

In this section, we formally described the structure of trust in an Emerging System (graphs and association matrices) and defined mathematical entities that describe the trust System (trust spaces,

Reputation Profiles and trust profiles). We established completeness for these entities as the set of all components.

Weighted arcs in graphs denoted the Trust Values between nodes and the direction of the opinion held.

Node Systems can be partitioned in some cases and change over time.

We considered a complete trust space comprising of two components from which we determined the trust and Reputation Profiles of a node in the System.

This section defines the fundamental nomenclature of the trust framework. This is an original contribution to the subject and establishes a flexible foundation. All further sections build on these concepts.

In the following section, we consider how the elements of the trust profile vector are established with respect to the opinions of others within the System. We determine how *final Trust Values* are determined from *initial Trust Values*.

The next section elaborates on the fundamental principles established in this section.

3.3 Final Reputation Profiles

3.3.1 Introduction

Now that we have the formal definitions of the mathematical elements that describe the trust framework for a System of nodes, specifically an Emerging System, we need to consider how these Trust Values are established.

The *initial opinion values* that constitute any single node's trust vector for some trust component, are derived from the node's experience of other nodes within the System. Each trust component is defined as part of the System and will be arbitrarily considered at a theoretical level.

Considering Trust Values in relation to other nodes in the System, forms a local consensus that is of the System and manages potential outliers that may unreasonably skew opinion.

Some examples of *trust components* might be the communication efficiency between network elements, the reliability of a processor to carry out some work, or some more elaborate, specific criteria inherent in a software application running on a device. There are no limits on the size or complexity of a defined trust component set.

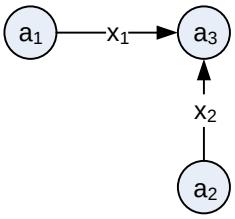
Here we will concern ourselves at least initially, with the underlying trust framework of the System and arbitrarily observe initial opinion values for *trust components*.

In this section, we describe how a *final Trust Value* can be determined from an *initial Trust Value*.

This *final Trust Value* is arrived at with consideration for the opinions of other nodes in the System for the same *trust component*. That is, a node's final opinion of another node, is influenced by all other local nodes that hold similar opinions of the same node for some *trust component*.

3.3.2 Final Trust Values

Considering the case:



The initial trust space for this System of a single trust component is:

$$M_1 = \begin{bmatrix} 0 & 0 & x_1 \\ 0 & 0 & x_2 \\ 0 & 0 & 0 \end{bmatrix}$$

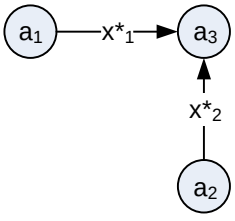
with the set of Reputation Profiles in the trust space:

$$R(a_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad R(a_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad R(a_3) = \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix}$$

and corresponding trust profiles:

$$T(a_1) = \begin{bmatrix} 0 \\ 0 \\ x_1 \end{bmatrix} \qquad T(a_2) = \begin{bmatrix} 0 \\ 0 \\ x_2 \end{bmatrix} \qquad T(a_3) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Then, there is a System of functions $f(x_1, x_2)$ that determines the final Trust Values x_1^*, x_2^*, for the nodes in the System:



This function is dependent on all Trust Values of the other local nodes in the System.

We have the following trust space:

$$M_1^* = \begin{bmatrix} 0 & 0 & x_1^* \\ 0 & 0 & x_2^* \\ 0 & 0 & 0 \end{bmatrix}$$

□

3.3.3 Numerical Example

By way of numerical example, consider the case where $f(x_1, x_2)$, in words, "closes the gap" between the two initial Trust Values by 5% of the difference between the two, resulting in a System of equations, $f(x_1, x_2)$, such that:

$$x_1^* = x_1 + \frac{1}{20}(x_2 - x_1)$$

$$x_2^* = x_2 + \frac{1}{20}(x_1 - x_2)$$

In this case, we are considering a single component trust space with the minimum number of influential nodes (before we have a truly arbitrary case) with a linear System of equations describing their association over a single iteration.

For completeness, we can observe an example case where the initial trust space is:

$$M_1 = \begin{bmatrix} 0 & 0 & 12 \\ 0 & 0 & 7 \\ 0 & 0 & 0 \end{bmatrix}$$

Then, the final trust space becomes:

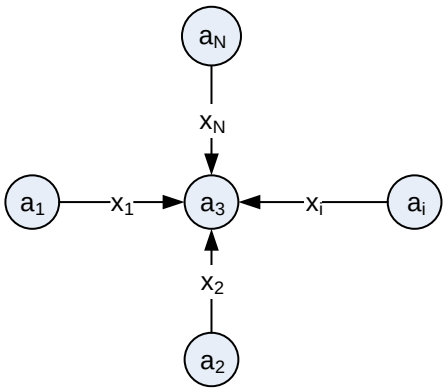
$$M_1^* = \begin{bmatrix} 0 & 0 & 11\frac{3}{4} \\ 0 & 0 & 7\frac{1}{4} \\ 0 & 0 & 0 \end{bmatrix}$$

with associated reputation and trust profiles.

□

3.3.4 General Case

Having considered a single case, we can now extend this System to many nodes and generalise the result. Graph G_1, represents the case of a System with N nodes, $A = \{a_1, \dots, a_i, \dots, a_N\}$ and $\mathbf{x} = \{x_1, \dots, x_i, \dots, x_N\}$ opinions where $\mathbf{x}$ is the initial Reputation Profile vector, $R(a_3)$:



with $R(a_3) = \begin{bmatrix} x_1 \\ x_2 \\ x_i \\ x_N \end{bmatrix}$

Now we need a more generalised form of the System of equations that represent the final Trust Values for Reputation Profile, $R(a_3)$.

Definition: We introduce the notation $\mathbf{x}_{-\mathbf{i}}$ to represent the set (Alpcan, Rencik et al. 2010):

$$\{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N\}$$

That is, the vector $\mathbf{x}$ of all Trust Values, less the Trust Value x_i.

So, we can generalise our System that "closes the gap" between Trust Values but in this case, the Trust Values move closer to the mean Trust Value of the Reputation Profile excluding x_i, by 5%. Our System of equations becomes:

$$f(x_i, \mathbf{x_{-i}}) = x_i - \frac{1}{20}(x_i - \frac{1}{N-1}\sum_{j \neq i}^{N} x_j)$$

where we are considering the mean value of the Reputation Profile excluding the Trust Value being established hence, we have $N - 1$ elements and the need for additional notation.

We can denote:

$$\bar{\mathbf{x}}_{-\mathbf{i}} = \frac{1}{N-1}\sum_{j \neq i}^{N} x_j$$

Which yields a System of equations of the form:

$$f(x_i, \mathbf{x_{-i}}) = x_i - \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}}) \tag{3.1}$$

Again, we are considering a single component trust space with a linear System of equations describing their association over a single iteration, but with any whole number of influential nodes.

3.3.5 Numerical Example

By way of numerical example, consider the initial Reputation Profile:

$$R(a_1) = [5 \quad 1 \quad -4 \quad -3 \quad -4 \quad 1 \quad 5 \quad 5 \quad -4 \quad 2]^T$$

with $N = 10$,

$$\bar{\mathbf{x}}_{\mathbf{-i}} = \{-0.1111, 0.3333, 0.8889, 0.7778, 0.8889, 0.3333, -0.1111, -0.1111, 0.8889, 0.2222\}$$

(note that in this case $\bar{\mathbf{x}}_{\mathbf{-i}}$ is calculated using the *initial* reputation Trust Values only, not any final reputation values), and $\mathbf{x} = \{5,1,-4,-3,-4,1,5,5,-4,2\}$.

Then:

$$\mathbf{x}^* = \{4.7444, 0.9667, -3.7556, -2.8111, -3.7556, 0.9667, 4.7444, 4.7444, -3.7556, 1.9111\}$$

subject to (3.1) and the final Reputation Profile is (to two decimal places):

$$R^*(a_1) = [4.74 \quad 0.97 \quad -3.76 \quad -2.81 \quad -3.76 \quad 0.97 \quad 4.74 \quad 4.74 \quad -3.76 \quad 1.91]^T$$

The table below shows the change in Trust Values from initial to final:

Initial Trust Value	Final Trust Value	Value Change
5	4.74	-0.26
1	0.97	-0.03
-4	-3.76	0.24
-3	-2.81	0.19
-4	-3.76	0.24
1	0.97	-0.03
5	4.74	-0.26
5	4.74	-0.26
-4	-3.76	0.26
2	1.91	-0.09

Figure 5 Initial and final Trust Values of a Reputation Profile for a node in the System.

It makes intuitive sense that the change in value should be similar for similar initial Trust Values as this is a direct substitution in the calculation of the mean.

It is clear that in a single application or *iteration* of the calculation, the final trusts values are tending towards the mean of the other initial Trust Values. Further iterations would see further convergence.

We have seen a change in Trust Value influenced by a derivation from the Trust Values of local nodes.

□

3.3.6 Summary

In this section, we explored how final Trust Values are established from initial values my means of a System of equations that describe the dependency of a node's opinion upon the opinion of others local in the System. We considered the simplest case, the more general case and a numerical examples. Further, we introduced the notation $\mathbf{x_{-i}}$ to represent a closed set containing all values of $\mathbf{x}$, excluding x_i.

System consensus is obtained by considering the trust profiles of other local nodes in the System when determining the final Reputation Profile of a node. This is a better representation of a node's trust than any single Trust Value assigned to it.

We considered some numerical examples to elucidate the principles in the section.

In the next section, we consider a trust space of multiple components. Each node holds a Reputation Profile for each component in the complete trust space.

We formally define the *completeness* of opinion and reputation, and define the *dimension* of a trust space.

3.4 Multi-Component Trust Spaces

3.4.1 Introduction

So far we have considered trust spaces of a single component. Trust spaces typically would consist of many more components, each component representing a trust criteria valued by the System.

Appealing to previous examples, the trust criteria for nodes in an Emerging System might include responsiveness, speed, proximity, accuracy, reliability, scale and any other criteria deemed important to the effective functioning of the System to achieve some goal.

As in the previous section, we will explore how a final Reputation Profile is obtained from the trust profiles of the System as a consensus of local nodes in the System. In this case, however we will consider dependences between the components of a trust space, not just within the same component trust sub-space. From our example previously again, the System might consider a responsiveness trust criteria as a derivative of speed, proximity and reliability.

3.4.2 Trust Space Revisited

The set $\boldsymbol{M}$ of all matrices M_i representing each trust component in the System, is a complete trust space for the System:

$$\boldsymbol{M} = \{M_1, \dots, M_i, \dots, M_n | \, j, n \in \mathbb{N}, j \leq n\}$$

Definition: The dimension of $\boldsymbol{M}$ is the number of elements in $\boldsymbol{M}$, formally:

$$\dim(\boldsymbol{M}) = n$$

and is also denoted:

$$\boldsymbol{M}^n$$

A final Reputation Profile can be established for a node with consideration for initial Trust Values and the Trust Values of other local nodes explicitly declared from a trust sub-space component.

Definition: The trust profile of a_i in the trust sub-space component M_j is $T_j(a_i)$ and the Reputation Profile is $R_j(a_i)$. More formally:

$$R_j(a_i) \text{ and } T_j(a_i) \in M_j$$

Definition: A *complete opinion* of a node a_i is the set of all trust profiles in all trust space components:

$$T(a_i) = \{T_1(a_i), \dots, T_j(a_i), \dots, T_n(a_i) | \, j, n \in \mathbb{N}, j \leq n\}$$

Definition: A *complete reputation* is:

$$R(a_i) = \{R_1(a_i), \dots, R_j(a_i), \dots, R_n(a_i) | \, j, n \in \mathbb{N}, j \leq n\}$$

A node must hold a complete opinion of another node, even if that opinion is zero or neutral. This condition should be upheld by the application making use of the System trust layer.

3.4.3 Example

Consider a two dimensional trust space $\boldsymbol{M}^2$ ($\dim(\boldsymbol{M}) = 2$) with the System of equations for Trust Values in M_1 that we have seen previously:

$$f(x_i, \mathbf{x_{-i}}) = x_i - \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}})$$

and the System of equations in M_2:

$$g(y_i, \mathbf{y_{-i}}) = y_i = y_i^*$$

is the trivial case where initial and final Trust Values are the same.

There is no reason why the function defined for final Trust Values could not be similar for groups of nodes in a System or even be unique for all nodes in the System, regardless of their membership to a trust space component. As part of a framework, it reduces complexity without these cases but it is not beyond the bounds of what is considered here.

We can have a case where M_1 is dependent on M_2. For example, f could be:

$$f(x_i, \mathbf{x_{-i}}, y_i, \bar{\mathbf{y}}_{\mathbf{-i}}) = x_i - \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}}) - \frac{1}{100}(y_i - \bar{\mathbf{y}}_{\mathbf{-i}}) \quad (3.2)$$

such that, the Trust Values of $\mathbf{x}^*$ are dependent on the values of $\mathbf{y}^*$ (which in this case, are trivial).

Consensus changes to Trust Values can be considered a *correction* of the Trust Value with respect to some System derived factor. In practice, though it is not necessary, we can enforce a similar bounded scale for Trust Values for all trust space components. In this example, we can enforce membership to the real and closed set:

$$x_i, y_i \in [-5,5] \subset \mathbb{R}$$

In words, the trust profile of a node in the System is corrected by an overall 5% deviation from the average for one component of the trust space and by overall 1% deviation from the mean of a second trust space component.

□

3.4.4 Numerical Example

Consider the two Reputation Profiles for a_1 in M_1 and M_2:

$$R_1(a_1) = [5 \quad 1 \quad -4 \quad -3 \quad -4 \quad 1 \quad 5 \quad 5 \quad -4 \quad 2]^T \text{ and}$$

$$R_2(a_1) = [1 \quad 2 \quad 2 \quad -1 \quad 1 \quad -1 \quad -4 \quad 5 \quad 3 \quad 1]^T$$

subject to (3.2). We are going to establish the final Reputation Profile in $M_1$of a_1, that is $R_1^*(a_1)$.

Since we will shall assume $R_2(a_1) = R_2^*(a_1)$, we need only consider $R_2(a_1)$. We can consider the third term of f as a constant by calculating g vector first:

$$g(y_i, \mathbf{y_{-i}}) = \frac{1}{100}(y_i - \bar{\mathbf{y}}_{\mathbf{-i}})$$

$$= [0.0011 \quad 0.0122 \quad 0.0122 \quad -0.0211 \quad 0.0011 \quad -0.0211 \quad -0.0544 \quad 0.0456 \quad 0.0233 \quad 0.0011]^T$$

with:

$$\bar{\mathbf{y}}_{\mathbf{-i}} = \{0.8889, 0.7778, 0.7778, 1.1111, 0.8889, 1.1111, 1.4444, 0.4444, 0.6667, 0.8889\}$$

We have:

$$\bar{\mathbf{x}}_{\mathbf{-i}} = \{-0.1111, 0.3333, 0.8889, 0.7778, 0.8889, 0.3333, -0.1111, -0.1111, 0.8889, 0.2222\}$$

then:

$$R^*(a_1) = [4.74 \quad 0.95 \quad -3.77 \quad -2.79 \quad -3.76 \quad 0.99 \quad 4.80 \quad 4.70 \quad -3.78 \quad 1.91]^T$$

to two decimal places.

The comparative change of Trust Values from initial to final:

Initial Trust Value	Final Trust Value	Value Change
5	4.74	-0.26
1	0.95	-0.05
-4	-3.77	0.23
-3	-2.79	0.21
-4	-3.76	0.24
1	0.99	-0.01
5	4.80	-0.20
5	4.70	-0.30
-4	-3.78	0.22
2	1.91	-0.09

Figure 6 Initial and final Trust Values of a Reputation Profile for a node in a two dimensional trust space System.

Unlike in the previous numerical example (3.3.5), similar initial Trust Values do not yield similar final Trust Values because they are influenced by two components of the trust space in this case.

In the following analysis, inter-component trust space calculations of Trust Values will often be considered without loss of generality, as independent trust spaces. That is, we can restrict our analysis to trust spaces of single components by representing other component Trust Values as constants in any System of equations:

$$f(x_i, \mathbf{x_{-i}}) = x_i + \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}}) + \mathbf{c}$$

This approach is not always possible. It can be dependent on the algorithm used to establish the Trust Values and / or the real-time relationship between the trust space components.

$$\mathbf{c} = g(y_i, \mathbf{y_{-i}}) = y_i = y_i^*$$

□

3.4.5 Summary

In this section, we defined the *dimension* of a trust space in terms of the set of all trust space components.

We defined a *complete* opinion and reputation in terms of profiles for multiple component trust spaces and notation for the membership of trust and Reputation Profiles to trust space components.

We extended a previous example to demonstrate how trust space component dependencies can be considered within Systems of equations that establish final Trust Values from their initial values.

In the next section, we consider the effects of System *Environmental Factors* on final Trust Values. We consider the symmetry of the factors and characteristics of their matrix representation.

3.5 Environmental Factors

3.5.1 Introduction

Within any System, there are factors that affect the determination of Trust Values that are defined for the whole System environment.

These factors can be universal such that all nodes in the System are affected similarly by their influence or they can be similar in nature but vary in weight between nodes. This is the concept of "symmetry" established by Alpcan, Rencik et al. (2010). This symmetry alludes to a conservation of

trust within the System. Environmental Factors can be normalised so that the total trust available in the System is always numerically one.

A System administrator might control *Environmental Factors* to alter the influence of the various facets of the final Trust Function. This could be in response to events outside the System such as a security breach or inside such as a change to the nature of an application. This calibration could be human or machine.

The significance then, of Environmental Factors is that they are not derived from the experiences of the nodes in the System or influenced by a consensus of nodes as we have seen already, rather, they are dictated at a System level.

3.5.2 Horizontally and Vertically Symmetric Environmental Factors

Definition: For each final Trust Function in component trust spaces, we define:

$$E = \{e_1, \dots, e_i, \dots, e_n\}$$

where E is the set of all Environmental Variables, n is the total number of variables and i is the identity of a specific e value.

In the case where all functions in a component of a trust space share the same Environmental Variables, we deem this the *vertical symmetric case*. That is, we have vertically symmetric Environmental Factors in the System that do not differ between final Trust Functions in a component trust space. The System of Environmental Factors is represented by the matrix:

$$E = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1m} \\ e_{21} & e_{22} & \dots & e_{2m} \\ e_{31} & e_{32} & \dots & e_{3m} \\ \vdots & \vdots & \ddots & e_{im} \\ e_{n1} & e_{n2} & e_{nj} & e_{nm} \end{bmatrix}$$

Definition: *Vertical symmetry* defines E such that:

$$e_{kj} = e_{lj}, \forall\ k, l, j$$

Definition: If E is *horizontally symmetric*, then:

$$\sum_{i=1}^{n} e_i = 1$$

such that every $0 \leq e_i \leq 1$. In this way, environment factors have complementary influence on the System such that any increase in influence for one Environmental Factor, decreases the influence of one or many of the other factors.

As in this case:

$$f(x_i, \mathbf{x_{-i}}) = x_i + e_1 \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}}) + e_2 \frac{1}{100}(y_i - \bar{\mathbf{y}}_{\mathbf{-i}}) \quad (3.3)$$

with $E = \{e_1, e_2\}$ and $e_1 + e_2 = 1$. This is a principle of conservation of trust within the System.

There is a balance of influence between other Trust Values in the same trust space component as $x_i$and the dependency on another trust space component, y_i.

In the case of (3.3) where either $e_1$or e_2 are zero, the other Environmental Factor is *strictly dominant* and exactly equal to one.

□

Definition: More generally, in the case where there exists some e_i^* such that:

$$e_i^* \geq e_{-i} \; \forall \; i$$

e_i^* is a *dominant* Environmental Factor.

Definition: In the case where there exists some e_i^* such that:

$$e_i^* > e_{-i} \; \forall \; i$$

e_i^* is the *strictly dominant* Environmental Factor.

□

We adopt familiar notation for element exclusion, e_{-i} to signify the set E less the element e_i.

3.5.3 Numerical Example

We can take (3.2) and write it in the form of (3.3) by letting $e_1 = 1$ and $e_2 = 0$. e_1 is the strictly dominant Environmental Factor since $e_1 > e_2$ and:

$$f(x_i, \mathbf{x_{-i}}) = x_i - \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}})$$

as before, which yields the same final Reputation Profile as we established for (3.1).

3.5.4 Horizontally and Vertically Non-Symmetric Environmental Factors

In the *vertical non-symmetric* case, each final Trust Function in some single component trust space, incorporates Environmental Factors that are not necessarily of the same value.

Definition: *Vertical non-symmetry* defines E such that:

$$e_{kj} \text{ does not necessarily equal } e_{lj}, \forall\ k, l, j$$

Definition: If E is *horizontally non-symmetric*, then:

$$\sum_{i=1}^{n} e_i \text{ does not necessarily equal 1}$$

We can extend this further into multidimensional trust spaces by denoting the component trust space in which the Environmental Factors apply by E_i where i corresponds to identifier for the component trust space.

Definition: The complete *Environmental Factor space* then, is the set:

$$\boldsymbol{E} = \{E_1, \dots, E_i, \dots E_n\}$$

□

3.5.5 Numerical Example

Consider the System of final Trust Value equations:

$$f(x_i, \mathbf{x_{-i}}) = x_i - e_{i1}\ \frac{1}{20}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}}) - e_{i2}\mathbf{c}$$

We have a two dimensional trust space where $\mathbf{c}$ is a trust component in M_2.

Setting the initial Reputation Profile:

$$R_1(a_1) = [1 \quad -3 \quad -4 \quad 1 \quad 5]^T$$

Setting $\mathbf{c}$ as a sub-vector of the vector used in (0):

$$\mathbf{c} = [0.0011 \quad 0.0122 \quad 0.0122 \quad -0.0211 \quad 0.0011]^T$$

The vertical non-symmetric Environmental Factor space with $m = 2$ and $n = 5$ is of the form:

$$E_1 = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \\ \dots & \dots \\ e_{51} & e_{52} \end{bmatrix}$$

Setting horizontally symmetric values for E_1:

$$E_1 = \begin{bmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \\ 0.1 & 0.9 \end{bmatrix}$$

Then, we have the final Reputation Profile (to two decimal places):

$$R_1^*(a_1) = [0.97 \quad -2.95 \quad -3.75 \quad 1.02 \quad 4.97]^T$$

Initial Trust Value	Final Trust Value	Value Change
1	0.97	-0.03
-3	-2.95	0.05
-4	-3.75	0.25
1	1.02	0.02
5	4.97	-0.03

Figure 7 Initial and final Trust Values of a Reputation Profile for a node under the influence of two Environmental Factors.

Assuring that each row is strictly equal to one as required, with four strictly dominant Environmental Factors. Overall, e_{i2} is the dominant factor for the E_1 (though not strictly dominant for all Environmental Factors) since:

$$\sum_{i=1}^{n=5} e_{i1} = 1.9 < 3.1 = \sum_{i=1}^{n=5} e_{i2}$$

We can check that these values are correct by ensuring that they sum to n:

$$1.9 + 3.1 = \ 5 = n$$

as required. This is a simple result that follows naturally from the requirement that each of the rows of the Environmental Factor space must be equal to one, if they are to be horizontally symmetric.

□

3.5.6 Summary

In this section, we discussed the nature of *Environmental Factors* within a System and some practical applications particularly for administrators, human or otherwise.

We formally defined *Environmental Factors* and their form as variables in our final Trust Value Systems of equations, and established some of their properties. We defined *dominance* and *strict dominance* of Environmental Factors, and *horizontal* and *vertical symmetry* and *non-symmetry*.

We extended the form of Environmental Factors to a complete *Environmental Factor space* for a trust space of one or many dimensions.

Extending a previous example, we constructed a numerical example showing the application of an *Environmental Factor space* and used it to make some checking observations about the form of the space.

Above all, we have demonstration that the formulation of constructs can define a trust nomenclature as a complete descriptive mathematical foundation for the trust framework.

In the next section, we will extend Trust Functions into more complex forms. Specifically, we will consider the properties of *convex functions* and how their properties are suited for use as Trust Functions.

We consider optimisation problems for *convex functions* – minimisation and maximisation. We consider the existence of optimal solutions locally and globally, and the necessary and sufficient conditions for *optimality*.

3.6 Convex Functions

3.6.1 Introduction

Convex functions (Sahinidis 2002) (Bertsekas and Tsitsiklis 1997) (Constantin P. Niculescu 2004) exhibit properties that are convenient for our consideration of Trust Functions. They are particularly well suited to optimisation problems where the solutions to which require global uniqueness of maximum or minimum points (Boyd and Vandenberghe 2004) (Borwein and Lewis 2000).

In this section, we will introduce convex functions in preparation for their use as Trust Functions.

We describe geometrically how a convex function is defined, then more formally and consider some standard examples. We instate the requirement for concavity as the inverse of convexity (Nocedal and Wright 1999) (Boyd, Ghaoul et al. 1994).

The properties of convex functions that will be of most use to us for Trust Function definitions are identified. We are particularly interested in the property that a convex function on an open set has no more than one minimum (Bazaraa, Sherali et al. 2006).

In this section, we are concern the more traditional concept of a graph as continuous points plotted on axes in some dimensional space.

3.6.2 Convex Functions

A real-valued function $f(x)$ (a function whose values are real numbers), defined on an interval is called *convex* if the points on the line between any two points on the graph of the function, lie above the graph:

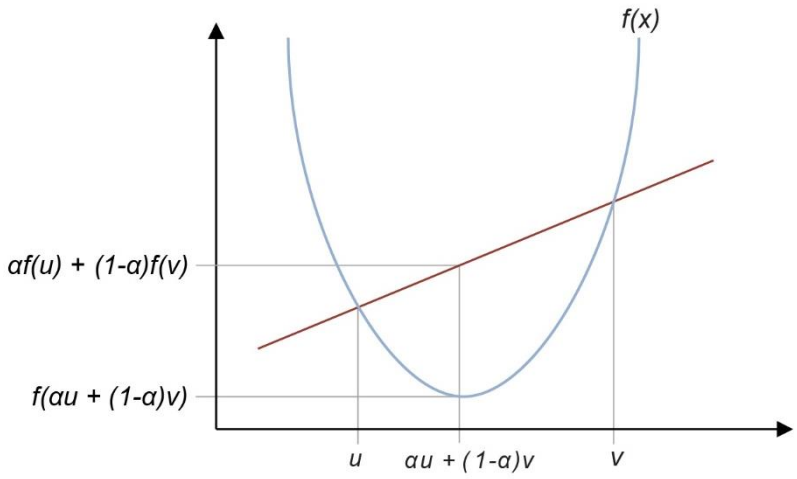


Figure 8 Convex Functions defined by the inequality between any straight line between two points on the graph of the function and the graph of the function itself.

Definition: The *epigraph* of a function $f: \mathbb{R}^n \to \mathbb{R}$ is the set of points lying on or above its graph (Cui 2013):

$$\text{epi}\, f = \{(x, \mu): x \in \mathbb{R}^n, \mu \in \mathbb{R}, \mu \geq f(x)\}$$

and the strict epigraph of the function is:

$$\text{epi}_s\, f = \{(x, \mu): x \in \mathbb{R}^n, \mu \in \mathbb{R}, \mu > f(x)\}$$

The same definitions are valid for a function that takes values in $\mathbb{R} \cup \infty$. In this case, the epigraph is empty if and only if f is identically equal to infinity.

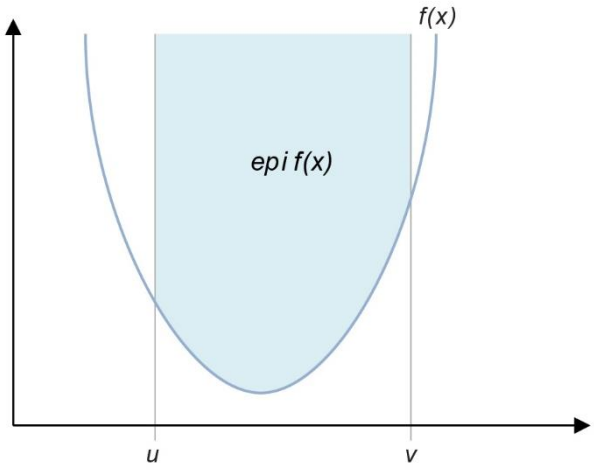


Figure 9 Epigraph of a Convex Function

Definition: Similarly, the set of points on or below the function is its *hypograph*.

Formally, then:

Definition: A subset C of a vector space S (Strang 2003) is said to be *convex* if for every $u, v \in C$ and every $\alpha \in [0,1]$, we have $\alpha u + (1-\alpha)v \in C$.

A function $f: C \to \mathbb{R}$ defined over a convex subset C of a vector space S is said to be *convex* if, for every $u, v \in C$ and every scalar $\alpha \in [0,1]$, we have:

$$f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v)$$

Definition: If this is a strict inequality for every $\alpha \in (0,1)$, then f is said to be *strictly convex*. Note that $(0,1)$ is an *open* set:

$$f(\alpha u + (1-\alpha)v) < \alpha f(u) + (1-\alpha)f(v)$$

for every α, $0 < \alpha < 1$, and $u \neq v$.

Definition: The function f is said to be *concave* if $(-f)$ is convex, and *strictly concave* if $(-f)$ is strictly convex.

Examples: Examples of convex functions include the quadratic function x^2 and the exponential function e^x for any real number x.

3.6.3 Properties

3.6.3.1.1 Additive / Sum

The additive, sum property of convex functions asserts that if f and g are convex functions, then so is:

$$h(x) = f(x) + g(x)$$

This property will be useful in the formulation of compound convex Trust Functions. It will allow us to create increasingly complex convex functions derived from other convex functions without compromising the convexity trait. Since the resulting compound function is also convex, we will continue to be able to identify global minima and assure their uniqueness.

Some further elementary properties (Cui 2013):

3.6.3.1.2 Positive Multiple

The positive multiple of a convex function is convex:

$$f \text{ is convex and } \alpha \geq 0 \Rightarrow \alpha f \text{ is also convex.}$$

3.6.3.1.3 Integrals

Extended sum properties to infinite sums and therefore, integrals:

$$g(x, y) \text{ is convex in } x \Rightarrow \int g(x, y)dy \text{ is also convex.}$$

3.6.3.1.4 Pointwise Maximum

Pointwise maximum corresponds to the intersection of the epigraphs of the two convex functions:

$$f_1, f_2 \text{ are convex} \Rightarrow \max\{f_1(x), f_2(x)\} \text{ is also convex.}$$

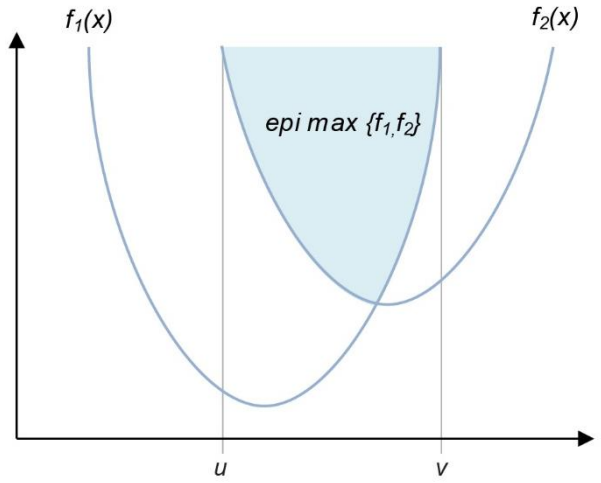


Figure 10 Intersection of two Epigraphs

3.6.3.1.5 Affine Domain Transformation

Under affine transformation of the functional domain:

$$f \text{ is convex} \Rightarrow f(Ax + b) \text{ is also convex.}$$

3.6.4 Optimisation of Functions

Optimisation of functions or *mathematical programming*, is the selection of a best element with regard to some criteria from some set of available alternatives (Dantzig 2014).

In the simplest case, an optimisation problem consists of maximising or minimising a real function by Systematically choosing input values from an allowed set and computing the value of the function. More generally, optimisation includes finding best available values of some *objective function* given a defined domain, including a variety of different types of objective functions and different types of domains (Thomson 1994) (Carathéodory, Hadjisavvas et al. 2001) (Luenberger 1997) (Bertsekas, Nedić et al. 2003) (Vandenberghe 2013).

Convex minimisation, a subfield of optimisation, studies the problem of minimising convex functions over convex sets. The convexity property can make optimisation simpler than the general case since, any *local minimum* must also be a *global minimum* (Luenberger 2003) (Krasnosel'skiĭ and Rutit͡skiĭ 1961).

Convex functions have properties that make them more conducive to optimisation than other function groups. These properties also make them ideal for use as Trust Functions since an optimal solution (maximum or minimum) can be found, a best final Trust Value can be established and assured. Advanced techniques become important in higher dimensions, with functions of higher powers and multiple variables.

Formally then:

Definition: Given a function $f: S \to \mathbb{R}$, where S is a vector space, and a subset $X \subseteq S$, by the optimisation problem:

$$\text{minimise } f(x) \text{ subject to } x \in X$$

we mean the problem of finding an element $x^* \in X$ (called a *minimising element* or an *optimal solution*) such that:

$$f(x^*) \leq f(x) \quad \forall\, x \in X$$

If such an $x^* \in X$ exists, then we use the notation $x^* = \arg\min_{x \in X} f(x)$.

This is often also referred to as the *global minimising solution*, in order to differentiate it from the alternative – a *locally minimising solution* (Başar and Olsder 1999).

Definition: An element $x^o \in X$ is called a locally minimising solution if we can find an $\varepsilon > 0$ such that:

$$f(x^o) \leq f(x) \quad \forall\, x \in N_\varepsilon\,(x^o) \cap X$$

That is, we compare $f(x^o)$ with values of $f(x)$ in that part of a certain ε-neighbourhood of x^o, which lies in X.

For a given optimisation problem, it is not necessary that an optimal solution will exist if the set of real numbers $\{f(x)\colon x \in X\}$ is bounded below and there exists an $x^* \in X$ such that $\{f(x)\colon x \in X\}= f(x^*)$, in which case we have:

$$f(x^*) = \inf_{x\in X} f(x) = \min_{x\in X} f(x)$$

If such an x^*cannot be found, even though $\inf\{f(x)\colon x \in X\}$ is finite, we simply say that an optimal solution does not exist, but we declare the quantity:

$$\inf\{f(x)\colon x \in X\} \text{ or } \inf f(x)$$

as the *optimal value* of the optimisation problem. If $\{f(x)\colon x \in X\}$ is not bounded below, $\inf_{x\in X} f(x) = -\infty$, then neither an optimal solution nor an optimal value exists (Başar and Olsder 1999).

An optimisation problem that involves maximisation instead of minimisation may be converted into a minimisation problem simply by replacing f by $-f$. Any optimal solution of this minimisation problem is also an optimal solution for the initial maximisation problem, and the optimal value of the latter, denoted $\sup_{x\in X} f(x)$, is equal to the minus optimal value of the former. If a *maximising element* $x^* \in X$ exists, then:

$$\sup_{x\in X} f(x) = \max_{x\in X} f(x) = f(x^*)$$

3.6.5 Existence of Optimal Solutions

In the minimisation problem, an optimal solution exists (Başar and Olsder 1999) if X is a finite set, since then there is only a finite number of comparisons to make. If X is not finite, however, existence of an optimal solution is not always guaranteed. It is guaranteed if f is continuous and X is compact. This result is the *Weierstrass Theorem* (Urruty and Lemaréchal 2001) (Donoghue 1969) (Bertsekas, Nedić et al. 2003).

Theorem (Weierstrass): If f is a real-valued continuous function on a non-empty compact domain S, then there exists an $x \in S$ such that $f(x) \geq f(y)$ for all y in S.

3.6.6 Necessary and Sufficient Conditions for Optimality

3.6.6.1 Convexity and Calculus

Definition: A differentiable function of one variable is convex on an interval if and only if its derivative is monotonically non-decreasing on that interval. If a function is differentiable and convex then it is also continuously differentiable (Vandenberghe 2013).

A continuously differentiable function of one variable is convex on an interval if and only if the function lies above all of its tangents (Vandenberghe 2013):

$$f(x) \geq f(y) + f'(y)[x - y]$$

for all x and y in the interval. In particular, if $f'(c) = 0$, then c is a global minimum of $f(x)$.

- A twice differentiable function of one variable is convex on an interval if and only if its second derivative is non-negative there; *this gives a practical test for convexity;*
- If its second derivative is positive at all points then the function is strictly convex, but the converse does not hold.

3.6.6.1.1 Hessian Matrix

Hessian matrix (Strang 2003) or *Hessian* (Horn and Johnson 1990) is a square matrix of second-order partial derivatives of a function. It describes the local curvature of a function of many variables.

Given the real-valued function:

$$f(x_1, x_2, \ldots, x_n)$$

If all second partial derivatives of f exist and are continuous in the domain of the function, then the Hessian matrix of f is:

$$H(f)_{ij}(x) = D_i D_j f(x)$$

where $x = (x_1, x_2, \ldots, x_n)$ and D_i is the differentiation operator with respect to the ith argument.

The Hessian matrix is of the form:

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

The determinant of a Hessian matrix is also referred to as *the Hessian* (Binmore and Davies 2001).

3.6.6.1.2 Jacobian Matrix

The Hessian matrix is related to the *Jacobian matrix* by:

$$H(x) = H(f)(x) = J(\nabla f)(x)$$

That is, the Jacobian matrix is the matrix of first-order derivatives of some function f.

3.6.6.1.3 Definiteness of a Hessian Matrix

We have all the necessary and sufficient conditions for optimality now. Practically, we need a suitable method to determine the definiteness of a Hessian matrix (or any square matrix, for that matter) (Akcigit 2004).

We have the definitions:

Definition: A symmetric n by n real matrix M is said to be *positive semidefinite* if $z^T M z \geq 0$, for any non-zero column vector z of n real numbers.

Moreover, all eigenvalues of M are non-negative (Horn and Johnson 1990) (Strang 2003).

Definition: A symmetric n by n real matrix M is said to be *positive definite* if $z^T M z$ is positive, for any non-zero column vector z of n real numbers (Horn and Johnson 1990) (Strang 2003).

Further, M is also positive semidefinite and $\det(M)$ is non-zero.

A practical approach to determining the state of definiteness for a symmetric matrix is the use of *minors*.

Definition: Let A be an $n \times n$ matrix. A $k \times k$ submatrix of A formed by deleting $n - k$ rows of A, and the same $n - k$ columns of A, is called the *principle submatrix* of A. The determinant of the principle submatrix of A is called a *principle minor* of A.

The definition does not specify which rows and columns to delete, only that their indices must be the same.

For a general 3×3 matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

there is one third-order principle minor, namely $|A|$. There are three second-order principle minors:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}$$

and there are three first-order principle minors:

$$[a_{11}] \qquad [a_{22}] \qquad [a_{33}]$$

Definition: Let A be an $n \times n$ matrix. The kth order principle submatrix of A obtained by deleting the last $n - k$ rows and columns of A is called the kth order *leading principle submatrix* of A and its determinant is called the kth order *leading principle minor* of A.

We will denote the kth order leading principle submatrix of A by A_k, and its kth order leading principle minor by $|A_k|$.

Theorem: Let A be an $n \times n$ symmetric matrix. Then:

1. A is positive definite if and only if all its n leading principle minors are strictly positive;
2. A is negative definite if and only if its n leading principle minors alternate in sign as follows:
$$|A_1| < 0, |A_2| > 0, |A_3| < 0, \ldots$$
3. If some kth order leading principle minor of A is nonzero but does not fit the sign patters above, the A is indefinite (Akcigit 2004).

Theorem: Let A be an $n \times n$ symmetric matrix. Then, A is positive semidefinite if and only if every principle minor of $A \geq 0$. A is negative semidefinite if and only if every principle minor of odd order is ≤ 0 and every principle minor of even order is ≥ 0 (Akcigit 2004).

In summary:

- A continuous, twice differentiable function of several variables is convex on a convex set if and *only if its Hessian matrix is positive semidefinite* on the interior of the convex set;
- Any local minimum of a convex function is *also a global minimum, and*;
- A strictly convex function will have *at most one global minimum*.

3.6.6.2 Strong Convexity

The concept of strong convexity extends and parameterises the notion of strict convexity. A strongly convex function is also strictly convex, but it is not the case that every strictly convex function is strongly convex.

A differentiable function f is called strongly convex with parameter $m > 0$ if the following inequality holds for all points x, y in its domain (Bertsekas, Nedić et al. 2003):

$$\big(\nabla f(x) - \nabla f(y)\big)^T (x - y) \geq m\|x - y\|_2^2$$

An equivalent condition is the following (Nesterov and Nesterov 2004):

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2$$

It is not necessary for a function to be differentiable in order to be strongly convex.

A third definition (Nesterov and Nesterov 2004) for a strongly convex function, with parameter m, is that, for all x, y in the domain and $\alpha \in [0,1]$,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \frac{1}{2}m\alpha(1 - \alpha)\|x - y\|_2^2$$

This definition approaches the definition for strict convexity as $m \to 0$, and is identical to the definition of a convex function when $m = 0$.

Theorem (Clairaut): The mixed derivatives of f are the entries of the main diagonal in the Hessian matrix. Assuming that they are continuous, the order of differentiation does not matter:

$$f_{yx} = f_{xy}$$

If the function f is twice continuously differentiable, then it is strongly convex with parameter m if and only if:

$$\nabla^2 f(x) \geq mI$$

for all x in the domain, where I is the identity and $\nabla^2 f$ is the Hessian matrix, and:

$$\nabla^2 f(x) - mI$$

is positive definite.

This is equivalent to requiring that the minimum eigenvalue of $\nabla^2 f(x)$ be at least m for all x.

If the domain is real, then $\nabla^2 f(x)$ is the second derivative $f''(x)$, so the condition becomes:

$$f''(x) \geq m$$

If $m = 0$, then the Hessian is positive semidefinite then the function is convex, and perhaps strictly convex, but not strongly convex. Equivalently, if the domain is real, then $f''(x) \geq 0$.

Assuming that the function f is twice continuously differentiable, the lower bound of $\nabla^2 f(x)$ implies that it is strongly convex, determinable from Taylor's theorem (Horn and Johnson 1990).

A twice continuously differentiable function f with a real domain, can be characterised as:

- f is *convex* if and only if $f''(x) \geq 0$ for all x;
- f is *strictly convex* if $f''(x) > 0$ for all x which is sufficient, but not necessary;
- f is *strongly convex* if and only if $f''(x) \geq m > 0$ for all x.

In summary for optimal solutions (Başar and Olsder 1999), let $S = \mathbb{R}^n$, and $f: \mathbb{R}^n \to \mathbb{R}$ be a differentiable function (or *functional* as it is a map from a vector space $\mathbb{R}^n$ into its underlying scalar field, $\mathbb{R}$). If X is an open set, a first-order necessary condition for an optimal solution must satisfy:

$$\nabla f(x^*) = 0$$

If in addition, f is twice differentiable on $\mathbb{R}^n$, a second-order necessary condition is:

$$\nabla^2 f(x^*) \geq 0$$

The pair of conditions $\{\nabla f(x^*) = 0, \nabla^2 f(x^*) > 0\}$ is sufficient for $x^* \in X$ to be a locally minimising solution.

These conditions are also sufficient for global optimality if, in addition, X is a convex set and f is a convex function on X.

3.6.7 Fundamental Results

The following important fundamental results help us to derive well-structured convex Trust Functions (Vandenberghe 2013) (Cui 2013). They can also be used in combination with respect to the properties of convex functions, particularly the summation property (Wright 1997) (Nguyen, Strodiot et al. 2000) (Fletcher 1987) (Miller 1999).

3.6.7.1.1 Quadratic

The quadratic function $f(x) = x^2$ has $f''(x) = 2 > 0$ at all points, so f is a convex function. It is also strongly convex and hence strictly convex too, with strong convexity constant 2.

3.6.7.1.2 Polynomial

The polynomial function $f(x) = x^4$ has $f''(x) = 12x^2 \geq 0$, so f is a convex function. It is strictly convex, even though the second derivative is not strictly positive at all points. It is not strongly convex.

3.6.7.1.3 Absolute

The absolute value function $f(x) = |x|$ is convex, even though it does not have a derivative at the point $x = 0$. It is not strictly convex.

3.6.7.1.4 Exponential

The exponential function $f(x) = e^x$ is convex. It is also strictly convex, since $f''(x) = e^x > 0$, but it is not strongly convex since the second derivative can be arbitrarily close to zero (Kingman 1961) (Cui 2013) (Rockafellar 1997). More generally, the function $f(x) = e^{f(x)}$ is logarithmically convex if f is a convex function (Polak 1997) (Diwekar 2003).

3.6.7.1.5 Inverse

The inverse function $f(x) = \frac{1}{x}$ $f''(x) = \frac{2}{x^3}$ which is greater than 0 if $x > 0$, so f is convex on the interval $(0, +\infty)$. It is concave on the interval $(-\infty, 0)$.

3.6.7.1.6 Inverse Quadratic

The inverse quadratic function $f(x) = \frac{1}{x^2}$ with $f(0) = +\infty$, is convex on the interval $(0, +\infty)$ and convex on the interval $(-\infty, 0)$, but not convex on the interval $(-\infty, +\infty)$, because of the singularity at $x = 0$.

3.6.7.1.7 Compound Sum and Scalar Multiples

We can extend the results by combining convex functions by virtue of their additive and scalar multiplicative properties.

The compound quadratic and exponential function composed of convex functions $f(x) = x^2$ and $g(x) = 3e^x$, $h(x) = 2f(x) + g(x)$ has $h''(x) = 4 + 3e^x \geq 4 > 0$ as $x \to -\infty$, at all points, so h is a convex function. It is also strongly convex and hence strictly convex too, with strong convexity constant 4.

□

3.6.7.1.8 Hessian Matrix

Consider the quadratic Trust Function of three variables in a one dimensional trust space with no Environmental Factors:

$$f(x_1, x_2, x_3) = x_1^2 + 2x_2^2 + 3x_3^2 + 2x_1x_2 + 2x_1x_3$$

The first partial derivatives that form the Jacobian matrix (vector in this case) are:

$$\frac{\partial f}{\partial x_1} = 2x_1 + 2x_2 + 2x_3$$

$$\frac{\partial f}{\partial x_2} = 4x_2 + 2x_1$$

$$\frac{\partial f}{\partial x_3} = 2x_1 + 6x_3$$

$$\Rightarrow \nabla f(\mathbf{x}) = J(f(\mathbf{x})) = [2x_1 + 2x_2 + 2x_3, 4x_2 + 2x_1, 2x_1 + 6x_3]$$

We have a Hessian matrix of the form H^{3x3}:

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix}$$

Then:

$$H(f) = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 4 & 0 \\ 2 & 0 & 6 \end{bmatrix}$$

The leading principle minors of the Hessian are $2 > 0$, $4 > 0$ and $8 > 0$ (Eriksen 2010).

Hence, the Hessian is positive definite and the Trust Function, f is strictly convex.

□

3.6.8 A Quadratic Trust Function

To reconcile this exploration of convex functions explicitly with Trust Functions and the definitions of trust elements within a trust space, consider a modified quadratic incarnation of a Trust Function we have seen previously (3.1). The function has not been simplified in order to accentuate its form:

$$f(x_i, \mathbf{x_{-i}}) = \frac{1}{2}\left(\frac{1}{20(N-1)}\left(x_i - \frac{1}{N-1}\sum_{j \neq i}^{N} x_j\right)\right)^2 \qquad (3.4)$$

with a single dimensional trust space. The Trust Function quadratically influences a 5% Trust Value correction weighted by the number of consensus Trust Values considered, represented by $N-1$.

Using the substitutions:

$$A = \frac{1}{20(N-1)} \qquad B = \frac{1}{N-1}\sum_{j\neq i}^{N} x_j = \bar{\mathbf{x}}_{-\mathbf{i}}$$

we have:

$$f(x_i, \mathbf{x}_{-\mathbf{i}}) = \frac{1}{2}\big(A(x_i - B)\big)^2$$

$$= \frac{1}{2}A^2(x_i - B)^2 = \frac{1}{2}A^2(x_i^2 - 2Bx_i + B^2)$$

Then the first derivative is:

$$\frac{\partial f}{\partial x_i} = A^2(x_i - B)$$

with the second derivative condition is:

$$\frac{\partial^2 f}{\partial x_i^2} = A^2 = \left(\frac{1}{20(N-1)}\right)^2 > 0$$

since $N > 0$, hence this function is a strictly convex in x_i and emits a uniquely global minimal solution x_i^* at:

$$\frac{\partial f}{\partial x_i} = A^2(x_i^* - B) = 0$$

$$\Rightarrow x_i^* = B = \frac{1}{N-1}\sum_{j\neq i}^{N} x_j$$

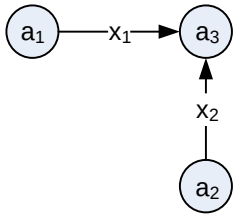
Substitution into $f(x_i^*, \mathbf{x}_{-\mathbf{i}})$ yields:

$$f(x_i^*, \mathbf{x}_{-\mathbf{i}}) = \frac{1}{2}\big(A(B - B)\big)^2 = 0$$

This is not a surprising result due to the nature of the function, cf with f being a single variable quadratic function, for some scalar, c. This follows directly from the fundamental quadratic result (3.6.7.1.1).

□

Consider the elementary three node relationship within a System:



We can take the Reputation Profile to be of two Trust Values, x_1 and x_2 so $N = 2$ (a three node relationship) with a 10% Trust Value correction such that:

$$f_1(x_1, x_2) = \frac{1}{2}\left(\frac{1}{10}(x_1 - x_2)\right)^2$$

and

$$f_2(x_2, x_1) = \frac{1}{2}\left(\frac{1}{10}(x_2 - x_1)\right)^2$$

The Jacobian of f is

$$J(f) = \left[\frac{(x_1 - x_2)}{100}, \frac{(x_2 - x_1)}{100}\right]$$

and the Hessian with a zero determinant is:

$$H(f) = \begin{bmatrix} 1/100 & -1/100 \\ -1/100 & 1/100 \end{bmatrix}$$

In this case:

$$x_1^* = x_2 \Rightarrow f_1(x_1^*, x_2) = 0 \qquad x_2^* = x_1 \Rightarrow f_2(x_2^*, x_1) = 0$$

as expected.

□

3.6.9 Summary

In this section, we considered convex functions and their properties as suitable candidates for Trust Functions.

We provided formal definitions of strict and non-strict convex functions as an algebraic inequality derived from geometric representation, as the consequence of a convex set and as a result following from the characteristics of a function's epigraph.

We gave a formal definition of a function's epigraph as the area above a functions graph between values in the function's domain. The set of points in the area can exclude values on the functions as a necessary requirement for strict convexity.

We listed some fundamental properties of convex functions that will ensure preservation of convexity when we consider complex Trust Functions component-wise. Of particular note is the additive or sum property that ensures convexity after additive concatenation of convex functions.

We explored optimisation of convex functions and the global minimising element or optimal solutions. We eluded to maximisation problems as the direct inverse of minimisation.

We considered the existence of optimal solutions, their first and second order necessary and sufficient conditions for strong and strictly convex functions. We guaranteed the existence of an optimal solution from *Weierstrass' Theorem* when the convex function is continuous and the domain is compact.

Higher dimensional problems lead to a discussion of *Jacobian matrices* as the System of first-order partial derivatives and then, *Hessian matrices* as a System of second-order partial derivatives of a function. *Clairaut*'s *Theorem* contributed to the formulation of these matrix forms and their positive definite and semi-definite characteristics. These matrix forms are well suited to Trust Function Systems with Environmental Factor spaces.

Two key results are:

1. any local minimum of a convex function is also a *global minimum*, and;
2. a strictly convex function will have *at most one global minimum*.

Using the conditions for convexity, we provided the results of some fundamental functions as the foundation components for more complex convex Trust Functions.

This section consolidates some important results for our future consideration of Trust Functions of several variables. It is not intended as an exhaustive exploration of all the material but as a necessary summary of the results pertinent to our future derivations of Trust Functions and their behaviour. Ensuring that a Trust Function is convex affords us some favourable characteristics most importantly, uniqueness of a global minimum solution.

In the next section we consider all these previous concepts in the context of *Game Theory*. We extend our final Trust Functions to more complex cases and define them as *cost functions* in a game

theoretical fashion. We formally introduce equilibrium in the System and specifically, as the fundamental concept behind our trust framework, *Nash equilibrium*.

3.7 Game Theory

3.7.1 Introduction

In the previous section, we consider the properties and form of convex functions, and how suitable they are as Trust Functions. The existence and uniqueness of minimum solutions make them analytically well suited. In this section, we will build on these desirable characteristics in their use as *cost functions* in theoretical *games*.

A more sophisticated approach to trust computation than those we have seen thus far is a game theoretical one. Game Theory contributes a wealth of rich, established mathematical tools and approaches to the analysis of trust. Trust games can be played out over time and under changing circumstances to determine different outcomes and states of agreement between participants.

Emerging Systems can be modelled mathematically and trust evaluation can be analysed based on some simulated behaviour, influences and initial states, within a trust game. The trust game allows each node to re-evaluate and update its individual opinion of another node based on some criteria.

In this section, we explore the definitions of types of games and from these, identify the type of game upon which our trust framework is based.

We shift our notation slightly to map what has been defined in previous sections to a game theoretical environment. We want to establish a game theoretical approach in our already defined trust space nomenclature without compromising the theory. To this end, we establish the finding of solutions to trust games as our approach to determining final Trust Values in trust and Reputation Profiles.

Nodes are often referred to as *agents* or *players* and the reputation space of these nodes is referred to as the *decision space* of the game.

We evolve our Trust Functions to game theoretical cost functions, the convex minimisation characteristics of which allow us to ensure equilibrium and stability results.

We formally introduce *Nash Equilibria* as solutions to trust games and establish suitable criteria for their existence and uniqueness.

3.7.1.1 Definitions

For Game Theoretical nomenclature, we stipulate the following equivalent definitions:

Trust Framework	Game Theory
Process to establish final Trust Values	Game
Node	Agent / player
Trust Function	Cost / objective function
Reputation Profile	Decision space
Final Trust Value	Nash Equilibrium (NE)

□

3.7.2 Game Types

Game theory deals with strategic interactions among multiple, rational decision makers, called agents or players or nodes, with each agent's ordered preference among multiple alternatives captured in an objective or cost function (Trust Function) for that agent, which the agent attempts to maximise or minimise.

For a non-trivial game, the objective function of an agent depends on the choices of at least one other agent, and generally of all the agents, and hence agents cannot simply optimise their own objective functions independently of the choices of other agents. This brings in a coupling between the actions of the agents, and binds them together in decision making even in a non-cooperative situation.

If the agents were able to enter into a cooperative agreement so that the selection of actions or decisions is done collectively and with full trust, so that all agents would benefit to the extent possible, and no inefficiency would arise, then we would be in the realm of cooperative game theory, with issues of bargaining, coalition formation, excess utility distribution, etc. (Başar and Olsder 1999). This is the case with closed Systems where a central trust authority, a known inventory and the fair distribution of resources can be assumed.

In the case of Emerging Systems, from the literature (Shah, Jan et al. 2012) (Urpi, Bonuccelli et al. 2003) we have seen that no such cooperative agreement can be assumed and we consider only non-cooperative games. We will determine the trust framework's game type from the definitions of possible types.

3.7.2.1 Non-Cooperative

Fundamental to this work is the concept of *non-cooperation*. The application of the trust framework is to Emerging Systems. These Systems assume no cooperative characteristics where nodes have no inherent knowledge of each other, act independently and ultimately, have selfish goals.

From Başar and Olsder (1999), for a precise formulation of a non-cooperative game, we have to specify:

1. the number of agents;
2. the possible actions available to each agent and any constraints that may be imposed on them;
3. the cost function of each agent to be optimised (minimise or maximise);
4. any time ordering of the execution of the actions if the agents are allowed to act more than once;
5. any information acquisition that takes place and how the information available to an agent at each point in time depends on the past actions of other agents, and;
6. whether there is an agent akin to nature whose action is the outcome of a probabilistic event with a fixed distribution.

The definitions of game types originates in Başar and Olsder (1999) also:

3.7.2.2 Non-Zero / Zero Sum

A non-cooperative game is *nonzero-sum* if the sum of the agents' objective functions cannot be made zero after appropriate positive scaling and/or translation that do not depend on the agents' decisions.

A two-agent game is zero-sum if the sum of the objective functions of the two agents is zero or can be made zero by appropriate positive scaling and/or translation that do not depend on the decisions of the agents. If the two agents' objective functions add up to a constant (without scaling or translation), then the game is deemed *constant sum*.

3.7.2.3 Finite / Infinite

A game is a *finite* game if each agent has only a finite number of alternatives that is, the agents pick their actions from finite sets. Otherwise the game is an *infinite* game. Finite games are also known as matrix games. An infinite game is said to be a *continuous-kernel game* if the action sets of the agents are continua and the agents' objective functions are continuous with respect to the actions of all agents.

3.7.2.4 Static / Dynamic / Differential

A game is *static* if agents have access to only the a priori information shared by all, and none of the agents has access to information on the actions of any of the other agents; otherwise we have a *dynamic* game.

A *dynamic* game is a *differential* game if the evolution of the decision process controlled by the agents over time takes place in continuous time, and generally involves a differential equation. If it takes place over a discrete-time horizon, a dynamic game is sometimes called a *discrete-time* game.

3.7.2.5 Deterministic / Stochastic or Pure / Mixed

A game is *deterministic* or *pure* if the agents' actions uniquely determine the outcome, as captured in the objective functions, whereas if the objective function of at least one agent depends on an additional variable (state of nature) with a known probability distribution, then we have a *stochastic* or *mixed strategy* game.

3.7.2.6 Single / Multi-Act

A game is a *single-act* game if every agent acts only once. Otherwise the game is *multi-act*.

3.7.2.7 Complete / Incomplete Information

A game has *complete information* if the description of the game (that is, the agents, the objective functions, and the underlying probability distributions if stochastic) is common information to all agents; otherwise we have an *incomplete information* game.

From the definitions of game types, we can establish the game type best suited to our trust framework for Emerging Systems as:

1. *Non-zero sum* as Emerging Systems do not limited the Trust Values attributable to any node nor is any node attributed a Trust Value to the detriment of any other;
2. *Continuous-kernel, infinite* as a consequence of the Trust Value (cost) functions being continuous and unbounded in the domain. As we are trying to minimise a convex function, we are assured of a global, unique minimum particularly in the quadratic case;
3. *Static* since the whole trust space is shared by all nodes in the System in principle. There is potential for partitioning within the System but then, the different factions of the System should be considered independently as isolated games;
4. *Deterministic* or *pure strategy* since there are no probabilistic factors that determine the establishment of final Trust Values. We do consider Environmental Factors and other components of the trust space's influence on the Trust Function, however;
5. The game can be considered iterative *single-act.* Information is only required from one previous iteration (or an initial condition) but the number of iterations is not bounded except by some convergence condition. The game is completed in its entirety, each iteration. Considering the game as the complete process of obtaining a final Trust Value, then the game is *multi-act*, and;

6. *Complete information* due to all limited information being available in the System. No node or set of nodes have access to any exclusive information.

The establishment of these game qualities, completes the fulfilment of the formulation criteria (3.7.2.1) for a non-cooperative game, as required.

□

3.7.3 Nash Equilibrium

3.7.4 Sufficient Conditions

If a game has a unique Nash equilibrium and is played among agents under certain conditions, then the Nash equilibrium strategy set will be adopted. Sufficient conditions to guarantee that a Nash equilibrium game is played are (Aumann and Brandenburger 1995) (Nash 1951):

1. The agents all will do their utmost to maximise their expected payoff as described by the game;
2. The agents are flawless in execution;
3. The agents have sufficient intelligence to deduce the solution;
4. The agents know the planned equilibrium strategy of all of the other agents;
5. The agents believe that a deviation in their own strategy will not cause deviations by any other agents, and;
6. There is common knowledge that all agents meet these conditions, including this one. So, not only must each agent know the other agents meet the conditions, but also they must know that they all know that they meet them, and know that they know that they know that they meet them, and so on.

3.7.5 Formulation

A node a_i forms an opinion, $x_i \in \mathbb{R}$ of a particular node after consultation with other nodes in the System (as we have seen previously). The set of opinions of all nodes is represented by the vector:

$$\mathbf{x} = [x_1, \dots, x_i, \dots, x_N] \in X \subset \mathbb{R}^N$$

which defines the *decision space* (Reputation Profile) for the game where $x_i = 0$ is the default and neutral opinion position. Accordingly, all $x_i > 0$ are positive and $x_i < 0$ are negative opinions.

Definition: The Nash equilibrium (NE) of a trust game is defined as the set of opinions, $\mathbf{x}^* = [x_i^*, \mathbf{x}_{-\mathbf{i}}^*]$ and the corresponding costs J^*, such that:

$$J_i(x_i^*, \mathbf{x}_{-\mathbf{i}}^*) \leq J_i(x_i, \mathbf{x}_{-\mathbf{i}}^*) \; \forall \; x_i \in \mathbb{R}, \forall \; i$$

The set of nodes A, the decision space X and the cost function J_i, can be considered together to define a non-cooperative trust game, $G_i(A, X, J)$ whereby each individual node a_i minimises its own cost J_i by making a trust decision $x_i \in \mathbb{R}$ given the decisions of others $\mathbf{x_{-i}}$. Formally:

$$x_i := \arg\min_{x_i} J_i(x_i, \mathbf{x_{-i}})$$

For complete proofs using the Kakutani fixed point theorem see Nash (1950) and an alternate proof using the Brouwer fixed-point theorem see Nash (1951).

We can then establish the existence of the NE for convex sets:

Theorem: For each $i \in \mathbb{N}$, let X_i be a closed, bounded and convex subset of the a finite-dimensional Euclidean space, and the cost functional $J_i: X_1 \times \ldots \times X_N \to \mathbb{R}$ be jointly continuous in all its arguments and strictly convex in x_i for every $x_j \in X_j$, $j \in \mathbb{N}$, $i \neq j$. Then the associated N-person non-zerosum game admits a NE in pure strategies.

Moreover, we establish the result for continuous functions:

Theorem: For each $i \in \mathbb{N}$, let $X_i = \mathbb{R}^{m_i}$, the cost functional $J_i: X_1 \times \ldots \times X_N \to \mathbb{R}$ be jointly continuous in all its arguments and strictly convex in x_i for every $x_j \in X_j$, $j \in \mathbb{N}$, $i \neq j$. Furthermore, let:

$$J_i(x_i, \mathbf{x_{-i}}) \to \infty \; as \; |x_{-i}| \to \infty \; \forall \; x_{-i} \in X_{-i}, i \in \mathbb{N} \tag{3.5}$$

Then, the associated N–person nonzero-sum game admits a NE in pure strategies.

For complete proofs, see Başar and Olsder (1999).

These are the key results for establishing *optimal* final Trust Values in reputation and trust profiles.

□

3.7.6 Quadratic Games

Having established the type of game suited to our trust framework, we will restrict the class of non-cooperative game even further.

The following results are from Başar and Olsder (1999) with some notational translation. They establish the general quadratic form of a cost function, determining NE within the game and its uniqueness.

Quadratic games will form the class of game used within the trust framework. From our fundamental convex function results, quadratic functions (3.6.7.1.1) and compound and scalar multiples (3.6.7.1.7) we are able to establish the convexity of quadratic functions rigorously.

A general quadratic cost function for a_i, which is strictly convex in its cost function, can be written as:

$$J_i = \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} x_j' R_{jk}^i x_k + \sum_{j=1}^{N} x_j' r_j^i + c_i \tag{3.6}$$

where $x^j \in X^j = \mathbb{R}^{m_i}$ is the m_i-dimensional decision variable of a_j, R_{jk}^i is the $(m_j \times m_k)$-dimensional matrix with $R_{ii}^i > 0$, r_j^i is an m_i-dimensional vector and $c_i$is a constant.

Quadratic cost functions are of particular interest in game theory (Başar and Olsder 1999), firstly because they constitute second-order approximation to other types of nonlinear cost functions and secondly, because they are analytically tractable, admitting, in general, closed-form equilibrium solutions which provide insight into the properties and features of the equilibrium solution concept under consideration.

To determine the NE solution in strictly convex quadratic games, we differentiate J_i with respect to x_i ($i \in \mathbb{N}$), set the resulting expressions equal to zero, and solve the resulting set of equations. This set of equations which provide a sufficient condition because of strict convexity, takes the form:

$$R_{ii}^i x_i + \sum_{j \neq i} R_{ij}^i x_j + r_i^i = 0 \ \ (i \in \mathbb{N})$$

which can be written in compact form as:

$$Ru = -r \tag{3.7}$$

where:

$$R \triangleq \begin{bmatrix} R_{11}^{1} & R_{12}^{1} & \cdots & R_{1N}^{1} \\ R_{11}^{1} & R_{11}^{1} & \cdots & R_{N2}^{2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{11}^{1} & R_{11}^{1} & \cdots & R_{NN}^{N} \end{bmatrix} \tag{3.8}$$

$$x' \triangleq (x_1, x_2, \ldots, x_N)$$

$$r' \triangleq (r_1^1, r_2^2, \ldots, r_N^N)$$

This then leads to the following proposition:

Proposition: The quadratic N-Agent nonzero-sum static game defined by the cost function class (3.10) and with $R_{ii}^{i} > 0$ (a strictly positive diagonal matrix), admits a Nash equilibrium solution, if and only if, (3.7) admits a solution x^*. This Nash solution is unique if the matrix R defined by (3.8) is invertible, in which case it is given by:

$$x^* = -R^{-1}r \tag{3.9}$$

Since each agent's cost function is strictly convex and continuous, quadratic non-zero sum games cannot admit a Nash equilibrium in mixed strategies. Hence, in strictly convex quadratic games, the equilibrium analysis can be confined to the class of pure strategies. This results suits our choice of game for the framework.

□

3.7.6.1 Example Continuation

The first-order function from (3.4):

$$x_i^* = \frac{1}{N-1}\sum_{j \neq i}^{N} x_j$$

can be expressed in the form:

$$\mathbf{x}^* = \mathbf{A}\mathbf{x}^* + \mathbf{c}$$

We have the matrix:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{N-1} & \cdots & \frac{1}{N-1} \\ \frac{1}{N-1} & 0 & \cdots & \frac{1}{N-1} \\ \vdots & \vdots & \ddots & \frac{1}{N-1} \\ \frac{1}{N-1} & \frac{1}{N-1} & \frac{1}{N-1} & 0 \end{bmatrix}$$

The solution is:

$$\mathbf{x}^* = (\mathbf{I} - \mathbf{A})^{-1}$$

where $\mathbf{I}$ is the identity matrix and $(\cdot)^{-1}$ is matrix inversion operation and so:

$$\mathbf{I} - \mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & \frac{1}{N-1} & \cdots & \frac{1}{N-1} \\ \frac{1}{N-1} & 0 & \cdots & \frac{1}{N-1} \\ \vdots & \vdots & \ddots & \frac{1}{N-1} \\ \frac{1}{N-1} & \frac{1}{N-1} & \frac{1}{N-1} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \frac{-1}{N-1} & \cdots & \frac{-1}{N-1} \\ \frac{-1}{N-1} & 1 & \cdots & \frac{-1}{N-1} \\ \vdots & \vdots & \ddots & \frac{-1}{N-1} \\ \frac{-1}{N-1} & \frac{-1}{N-1} & \frac{-1}{N-1} & 1 \end{bmatrix}$$

Since matrix $\mathbf{I} - \mathbf{A}$ has $|a_{ii}| > \sum_{j \neq i} |a_{ij}| \ \forall\, i$, it is strictly diagonally dominant and it is therefore non-singular and invertible by Levy–Desplanques theorem (Horn and Johnson 1990) (equivalent to the Gerschgorin Circle Theorem (Olver 2008)), and the determinant of $\mathbf{A}$ is not equal to zero, $det(\boldsymbol{A}) \neq 0$.

Theorem (Levy–Desplanques): A strictly diagonally dominant matrix is non-singular. In other words, let $\mathbf{A} \in \mathbb{C}^{n,n}$ be a matrix satisfying the property (Taussky 1949) (Schneider 1977):

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \ \forall \ i$$

Moreover, $\mathbf{I} - \mathbf{A}$ is full rank and therefore the linear System always admits a unique NE solution.

□

3.7.6.2 Environmental Factors

We can extend the previous case (3.4) to include Environmental Factors that yields a partial result similar to Alpcan, Rencik et al. (2010):

$$f(x_i, \mathbf{x_{-i}}) = \frac{1}{2}\left(\frac{1}{20(N-1)}\left(x_i - e_{i1}\frac{1}{N-1}\sum_{j \neq i}^{N} x_j\right)\right)^2 \tag{3.10}$$

In this case, we have a single dimensional Environmental Factor space with one element. If we were to enforce the symmetry of Environmental Factors, then e_{i1} would equal exactly one and we would have the function (3.4) as before. The Environmental Factor controls the influence of the averaging component of the Trust Function. A System administrator might want to control this influence when the number of nodes in the System is small so that a node's initial Trust Values are largely preserved from large swings of opinion swayed by relatively few extreme views.

Using the same approach as we took for (3.4), using similar substitutions:

$$A = \frac{1}{20(N-1)} \qquad\qquad B = \frac{1}{N-1}\sum_{j \neq i}^{N} x_j = \bar{\mathbf{x}}_{\mathbf{-i}}$$

we have:

$$f(x_i, \mathbf{x_{-i}}) = \frac{1}{2}\big(A(x_i - e_{i1}B)\big)^2$$

$$= \frac{1}{2}A^2(x_i - e_{i1}B)^2 = \frac{1}{2}A^2(x_i^2 - 2e_{i1}Bx_i + e_{i1}^2B^2)$$

Then:

$$\frac{\partial f}{\partial x_i} = A^2(x_i - e_{i1}B)$$

with a the second derivative condition as before:

$$\frac{\partial^2 f}{\partial x_i^2} = A^2 = \left(\frac{1}{20(N-1)}\right)^2 > 0$$

since $N \in \mathbb{N} > 0$, hence this function is a strictly convex in x_i and emits a uniquely global minimal solution x_i^* at:

$$\frac{\partial f}{\partial x_i} = A^2(x_i - e_{i1}B) = 0$$

$$\Rightarrow x_i^* = e_{i1}B = \frac{e_{i1}}{N-1}\sum_{j\neq i}^{N} x_j$$

This function can again be expressed in the form:

$$\mathbf{x}^* = \mathbf{A}\mathbf{x}^* + \mathbf{c}$$

We have the matrix $\mathbf{A}$ with the additional Environmental Factors incorporated:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{e_{11}}{N-1} & \cdots & \frac{e_{11}}{N-1} \\ \frac{e_{21}}{N-1} & 0 & \cdots & \frac{e_{21}}{N-1} \\ \vdots & \vdots & \ddots & \frac{e_{i1}}{N-1} \\ \frac{e_{N1}}{N-1} & \frac{e_{N1}}{N-1} & \frac{e_{N1}}{N-1} & 0 \end{bmatrix}$$

The solution is:

$$\mathbf{x}^* = (\mathbf{I} - \mathbf{A})^{-1}$$

and:

$$\mathbf{I} - \mathbf{A} = \begin{bmatrix} 1 & \frac{-e_{11}}{N-1} & \cdots & \frac{-e_{11}}{N-1} \\ \frac{-e_{21}}{N-1} & 1 & \cdots & \frac{-e_{21}}{N-1} \\ \vdots & \vdots & \ddots & \frac{-e_{i1}}{N-1} \\ \frac{-e_{N1}}{N-1} & \frac{-e_{N1}}{N-1} & \frac{-e_{N1}}{N-1} & 1 \end{bmatrix}$$

By the same reasoning as before, $\mathbf{I} - \mathbf{A}$ is again full rank, strictly diagonally dominant and it is therefore non-singular and invertible, hence the linear System always admits a unique NE solution.

□

3.7.7 Summary

In this section we, described Game Theory and its role within the framework, specifically, its use for establishing final Trust Values optimised as the solution to a game.

To do this, we identified game theoretical constructs for the nomenclature we have established for the framework previously: game, agents, players or nodes, cost or objective function, decision space and most importantly, Nash equilibrium.

From the definitions of types of games, we identified the type best suited to the framework, and fulfilled the criteria for the formulation of a non-cooperative game. The non-cooperative game was identified as:

1. non-zero sum;
2. continuous-kernel;
3. static;
4. deterministic or pure strategy;
5. multi-act;
6. complete information, and;
7. quadratic (and convex in the cost function).

We provided a formal definition of Nash equilibria, their existence and their uniqueness for convex sets and functions.

We went on to describe the general matrix form of a quadratic game under strict convexity. Hence, in strictly convex quadratic games, the equilibrium analysis can be confined to the class of pure strategies. These are key results that ensure that we are able to determine final Trust Values for reputation and trust profiles, as long as our Trust Functions exhibit the necessary characteristics and we constrain the trust game type suitably.

The section concludes with the continuation of a previous example that makes use of these results, and establishes the form that the Trust Functions will take in the framework.

Significantly, we have provided proof of the suitability of rigorous applications of non-cooperative game theoretical techniques to establish stability and equilibrium applied to the foundational mathematical constructs of the trust framework.

In the next section, we are concerned primarily with the game properties *multi-act* and *complete information* to determine a suitable approach to establishing final Trust Values (Nash equilibria) by iterative methods.

We consider iterative (multi-act) solution methods to quadratic class cost functions, review their relative merits and determine their suitability for the framework.

We establish the concept of stability formally and informally, and identify a suitable *readjustment scheme* for Emerging Systems to cope with equilibrium instability from the complete information available – when Trust Values are altered during the process of establishing their final values.

We identify algorithms suitable for carrying out the iterative analysis and determine the class best suited to our framework.

3.8 Iterative Computation for Trust Spaces

3.8.1 Introduction

In the previous section, we established some suitable game theoretical concepts for the framework. In particular, we rigorously interrogated the existence and uniqueness of NE for the quadratic class of cost functions.

In this section, we consider how we can compute NE (and final Trust Values) for cost functions with suitable iterative methods, their algorithmic implementation, and the stability of these solutions.

The computation method for the framework must be able to support the distributed nature of an Emerging System and under rapidly fluctuating conditions. There also needs to be consideration for the potential very large size of Emerging Systems.

The approaches make use of the fact that the game class we have identified is multi-act corresponding to the iterations of the computation and full information, that is, that all nodes are privy to the same information in the System.

An iterative method is a mathematical procedure that generates a sequence of improving approximate solutions for a class of problems.

We consider classical stationary iterative methods - Jacobi and Gauss-Seidel – the acceleration of these methods by Successive Relaxation and briefly acknowledge more modern methods such as Krylov Subspaces.

We also consider the main methods in terms of ease of computation, convergence rate and factors that assure convergence.

This section is not intended to be an exhaustive exploration of iterative methods (see Reich (1949), Gupta (1995), Balagurusamy (1999), Saad (2003), Mathews and Fink (2006) or Parnell (2013), but sufficient to identify suitable computation approaches for the framework.

3.8.2 Iterative Methods

Classical stationary iterative methods for solutions to Systems of linear equations are Jacobi (Saad 2003) and Gauss-Seidel (Reich 1949) methods (Gupta 1995) (Balagurusamy 1999). These methods can be adapted using successive reductive techniques to establish convergence in some cases or increase the rate.

These iterative methods converge to a solution to Systems of linear equations of the form:

$$\mathbf{Ax} = \mathbf{b}$$

when the iteration matrix, $\mathbf{A}$ exhibits certain characteristics.

These methods will be used by the framework to iteratively establish final Trust Values for trust spaces. They will provide a numeric solution to the optimal minimisation problem for the cost functions of games.

If we are able to prove that the cost function (first derivative) matrix form exhibits these characteristics, we can ensure that an iterative method will converge to a solution. This solution, is the NE for the game and equivalently, the final Trust Value.

3.8.2.1 Jacobi

In the Jacobi case, $\mathbf{A}$ can be decomposed into a diagonal component matrix $\mathbf{D}$ and the remainder matrix component $\mathbf{R}$:

$$\mathbf{A} = \mathbf{D} + \mathbf{R}$$

The solution is then obtained iteratively by:

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x}^{(k)}) \tag{3.11}$$

The element-based formulation is then:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij}\, x_j^{(k)}\right), i = 1,2, \ldots, n$$

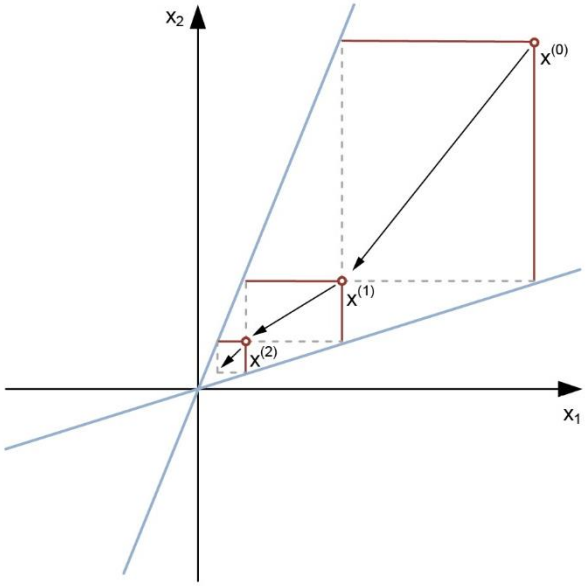


Figure 11 Convergence of Jacobi method (Bertsekas and Tsitsiklis 1997)

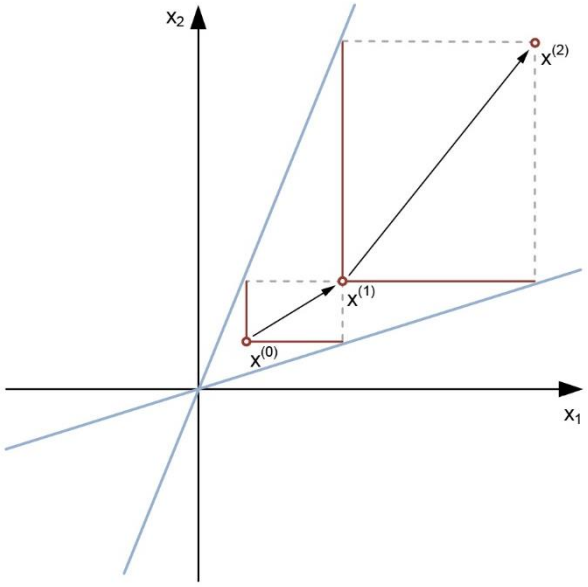


Figure 12 Divergence of Jacobi method (Bertsekas and Tsitsiklis 1997)

The computation of $x_i^{(k+1)}$ requires every element in $\mathbf{x}^{(k)}$ except $x_i^{(k)}$. This could be represented as $\mathbf{x}_{-\boldsymbol{i}}^{(k)}$ as we have seen previously.

Unlike the Gauss–Seidel method, $x_i^{(k)}$ is not overwritten with $x_i^{(k+1)}$, as that value will be needed for the rest of the computation. The minimum amount of storage is two vectors of size n.

3.8.2.2 Gauss-Seidel

To produce a faster iterative method Gauss-Seidel amends the Jacobi Method to make use of new values as they become available (Parnell 2013).

For Gauss-Seidel, $\mathbf{A}$ can be decomposed into a lower triangular component matrix $\mathbf{L}_*$ and a strictly upper triangular matrix component $\mathbf{U}$:

$$\mathbf{A} = \mathbf{L}_* + \mathbf{U}$$

The System of linear equations becomes:

$$\mathbf{L}_*\mathbf{x} = \mathbf{b} - \mathbf{U}\mathbf{x}$$

Analytically, the Gauss-Seidel iteration can be written as:

$$\mathbf{x}^{(k+1)} = \mathbf{L}_*^{-1}(\mathbf{b} - \mathbf{U}\mathbf{x}^{(k)}) \tag{3.12}$$

By the triangular form of $\mathbf{L}_*$, the elements of $\mathbf{x}^{(k+1)}$ can be computed sequentially using forward substitution:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}\, x_j^{(k+1)} - \sum_{j>i} a_{ij}\, x_j^{(k)}\right), i = 1,2,\dots,n$$

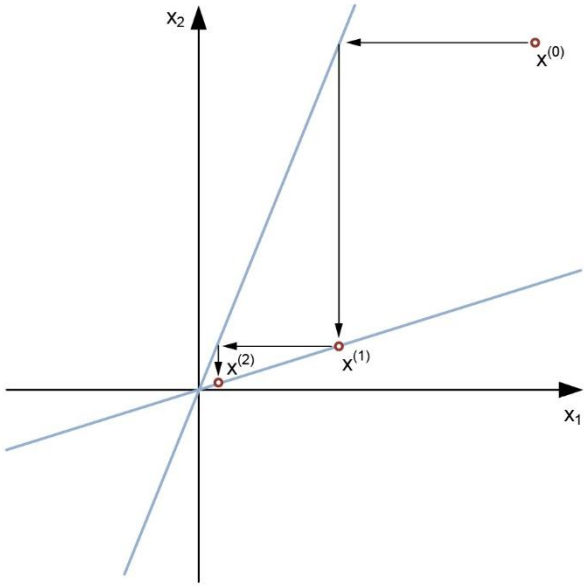


Figure 13 Convergence of Gauss-Seidel method (Bertsekas and Tsitsiklis 1997)

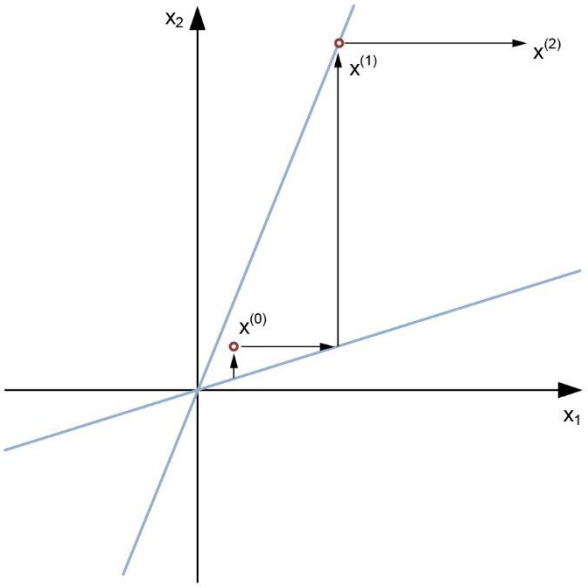


Figure 14 Divergence of Gauss-Seidel method (Bertsekas and Tsitsiklis 1997)

The computation of $x_i^{(k+1)}$ requires every element in $\mathbf{x}^{(k)}$ except $x_i^{(k)}$. This could be represented as $\mathbf{x}_{-\boldsymbol{i}}^{(k)}$ as we have seen previously.

The procedure is continued until the changes made between iterations breach some tolerance. The amount of storage required is one vector of size n.

3.8.2.3 Successive Relaxation

Successive relaxation methods can be considered as the acceleration of the convergence rate of Gauss-Seidel method. Jacobi methods can also be relaxed in this fashion.

For this derivation, we take as a variation on the definitions of $\mathbf{L}$ and $\mathbf{U}$ previously:

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$$

and make use of a *relaxation parameter* ω to reduce the a residual variable $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$ as much as possible for each approximation to the solution $\mathbf{x}^{(k)}$.

The recurrence relation is given by (Parnell 2013):

$$\mathbf{x}^{(k+1)} = (\mathbf{D} - \omega\mathbf{L})^{-1}((1-\omega)\mathbf{D} - \omega\mathbf{U})\mathbf{x}^{(k)} + (\mathbf{D} - \omega\mathbf{L})^{-1}\omega\mathbf{b} \tag{3.13}$$

This process of reducing residuals at each stage is successive relaxation.

- If $0 < \omega < 1$, the method is successive *under* relaxation and can be used to obtain convergence when the Gauss-Seidel method is not convergent;
- For $\omega > 1$, the method is successive *over* relaxation and can be used to accelerate the Gauss-Seidel method's convergence rate;
- With $\omega = 1$, we simply have the Gauss-Seidel method.

The Successive Over Relaxation (SOR) method ($\omega > 1$) is given by:

$$\mathbf{B}_{\mathrm{SOR}} = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1-\omega)\mathbf{D} + \omega\mathbf{U}] \tag{3.14}$$

The iteration matrix $\mathbf{B}_{\mathrm{SOR}}$ is derived by splitting $\mathbf{A}$:

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U} = \mathbf{D}\left(1 - \frac{1}{\omega}\right) + \frac{1}{\omega}\mathbf{D} - \mathbf{L} - \mathbf{U}$$

Then $\mathbf{A}\mathbf{x} = \mathbf{b}$ is:

$$\left(\frac{1}{\omega}\mathbf{D} - \mathbf{L}\right)\mathbf{x} = \left(-\left(1 - \frac{1}{\omega}\right)\mathbf{D} + \mathbf{U}\right)\mathbf{x} + \mathbf{b}$$

$$(\mathbf{D} - \omega\mathbf{L})\mathbf{x} = \big((1-\omega)\mathbf{D} + \omega\mathbf{U}\big)\mathbf{x} + \omega\mathbf{b}$$

which yields (3.14) as required.

To maximise the rate of convergence and equivalently, minimise the spectral radius $\rho(\mathbf{B}_{\mathrm{SOR}})$ (0) a well suited ω should be identified. While there is no complete method to do this, for $1 \leq \mathrm{i} \leq N$ with $a_{ii} \neq 0$, then:

$$\rho(\mathbf{B}_{\mathrm{SOR}}(\omega)) \geq |1 - \omega|$$

Convergence then, is achieved when $0 < \omega < 2$. Identifying a suitable ω can improve convergence significantly. For large Systems of equations, Gauss-Seidel method can be slow but improved substantially by successive relaxation given an optimal ω.

In general, a ω is chosen $1 < \omega < 2$ as it tends to yield a small spectral radius. Some matrices exhibit characteristics conducive to identifying a suitable ω relatively easily. Consider the linear System $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ again. If the eigenvalues (Strang 2003) of:

$$\left(\alpha \mathbf{D}^{-1}\mathbf{L} + \frac{1}{\alpha}\mathbf{D}^{-1}\mathbf{U}\right), \alpha \neq 0$$

are independent of α, the matrix is *consistently ordered* and the optimal ω for the SOR iteration method is:

$$\omega = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B}_J)}}$$

where $\mathbf{B}_J$ is the Jacobi method iteration matrix, $\mathbf{B}_J = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$ (Parnell 2013)**.**

3.8.2.4 Krylov Subspaces

Krylov subspace methods work by forming a basis of the sequence of successive matrix powers times the initial residual (the Krylov sequence). The approximations to the solution are then formed by minimising the residual in the subspace formed.

The order-r Krylov subspace generated by an n-by- n matrix $\mathbf{A}$ and a vector $\mathbf{b}$ of dimension n is the linear subspace spanned (Strang 2003) by the images of $\mathbf{b}$ under the first r powers of $\mathbf{A}$ starting from $\mathbf{A}^0 = \mathbf{I}$:

$$\mathcal{K}_r(\mathbf{A}, \mathbf{b}) = \mathrm{span}\{\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \dots, \mathbf{A}^{r-1}\mathbf{b}\}$$

The basis (Strang 2003) for the Krylov subspace (Strang 2003) that the inverse of a matrix can be found in terms of a linear combination of its powers.

Modern iterative methods for finding one or many eigenvalues large Systems of linear equations avoid matrix-matrix operations, but rather multiply vectors by the matrix and work with the resulting vectors.

From the vector $\mathbf{b}$, $\mathbf{Ab}$ is computed. Further multiplicative iterations of this follow:

$$\mathbf{A}^2\mathbf{b}, \mathbf{A}^i\mathbf{b}, \dots, \mathbf{A}^{r-1}\mathbf{b}$$

All methods that progress this way are of the class of Krylov subspaces (Saad 2003).

Krylov subspace methods include Arnoldi, Lanczos, Conjugate gradient, GMRES (Generalized Minimum RESidual), BiCGSTAB (BiConjugate Gradient STABilized), QMR (Quasi Minimal Residual), TFQMR (Transpose-Free QMR), and MINRES (MINimal RESidual).

Arnoldi and Lanczos are iterative methods for the numerical calculation of eigenvalues of general matrices and could be used in conjunction with SOR for instance, to establish a suitable ω for optimal convergence.

GMRES and BiCGSTAB are iterative methods for the solution to non-symmetric linear Systems. The Arnoldi method is used to identify the approximation vector of the solution for GMRES (Saad 2003).

QMR is an iterative residual method for large, square and sparse matrix (a matrix comprising predominately of zeros) Systems of linear equations. TFQMR and MINRES are developed for sparse matrices, particularly applied to optimisations problems (Saad 2003). Sparse matrices are typical for partial differential equation problems within science and engineering. Most large matrices that arise in the analysis of the physical sciences are sparse and the recognition of this fact makes the solution to linear Systems of millions of coefficients feasible (John R. Gilbert 1992) (Lindfield and Penny 2012).

3.8.2.5 Convergence

The following table describes the necessary conditions on $\mathbf{A}$ for the stationary iterative methods to converge:

Conditions on $\mathbf{A}$	Jacobi	Gauss-Seidel	Successive Relaxation
Symmetric positive definite	If $\mathbf{2D} - \mathbf{A}$ is positive definite (Young 2003)	Yes (Young 2003)	If $0 < \omega < 2$ (Young 2003)
Irreducible diagonally and dominant with: $\mathbf{D} > 0$	Yes (Young 2003)	Yes (Young 2003)	If $0 < \omega \leq 1$ (Young 2003)
Real, symmetric and non-singular	Iff $\mathbf{A}$ and $\mathbf{2D} - \mathbf{A}$ is positive definite (Young 2003)	Iff $\mathbf{A}$ is positive definite (Young 2003)	Iff $\mathbf{A}$ is positive definite and $0 < \omega < 2$ (Young 2003)
$\mathbf{L}$-matrix	Iff $\mathbf{M}$ matrix (Young 2003)	Iff $\mathbf{M}$ matrix (Young 2003)	Iff $\mathbf{A}$ is positive definite and $0 < \omega \leq 1$ (Young 2003)
Consistently ordered and symmetric with: $\mathbf{D} > 0$	Iff $\mathbf{A}$ is positive definite (Young 2003)	If Jacobi does (Ortega 1990)	If Jacobi does and $0 < \omega < 2$ (Ortega 1990)
Strictly diagonally dominant	Yes (Varga 2000)	Yes (Varga 2000)	If $\omega = 1$ then we have Gauss-Seidel

Table 4 Necessary conditions on $\boldsymbol{A}$ for the stationary iterative methods to converge

Such that a matrix $\mathbf{A}$ is an $\mathbf{L}$-matrix if $a_{ii} > 0$ for all i and $a_{ij} \leq 0$ for $i \neq j$ and an $\mathbf{M}$-matrix is a real non-singular $\mathbf{L}$-matrix with $\mathbf{A}^{-1} \geq 0$ (element-wise).

The framework makes use of cost functions which yield Systems of linear equations of the form where the iteration matrix exhibits these characteristics, so we can assure convergence to a NE final Trust Value through computation.

□

3.8.2.5.1 Spectral Radius

Suppose the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ converges to $\mathbf{x}$, where:

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{c}$$

is a System of linear equations in iterative form. If $\mathbf{x}^{(k)} \to \mathbf{x}$ for $k \to \infty$, then $\mathbf{x}$ satisfies the equation:

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{c}$$

and so we have:

$$\mathbf{x}^{(k+1)} - \mathbf{x} = \mathbf{B}(\mathbf{x}^{(k)} - \mathbf{x})$$

Consider the situation where $\mathbf{B}_{N\times N}$ has N linearly independent eigenvectors (Strang 2003). We can substitute $\mathbf{v}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$**,** we have:

$$\mathbf{v}^{(k+1)} = \mathbf{B}\mathbf{v}^{(k)}$$

With $\mathbf{v}^{(0)} = \sum_{i=1}^{N} \alpha_i \mathbf{e}_i$ where $\mathbf{e}_i$ are the eigenvectors with associated eigenvalues λ_i of $\mathbf{B}$, continuing the sequence gives:

$$\mathbf{v}^{(k)} = \sum_{i=1}^{N} \alpha_i \lambda_i^k \mathbf{e}_i$$

Suppose $|\lambda_1| > |\lambda_i|$ $(i = 1,2, \dots, N)$, then:

$$\mathbf{v}^{(k)} = \alpha_1 \lambda_1^k \mathbf{e}_1 + \sum_{i=2}^{N} \alpha_i \lambda_i^k \mathbf{e}_i$$

$$= \lambda_1^k \left[\alpha_1 \mathbf{e}_1 + \sum_{i=2}^{N} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \mathbf{e}_i \right]$$

Given that $\frac{\lambda_i}{\lambda_1} < 1$, for large k:

$$\mathbf{v}^{(k)} \simeq \alpha_i \lambda_1^k \mathbf{e}_1$$

Hence, the error associated with $\mathbf{x}^{(k)}$, the k-th vector in the sequence, is given by $\mathbf{v}^{(k)}$ which varies as the k-th power of the largest eigenvalue. That is, it varies with the k-th power of the spectral radius $\rho(\mathbf{B}) = |\lambda_1|$. So the spectral radius is a good indication of the rate of convergence for an iterative method (Parnell 2013).

While this result is useful when comparing the rate of convergence for iterative methods, it requires the magnitude of all eigenvalues to be known. This can be computationally expensive and difficult to do in practice, requiring some numerical iterative method itself.

Theorem (Gerschgorin's Circle):

Seen previously in (3.7.6.1), Gerschgorin's Theorem, allows us to bound eigenvalues without having to actually find them.

Consider:

$$\mathbf{Ae} = \lambda\mathbf{e}$$

where λ and $\mathbf{e}$ are the eigenvalue, eigenvector pair of the matrix $\mathbf{A}$ (Strang 2003). In component form, we have:

$$\sum_{j=1}^{N} a_{ij}e_j = a_{ii}e_i + \sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}e_j = \lambda e_i$$

$$\Rightarrow e_i(a_{ii} - \lambda) = -\sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}e_j$$

$$\Rightarrow |e_i||a_{ii} - \lambda| \leq \sum_{\substack{j=1 \\ j\neq i}}^{N} |a_{ij}||e_j|$$

with $e_j \neq 0 \; \forall \; j$. If $|e_l|$ is a the magnitude of the largest component of $\mathbf{e}$, such that $|e_l| \geq |e_j| \; \forall \; j$, then:

$$|a_{ll} - \lambda| \leq \sum_{\substack{j=1 \\ j\neq l}}^{N} |a_{lj}| \tag{3.15}$$

Each eigenvalue lies within a circle with centre a_{ll} and radius $\sum_{j=1}^{N}|a_{lj}|, \; j \neq l$. Without knowing λ and $\mathbf{e}$, we cannot know l, however we can conclude that the union of all the circles must contain all the eigenvalues and we have an upper bound. The smaller the bound, the smaller the spectral radius and hence, faster the rate of convergence for the iterative method.

3.8.2.6 Discussion

3.8.2.6.1 Nash Equilibrium Algorithms

For completeness, we consider algorithms specifically designed for computation of Nash Equilibria.

Von Stengel (2002) and McKelvey and McLennan (1996) provide comprehensive surveys in the literature with more a more detailed description of the references here.

Commonly used for the calculation of Nash equilibrium (Porter, Nudelman et al. 2008) in a two-player game is the Lemke-Howson algorithm (Lemke and Howson 1964) which is a specialised case of Lemke's method (Lemke 1965). The algorithm uses a "pivoting" algorithm that establishes an arbitrary initial first pivot from which, successive pivots lead to an equilibrium solution. Each action of the first player can be thought of as defining a path from the starting point to the Nash equilibrium.

Gambit (McKelvey, McLennan et al. 1995) is an open-source collection of tools for carrying out computation in game theory. Game models can be built, explored and analysed. The Lemke-Howson algorithm is implemented in in Gambit where the initial pivot is selected.

For n-player games, until recently, "simplicial subdivison" Van der Laan, Talman et al. (1987) and variants were common. The approach approximates a fixed point of a cost function which is defined on a simplotope (Doup 1988). The approximation is achieved by triangulating the simplotope with a mesh of some granularity and traversing the triangulation along a fixed path.

More recently, Govindan and Wilson (2003) introduced a continuation method for determining Nash equilibrium. The approach first perturbs a game to one that has a known equilibrium and then traces the solution back to the original game as the magnitude of the perturbation approaches zero. The structure theorem by E. Kohlberg (1986) guarantees that the game and the solution can be traced simultaneously. This method has been implemented by He, Huang et al. (2003) and extended to solve "graphical games" and "mulit-agent influence diagrams" by Koller and Milch (2003).

Similar in approach is Dickhaut and Kaplan (1991) and Porter, Nudelman et al. (2008). Both use the enumeration of "supports" and the solution to a feasibility program to determine a Nash equilibrium. This approach was suggested earlier by (Mangasarian and Stone 1964) based on the enumeration of vertices of a polytope. For their experiments Porter, Nudelman et al. (2008), use Gambit (McKelvey, McLennan et al. 1995) and GAMUT (Nudelman, Wortman et al. 2004) tools as their test-bed.

Daskalakis, Goldberg et al. (2009) and (Chen and Deng 2006) discuss the complexity of computing a Nash equilibrium and how an algorithm can be constructed that runs in polynomial time, specifically on a complexity class, “Polynomial Parity Arguments on Directed” graphs (PPAD) – a subclass of “Total Function Nondeterministic Polynomial” (TFNP) - introduced by (Papadimitriou 2003). The problem class contains Nash equilibrium problems. Daskalakis, Goldberg et al. (2009) show that for this class of problem, three-player games are complete and that the (Chen and Deng 2006) result for two-player games follows from the proof.

In this work, we are most interested in the use of a similar algorithm for our experimental analysis to assure consistency of results. We are not so interested in the efficiency of the algorithm just that in basic terms, it can be test to ensure it supports the characteristics of an Emerging System.

3.8.2.6.2 Trust Equilibrium Iterative Methods

Iterative methods will transition initial trust and Reputation Profiles within the trust space, to an optimal equilibrium state in accordance to the solution to a game theoretical cost function and under the influence of Environmental Factors.

a_i	Iteration complete.
a_i	Update request made.
a_i	Update response received.
a_i	Unstable node.

Figure 15 Node states for iterative methods scheme

With respect to this scheme, the initial state of the System with an initial Reputation Profile takes the form:

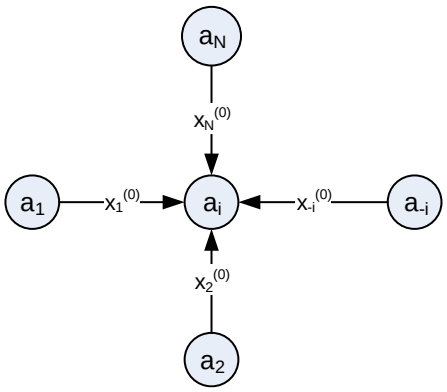


Figure 16 Initial state Reputation Profile node

The final state at equilibrium is:

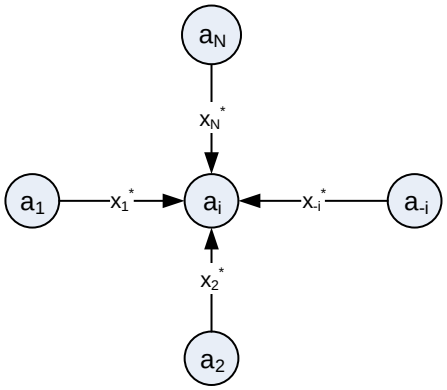


Figure 17 Final state equilibrium Reputation Profile

The element-wise formulation for the Gauss–Seidel method is extremely similar to that of the Jacobi method. The computation of $x_i^{(k+1)}$ uses only the elements of $\mathbf{x}^{(k+1)}$ that have already been computed, and only the elements of $\mathbf{x}^{(k)}$ that have not yet to be advanced to iteration $k+1$. This means that, unlike the Jacobi method, only one storage vector is required as elements can be overwritten as they are computed, which can be advantageous for very large problems (Saad 2003).

However, unlike the Jacobi method, the computations for each element cannot be done in parallel. Furthermore, the values at each iteration are dependent on the order of the original equations which correspond to an order of nodes in the System (Saad 2003).

For the Jacobi method:

Initial requests for update are made by all nodes in parallel after the initial state.

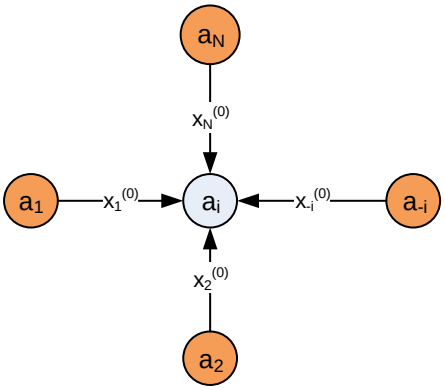


Figure 18 Jacobi initial requests

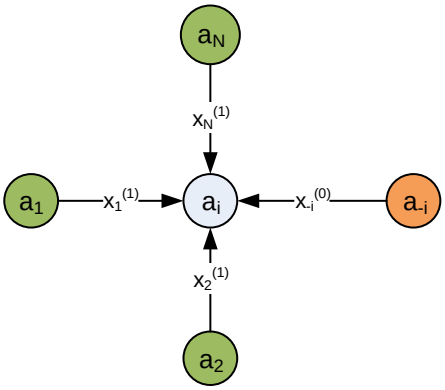


Figure 19 Jacobi responses received

Optimal final values are established for each node based on the previously completed iteration or from initial values.

This can be accomplished without the need for every node to have responded to an update request.

All final optimal values are established and the subsequent iteration can commence or the final equilibrium state is established.

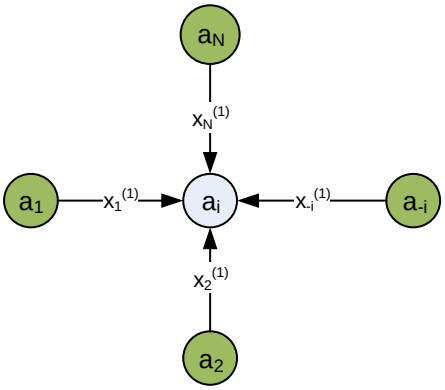


Figure 20 Jacobi iteration optimal values

For the Gauss–Seidel method:

Sequential requests are made for updates.

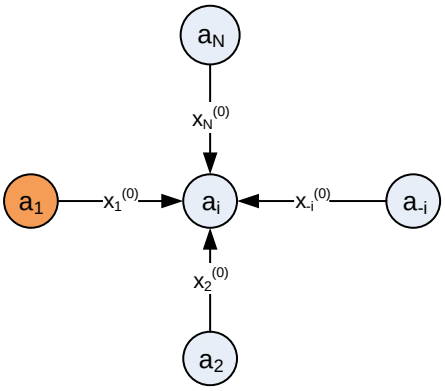


Figure 21 Gauss–Seidel initial request

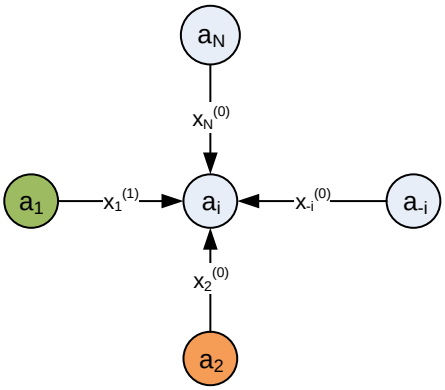


Figure 22 Gauss–Seidel initial response

Responses are received in a similar order.

Subsequent nodes incorporate previous update responses of the current iteration to establish their optimal Trust Values.

Consequent iterations follow with the same dependencies.

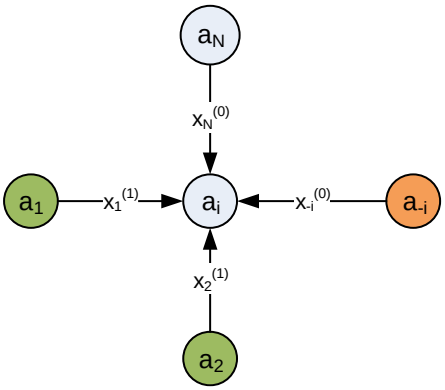
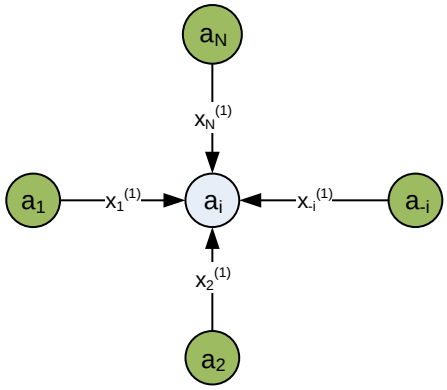


Figure 23 Gauss–Seidel Consequent requests and responses



All final optimal values are established and the subsequent iteration can commence or the final equilibrium state is established.

Figure 24 Gauss–Seidel iteration optimal values

Successive relaxation methods can accelerate the convergence rates of the Jacobi and Gauss–Seidel methods as long as a suitable relaxation parameter can be determined. Determining a suitable parameter incurs an additional computational overhead and is only optimally certain in a small number of cases and is extremely sensitive.

For the framework, while computational sluggishness is undesirable, the parallel nature of the Jacobi method is more appropriate for Emerging Systems prone to volatility in their topology since sources of data can be sought concurrently and alternative sources of data can be sought should the initial request prove fruitless. It cannot be ignored however, that Emerging Systems such as the Internet pose very large problems. In practice, Gauss–Seidel though a magnitude faster than Jacobi method, is not a viable option based on rate of convergence.

Modern methods based on Krylov subspaces offer approaches for numerically determining eigenvalues suitable for accelerating convergence of classical methods by successive relaxation and solutions to linear Systems. Most have specialist applications to sets of System characteristics (non-symmetric and sparse in particular) but provide accelerated convergence by avoiding computationally expensive matrix multiplication.

The focus of this is work is non-cooperative Emerging Systems. The most pertinent quality of the iterative method a framework needs to suit these types of Emerging Systems, is parallelism. Without it, the framework will not corroborate the distributed, fault-tolerant characteristics of Emerging Systems (Kovacs, Robrie et al. 2006). Further, there can be no assumption that sparse associative

matrices will be prevalent when finding equilibrium solutions as we hold that a complete opinion and reputation must be established within the System (3.2.3).

In the case where nodes become unresponsive either temporarily or permanently, the framework must be able to continue to establish suitable equilibrium Trust Values.

A tolerance for a response to a request for update is established where any breach of which, eliminates the node as an active member of the System or establishes results based on previous responses ($l > k$).

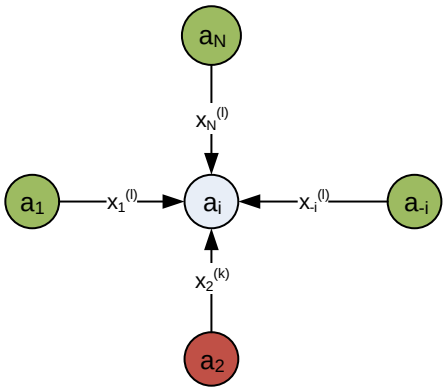


Figure 25 Unstable or rogue node

Co-operation between nodes cannot be assumed as has been established by the nature of the Emerging System to which the framework is to be applied. Contingencies have to be present in the application of an iterative method for when a node does not respond at all. This could be permanent or temporary but must be within a tolerance bound suited to the application of the framework.

This capacity to cope with deviations from optimal behaviour is *stability* and is the ability of the iterative method to cater for anomalistic and volatile node behaviour.

□

3.8.3 Stability

Anecdotally, the notion of stability of a NE solution can be elucidated through the following scenario (Başar and Olsder 1999). Given a NE solution, consider the following sequence of events in a simple two-node trust game System:

1. one of the nodes a_1 deviates from its equilibrium solution;
2. a second node a_2 observes the deviation and minimises its cost function in view of the new strategy adopted by a_1;
3. a_1 now optimally reacts to a_2, and;
4. the process continues ad infinitum.

If this infinite sequence of reactions converges to the original NE solution regardless of a deviation from optimality, the NE solution is deemed *stable*. If convergence is attained only under small

deviation, then the NE is considered *locally stable*. Otherwise, the solution is *unstable* and it diverges.

This scenario occurs as the result of a change in the topology of the Emerging System, a change of Environmental Variables or an erroneous error. The first circumstance could occur due to a break in communication resulting in partitioning of the System, the second could be an administrator's response to a security threat or any modification of the terms of engagement for the nodes, and the third could be an application fault or node resource limitation. The case could equally be a reaction to an error, as a response to administrative changes to the System.

Stability is a refinement of the NE concept. We have to ensure that the equilibrium is "restorable" (Başar and Olsder 1999) under any rational adjustment scheme when there is a deviation from it by any node.

This characteristic is key to ensuring the robustness of our Emerging System. If a cost function's stability can be established, then we are able to ensure a level of fault tolerance as partial fulfilment of the criteria for an Emerging System (Brewer 2000).

Formally then:

Definition: A Nash equilibrium $x_i^*, i \in \mathbb{N}$ is globally stable with respect to an adjustment scheme $\mathcal{S}$ if it can be obtained as the limit of the iteration:

$$x_i^* = \lim_{k\to\infty} x_i^k$$

$$x_i^{k+1} = \arg\min_{x_i \in X_i} J_i\left(x_{-i}^{(\mathcal{S}_k)}, x_i\right), x_i^0 \in X_i$$

where x_i^k is the k-th iteration of the multi-play game for Trust Value, x_i and $\mathcal{S}_k$ indicates the precise choice of $x_{-i}^{(\mathcal{S}_k)}$ depends on the readjustment scheme selected.

One possibility for the scheme is:

$$x_{-i}^{(\mathcal{S}_k)} = x_{-i}^{(k)}$$

which corresponds to the situation where the nodes readjust their optimisation simultaneously, in response to the most recently determined Trust Values of the other nodes.

Another possibility is:

$$x_{-i}^{(\mathcal{S}_k)} = (x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_{i+1}^{(k+1)}, \ldots, x_N^{(k+1)})$$

where the nodes update in a predetermined order (in this case, numerical).

A more complex case is where the readjustments occur at random with the delay not exceeding d time units:

$$x_{-i}^{(S_k)} = (x_1^{m_{1,k}^i}, \dots, x_{i-1}^{m_{i-1,k}^i}, x_{i+1}^{m_{i+1,k}^i}, \dots, x_N^{m_{N,k}^i})$$

where $m_{j,k}^i$ is an integer-valued random variable, satisfying the bounds:

$$\max(0, k-d) \leq m_{j,k}^i \leq k+1, i \neq j, i \in \mathbb{N}, j \in \mathbb{N}$$

If the iterations converge, then the NE solution is unique. Not all NE solutions are necessarily stable. Stability is established with respect to a declared scheme (Başar and Olsder 1999).

For Emerging Systems, the most frequently disruptive events will occur in communication (Kovacs, Robrie et al. 2006) apparently, at random. A simultaneous readjustment as in the first case is prohibitive since the level of integrity within an Emerging System cannot be assumed to be this high. The second case is more practical within an Emerging System since it reflects the more relaxed iterative nature of the System by not assuming a high level of integrity, but it still maintains a high level of structure not best suited to a rapidly changing topology. The final case is best suited to an Emerging System since it incorporates a suitable tolerance constraint in d and makes no assumptions about the topology of the System. The third case models the nature of an Emerging System closest.

□

3.8.3.1 Quadratic Case

We now consider the quadratic case explicitly since this is the class of game the framework will utilise (Bertsekas and Tsitsiklis 1997).

Proposition: For a two-person non-zero sum game, in addition to result (3.5), assume that J_i is twice differentiable in x_i for each $x_j \in \mathbb{R}^{m_i}$, $i, j = 1,2\ i \neq j$. Then assume that the matrix $C_1 C_2 \equiv C_2 C_1$ has operator norm strictly less than 1. Then the game admits a unique, stable NE.

This result is a direct specialisation for the quadratic case, for proof, see Başar and Olsder (1999). The reaction function general case is specialised to the quadratic form with $N = 2$:

$$x_1^{(k+1)} = C_1 x_2^{(k)} + d_1, x_2^{(k+1)} = C_2 x_1^{(k+1)} + d_2, k = 0,1, \dots$$

From (3.7) with arbitrary starting choice $x_2^{(0)}$, and:

$$C_i = -\left(R_{ii}^i\right)^{-1} R_{ij}^i, d_i = -\left(R_{ii}^i\right)^{-1} r_i^i, j \neq i, i, j = 1,2$$

This iteration corresponds to the sequential Gauss-Seidel update scheme where a_1 responds to the most recent past action of p_2, whereas $a_2$responds to the current action of a_1.

The alternative to this is the parallel Jacobi update scheme:

$$x_1^{(k+1)} = C_1 x_2^{(k)} + d_1, x_2^{(k+1)} = C_2 x_1^{(k)} + d_2, k = 0,1, \ldots$$

After some arbitrary re-indexing, the stability of these iterations is equivalent to the single iteration:

$$x_1^{(k+1)} = C_1 C_2 x_1^{(k)} + C_1 d_2 + d_1$$

A necessary and sufficient condition for this to converge from our initial proposition is that the eigenvalues of the matrix $C_1 C_2$ (and equivalently $C_2 C_1$) should be in the unit circle, that is:

$$\rho(C_1 C_2) \equiv \rho(C_2 C_1) < 1$$

where $\rho(A)$ is the spectral radius of the matrix A. This spectral radius condition is well held for the quadratic case.

The condition for stability is significantly more stringent than the condition for existence of a unique NE, which is:

$$\det(I - C_1 C_2) \neq 0$$

For the framework then, we establish the restorative capability of an iterative method in situations of non-optimal response for quadratic cases. This demonstrates the framework's ability to recover in situations where nodes are unavailable potentially leave the System completely, as can be expected in Emerging Systems.

□

3.8.4 Jacobi OverRelaxation (JOR) Algorithm

In order to consider iterative methods (3.8.2.1) as algorithms, we will transition to the traditional convention of using t as the iteration parameter. The parameter loosely represents time in discrete progression.

Then, the Jacobi algorithm for $t = 1,2, \ldots$, some $x(t)$ and initial vector $x(0)$ becomes (Bertsekas and Tsitsiklis 1997):

$$x_i(t+1) = -\frac{1}{a_{ii}}\left(\sum_{j \neq i} a_{ij}\, x_j(t) - b_i\right) \tag{3.16}$$

Having established the Jacobi method's parallel update approach as preferential for Emerging Systems, we can attempt to enhance the method by increasing the rate of convergence with a Jacobi OverRelaxation (JOR) algorithm. This is a similar approach to that taken for Gauss-Seidel previously (Bazaraa, Sherali et al. 2006) with the form:

$$x_i(t+1) = (1-\omega)x_i(t) - \frac{\omega}{a_{ii}}\left(\sum_{j\neq i} a_{ij}\, x_j(t) - b_i\right) \tag{3.17}$$

and with relaxation parameter ω. In particular, if $0 < \omega < 1$, the new value of x_i obtained from (3.17) is a convex combination of the old values of x_i and the new value x_i that would have been obtained from the Jacobi algorithm without the relaxation parameter (3.16) (Bertsekas and Tsitsiklis 1997). A common choice for the relaxation parameter for this method is $\omega = \frac{2}{3}$ (Saad 2003).

3.8.5 Implementation

We have identified an (classical) iterative method and algorithm that suits the needs of the Non-cooperative Programmable Open System Trust (NPOST) framework for establishing equilibrium solutions to our trust games. Specifically, we identified a Jacobi Overrelaxation algorithm (JOR) that complements a distributed topology well by permitting parallel value updates, makes no sparsity assumptions and we are able to accelerate convergence with a relaxation parameter.

To carry out an experimental analysis of the framework, we require a testable implementation of the iterative algorithm.

Adapted from the Ivos (2013) Matlab implementation of a JOR algorithm, with some modifications to naming and a commitment to the Euclidean norm, we have the following implementation in Matlab (Mathews and Fink 2006) (Lindfield and Penny 2012).

Simplified, the implementation accepts the variables:

Variable	Description
`A`	The matrix $\mathbf{A}$ from the system of equations, $\mathbf{Ax} = \mathbf{b}$.
`b`	The vector $\mathbf{b}$ from the system of equations, $\mathbf{Ax} = \mathbf{b}$.
`x0`	Initial Reputation Profile vector $R_j(a_i)$, $\mathbf{x}^{(0)}$ and $x(0)$.
`w`	JOR relaxation parameter ω.
`e`	Error tolerance or convergence condition. When this condition is met, we consider the solution to the system $\mathbf{Ax} = \mathbf{b}$, the NE solution, $\mathbf{x}^*$, and final Reputation Profile $R_j^*(a_i)$, established.
`maxt`	Restrict the maximum number of iterations of the algorithm to bound divergent cases where the convergence condition is not reached.

and returns:

Variable	Description
`x`	The approximate solution vector $\mathbf{x}$ from the System of equations, $\mathbf{Ax} = \mathbf{b}$. The solution to the system $\mathbf{Ax} = \mathbf{b}$, the NE solution, $\mathbf{x}^*$, and final Reputation Profile $R_j^*(a_i)$.
`t`	The iteration parameter, $x(t)$, where $t = 1,2, \ldots$
`dif`	Euclidean norm of the difference between two iterations.

The algorithm iterates over t from the initial Reputation Profile Trust Values, $R_j(a_i)$, to establish the final Reputation Profile, $R_j^*(a_i)$. The algorithm can be read largely as pseudo-code with elementary computational flow control and assignments, while making use of the following (Matlab) functions (Mathworks 2013):

Function	Description
`bsxfun`	Apply element-by-element binary operation to two arrays with singleton expansion enabled.
`diag`	Diagonal matrices and diagonals of matrix.
`length`	Length of vector or largest array dimension.
`norm`	Vector and matrix norms.
`numel`	Number of array elements.
`ones`	Create array of all ones.
`spdiags`	Extract and create sparse band and diagonal matrices.
`zeros`	Create array of all zeros.

Preliminarily, the algorithm establishes the form of the matrix $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ (3.8.2.3) as the iteration matrix required for the Jacobi method, and:

$$\mathbf{x}(t+1) = (\mathbf{I} - \mathbf{B}^{-1}\mathbf{A})\mathbf{x}(t) + \mathbf{B}^{-1}\mathbf{b}$$

with consideration for the relaxation parameter ω. No assumptions are made about the density of $\mathbf{A}$ despite the extraction of sparse band and diagonal matrices from it:

```
% Create iteration matrix
TJORw = spdiags((1-w)*ones(length(A),1),0,...
    length(A),length(A))+w*bsxfun(@times,1./diag(A),...
    -A+diag(diag(A)));
```

We also have the constant in explicit form, $\frac{\omega b_i}{a_{ii}}$ from (3.17) as a component-wise calculation:

```
% Set constant
cJORw = w*b./diag(A);
```

In the case where the maximum number of iterations is zero, the initial and final Reputation Profile vectors are the same, $R_j(a_i) = R_j^*(a_i)$:

```
if maxt == 0

    xnew = x0;
```

Otherwise, we carry out the first iteration with the initial Reputation Profile, $R_j(a_i)$ that is, $\mathbf{x}^{(0)}$ and initiate the next iteration at $t = 1$:

```
else

    % Execute first iteration
    x = x0;
    xnew = TJORw*x+cJORw;
    t = 1;
```

Testing against the convergence condition (or error tolerance) for each iteration of the multi-act (3.7.2.6) game:

```
    dif = norm(x-xnew);

    while t < maxt && dif(t) > e
```

Before we complete each iteration, the new $\mathbf{x}^{(k)}$ is copied to $\mathbf{x}$, the iteration is completed and t is incremented indicative of the desirable parallel qualities of the method suited to Emerging Systems:

```
        x = xnew;
        xnew = TJORw*x+cJORw;
        t = t+1;
```

Complete information is available to each node in the System at each iteration as is the requirement for non-cooperative game of this type (3.7.2.7).

Store the convergence norm values for each iteration (with padding zeros as required) - norm of the difference between two iterations:

```
        % Calculate and record difference norm values
        if(t > numel(dif))

            dif = [dif; zeros(numel(dif),1)];

        end

        dif(t) = norm(x-xnew);

    end

    dif = dif(1:t);

end
```

conclude the algorithm and return the resulting final Reputation Profile $R_j^*(a_i)$.

This algorithm implementation suits the theoretical needs of the framework and provides the structure for experimental analysis of game theoretical cost function NE solutions for reputation (and trust) profiles in a trust space. The implementation is minimal and does not incur any additional computational overheads for finding eigenvalues for instance. This makes it well suited to the relatively conservative programmable computation power of mobile nodes.

The algorithm can be parametrically enhanced to accelerate rates of convergence and restrict divergent behaviour as a means of experimental control.

□

3.8.6 Summary

In this section, we explored iterative methods for the solution of Systems of equations to establish NE Trust Values for trust spaces, and in particular, Reputation Profiles.

We derived Jacobi, Gauss-Seidel and Successive Relaxation classical stationary methods, and compared them in terms of their conditions for convergence to equilibrium solutions for trust games and their computational complexity.

We briefly consider Krylov Subspaces as the basis for more modern iterative methods, best suited to eigenvalue problems and Systems of sparse matrices.

We considered the method's efficiency of convergence and how bounds of eigenvalue spectral radii can be used as an indicative measure of rate.

Most importantly, we assessed the methods' suitability for use within the trust framework for Emerging Systems and explicitly modelled the framework undergoing iterative changes.

We established the notion of stability as the capacity of the method to ensure the robustness of the framework applied to volatile topologies - an established characteristic of Emerging Systems best suited to a random readjustment scheme in the quadratic case.

Having determined a suitable iterative method for the framework, we demonstrated a Matlab implementation of it in detail, highlighting the significant elements necessary for the framework and how they relate to the mathematical theory.

Most recently Yang and Mittal (2014) devised a methodology that accelerates the classical Jacobi iterative method by factors exceeding 100 when applied to the finite-difference approximation of elliptic equations on large grids. The method is based on a schedule of over- and under-relaxations that preserves the simplicity of the Jacobi method. Conditions and optimal schemes are applied to maximise the convergence rates that maximise convergence rates.

The key properties for the selection of the JOR iterative method for the framework are:

- classical methods' convergent properties are well established;
- no assumptions of sparsity can be made due to requirement that opinions and reputations are complete therefore matrices are dense and modern methods do not offer significant advantages;
- Jacobi method update parallelism is conducive to topologically volatile Emerging Systems;
- JOR convergence rate can be accelerated with a suitable relaxation parameter;
- JOR method models a multi-act game type, as required (3.7.2.6);
- JOR method models a full information game type, as required (3.7.2.7), and;
- JOR can be implemented programmatically, with a random, delay tolerant scheme to assure stability readjustment.

From this, we have established proof of the suitability of iterative methods and algorithms (in particular, JOR) as the computational mechanics of non-cooperative game theoretical solution techniques for the framework.

During experimental analysis, the Gauss-Seidel method will also be considered in some cases to determine what effect using values that have changed in the current iteration, rather than the values from the last iteration, has on convergence to Trust Values in the framework. This approach is part of modifications that are made to the iterative method to reflect the stability response scheme selected for testing (3.8.3).

In the next section, we identify well-defined cost functions for the Non-cooperative Programmable Open System Trust (NPOST) framework which we rigorously interrogate to determine the existence, uniqueness and stability of their NE solutions. This is carried out with a view to experimental analysis to determine robustness under scale, partitioning and with changing environmental influence.

This section concludes the establishment of the mathematical underpinnings of the NPOST framework.

3.9 A Game Theoretical Trust Framework for Emerging Systems

3.9.1 Introduction

Having identified a suitable class of function for the framework, we will formulate a well-defined Trust Function that is:

1. applied to multidimensional trust spaces;
2. influenced by multidimensional environmental spaces;
3. convex and quadratic, and therefore uniquely minimisable (NE), and;
4. first-order convergent for iterative methods (JOR)

for the purposes of experimental analysis, and to collaboratively apply the theoretical analysis and results from previous sections.

The Trust Function will be defined in terms of its components and what they influentially represent as part of the function. Similarly with Environmental Factors.

Mathematical interrogation based on what we have seen previously will ensure that the Trust Function adheres to the class stipulations and exhibits the characteristics identified necessary for the framework. Moreover, a Nash equilibrium solution to the non-cooperative game will be assured under stable circumstances.

3.9.2 Trust Functions

We specialise the game theoretical concept of cost function to a *Trust Function* in keeping with the nomenclature of the framework.

We adapt an incarnation of a Trust Function developed by Alpcan, Rencik et al. (2010) to our trust framework.

It is in fact, of little consequence what form the Trust Functions take within the class and convergence characteristics that we have set out in this section. The framework is highly adaptable in this way. The functions of Alpcan, Rencik et al. (2010) are analytically convenient and well-formed making them ideal for consideration here.

Zhang, Yu et al. (2004) propose a scheme for the classification of Trust Functions in reputation-based trust management Systems to try to increase the reuse of Trust Functions between different application domains. The scheme is designed to assist in assessing the suitability of a Trust Function by systematic analysis of its advantages and disadvantages when applied to a particular problem. The framework for the classification has its genesis in graph theory. Assumed to be transactional, Trust Functions can be described as "trust graphs" that model them in terms of trustworthiness, feedback, opinion, and source and destination of trust evaluation. Using this framework, Trust Functions can be classified by the scheme comprising four dimensions:

- subjective trust versus objective trust;
- transaction-based versus opinion-based;
- complete versus localised information, and;
- rank-based versus threshold-based.

A classification of this kind can be used in conjunction with the NPOST framework proposed here to describe the function in terms of the framework and test the suitability of a Trust Function to an Emerging Systems. It would be possible to identify classes of Trust Function, beyond just the mathematical characteristics as we have done already.

Using the Zhang, Yu et al. (2004) schema to classify seven Trust Functions from the literature:

Trust Function	Subjective trust versus objective trust	Transaction-based versus opinion-based	Complete versus localised information	Rank-based versus threshold-based
NICE (Lee, Sherwood et al. 2003)	Subjective	Transaction	Localised	Threshold
Evidence-based model (Yu and Singh 2002)	Subjective	Opinion	Localised	Threshold
PeerTrust (Xiong and Liu 2002) (Xiong and Liu 2003)	Objective	Transaction	Complete	Threshold
EigenRep (Kamvar, Schlosser et al. 2003)	Objective	Transaction	Complete	Threshold
Reputation Inference (Golbeck and Hendler 2004)	Subjective	Opinion	Localised	Rank
Trust for Semantic Web (Richardson, Agrawal et al. 2003)	Subjective	Opinion	Localised	Threshold
Heuristics Complaint Checking (Aberer and Despotovic 2001)	Objective	Transaction	Complete	Rank

Table 5 Trust Function classification

There is an alternative approach to classification proposed by Ziegler and Lausen (2004), though there is sufficient consistency between this and the Zhang, Yu et al. (2004) scheme that there is little value in comparing the two for our purposes.

3.9.2.1 NICE

NICE is a recursive acronym that stands for “NICE is the Internet Cooperative Environment” and is a cooperative framework for implementing scalable distributed applications over the Internet (Bhattacharjee 2015). Lee, Sherwood et al. (2003) propose a distributed scheme for trust inference in peer-to-peer networks based on NICE. The approach chains cookies that record the quality of

responses between nodes in the network, specifically, internet browsers. The chain forms a subgraph of the complete System and is used as evidence of how trustworthy a node is to other nodes in the network. Internet browsers are good examples of highly-programmable nodes within an Emerging System. Large amount of computation takes place within web pages that can make multiple requests for resources across the network.

The primary motivation for the approach is to form cooperative groups over large-scale networks, like the Internet. This is achieved by implementing a low overhead trust information storage and efficient search algorithm. The fundamental difference in approach for Lee, Sherwood et al. (2003) and this work is cooperation. Emerging Systems do not assume any form of cooperation between nodes. The framework could easily incorporate a cooperative approach but it would contradict our definition of an Emerging System.

3.9.2.2 PeerTrust

Based on a P-Grid (Sundaram and Babu 2015) data storage structure, PeerTrust is a reputation-based trust model for peer-to-peer Systems. A node's trustworthiness is evaluated as a normalised, weighted value derived from consensus from all nodes in the System. This results in a common view of the (objective) trustworthiness of nodes in a closed System. This approach relies on the stability of the System and assumes that all nodes can access all information in the System. This cannot be assumed in an Emerging System. However, the trust framework proposed here would satisfactorily apply to this environment as it is a simple case of an Emerging System.

3.9.2.3 Trust for Semantic Web

Richardson, Agrawal et al. (2003) propose two approaches to reconciling the local opinions of nodes into an overarching, global trust matrix, within a reputation-based trust management System. The approach is opinion-based and relies on complete information within the System. The approach makes use of Markov chains (Ephraim and Mark 2015) and an aggregation function.

The solution to the Nash equilibrium game problem proposed in this work addresses the problem of reconciling local and global opinion to some extent. Increasing the coverage of the nodes within the System to complete information makes this possible. The Richardson, Agrawal et al. (2003) approach to representation of the System as a trust matrix is similar to the apporach proposed here in that it serves as relational respresentation of the trust associations between nodes in the System. This record of relationship is the basis for all calculations in both cases.

3.9.2.4 EigenRep

EigenRep is a rank-based Trust Function with System-wide complete transaction information. The number of satisfactory and unsatisfactory transactions between each pair of nodes is collected and used to construct a matrix. The matrix is repetitively multiplied with an initial vector, until it converges. This is very similar to the iterative approach to the solution of the Nash equilibrium game proposed here. Similarly still, the initial vector is a pre-defined System parameter which contains the default trustworthiness of each node. Each entry of the converged trust vector represents a node's final trustworthiness. Every node will get the same trust vector, since the matrix and the computation process is the same for all nodes.

3.9.2.5 Reputation Inference

Golbeck and Hendler (2004) propose a wholly localised approach to their model. Each node has several trusted neighbours. Trustworthiness of a node, is inferred by polling neighbours about their trust opinion of that node. Once a binary response is received from all neighbouring nodes, to trust or not to trust, the majority opinion is adopted. A recursive process is used to poll more remote nodes until a direct connection with the node that's trustworthiness is being established, is made.

The trust graph is implicitly explored through recursive trust evaluation, which offers a simple protocol. However, since a node does not have a relatively global view of the System and no transaction information is ever collected, it is critical to choose trusted neighbours. If one or more neighbours are at enmity, a node's trust decision can be significantly influenced. This approach is potentially a very simple implementation of the trust framework proposed here. Adopting a single binary Trust Value with a rudimentary game akin to physically tossing a coin (50:50 chance game).

3.9.2.6 CloudArmor

Noor, Sheng et al. (2015) acknowledge the inherent difficulties of highly dynamic and distributed Systems. The challenges exist particularly in security, privacy and availability, all crucial for the in the delivery of cloud services. CloudArmor (CLOud consUmers creDibility Assessment and tRust manageMent of clOud seRvices) is a reputation-based trust management framework that provides a set of functionalities to deliver "Trust as a Service" (TaaS) as a "trust layer" over Service-Oriented Architecture (SOA), in a similar fashion to the layer proposed in this work. The implementation includes Trust Management Service (TMS) nodes as part of the SOA set of services, and provides a protocol and model that assures the credibility of feedback from consumers of the SOA, and an availability model that moderates and controls availability. The focus of the approach is on the robustness of the proposed credibility model against different malicious behaviours, namely

collusion and Sybil attacks (where perpetrators assume multiple identities within the System) (Sher 2015) under several behaviours, as well as the performance of the availability model.

CloudArmor is applied to a closed Emerging System where the nodes are highly programmable – they are often very large computational machines used to delivery services to consumers. While the topology is volatile, it is bounded by the cloud provider's resources which means that the TMS can act as a central authority (there is a "main" principle TMS service responsible for collation and prediction based on the "normal" instances' feedback). CloudArmor takes a bespoke probabilistic approach to its consideration of trust unlike the game theoretical one proposed here.

3.9.2.7 Trust Function

3.9.2.7.1 Description

Alpcan, Rencik et al. (2010) developed their Trust Function to model the behaviour of human agents participating in a digital trust game that tries to model users of a social network or e-commerce environment. While this is a possible application of the framework - using the framework as a formal and standard way to describe this application – the framework can be applied much more broadly to Emerging Systems.

The three terms of the Trust Function are designed to model human behaviour and can therefore be described through human (agent) characteristics as proxy variables:

- The first term quantifies the "timidness" of an agent or the willingness of the agent to pass judgement on others;
- the second term quantifies the influence of "peer pressure" on the agent, and;
- the third term quantifies the "steadfastness" of the agent to change an initial opinion.

The (quadratic) combination of these factors serves to model the tacit nature of a human agent.

Alpcan, Rencik et al. (2010) conduct an experimental study to examine how accurately the model reflects real human agent behaviour. The experiment consisted of a survey component to try to establish the traits of the agent and ultimately, how they could be reflected by weighting the Trust Function terms (timidness, peer pressure and steadfastness), and a dynamic component that measured the agent's iterative response questions when exposed to community consensus responses. Here, we are concerned with how the Trust Functions can be described within the framework, and ultimately, what happens to convergence to Nash equilibrium under conditions reflective of Emerging Systems.

TF1 is a:

- Strictly quadratic and consequently, convex three term Trust Function (3.6.7.1.1 Quadratic).

Two dimensional trust space, $\boldsymbol{M}^2$ with M_2 present as a scalar constant, $\mathbf{c}$ (0

- Numerical Example).
- Three Environmental Factors, e_{i1}, e_{i2} and e_{i3}.

3.9.2.7.2 Classification

Without specific application, it is prohibitive to apply Zhang, Yu et al.'s (2004) classification of Trust Functions to TF1. TF1 could be applied either subjectively or objectively; that a node does not evaluate trustworthiness based on quality of service but opinion, it can be argued that TF1 is subjective. The antithesis of this is that quality of service is managed by the framework (stability readjustment schema) and the game theoretical solution to TF1 would demand an objective classification. TF1 can more easily be argued to be opinion-based because of its treatment here, however there is no reason to exclude that initial Trust Values were not determined by some transactional interaction between nodes. In combination, initial values can be established through analysis of interactions and then compared as the solution to a game theoretical opinion problem between nodes to establish a final reputation for a node. The potentially arbitrary range of initial Trust Values suitable for the framework, lends itself to both threshold and ranking applications. Following a similar argument, the TF1 could be suitable for any combination to a rank and threshold application. Initial Trust Values could indicative of a threshold in the sense that they are finitely bounded with a predetermined values that determine suitability for interaction, or they could be an unbounded (or normalised) ranking.

Finally, TF1 can be applied to both complete and localised information. By definition, an Emerging System is unbounded and changing which lends itself practically, to only ever being localised information. There is no reason however, that at some time, the information is complete. From our theoretical definition of the type of game (3.7.2 Game Types), the information is complete and by Zhang, Yu et al.'s (2004) classification, local. Better suited to Emerging Systems, Zhang, Yu et al.'s (2004) observe that localised Trust Functions scale more effectively and are therefore better suited to decentralised environments. They also avoid privacy concerns (not something specifically considered here) which may arise with global Trust Functions. However, they go on to observe that global Trust Functions tend to produce preferable results due to access to all the information in the System.

3.9.2.7.3 Component Terms

$$TF1_i(x_i, \mathbf{x_{-i}}) = \frac{e_{i1}}{2}x_i^2 + \frac{e_{i2}}{2}\left(x_i - \frac{1}{N-1}\sum_{j\neq i}^{N} x_j\right)^2 + \frac{e_{i3}}{2}(x_i - c_i)^2$$

$$= \frac{e_{i1}}{2}x_i^2 + \frac{e_{i2}}{2}(x_i - \bar{\mathbf{x}}_{\mathbf{-i}})^2 + \frac{e_{i3}}{2}(x_i - c_i)^2$$

3.9.2.7.4 Environmental Factors

Environmental Factor e_{i1} weights the consensus opinion of Trust Values while e_{i2} influences the significance of the consensus trust opinion, and e_{i3} moderates the influence of M_2.

Environmental Factors are horizontally symmetric that is, $e_{i1} + e_{i2} + e_{i3} = 1$.

3.9.2.7.5 Convergence

From (3.4) and (3.10), and property (3.6.3.1.1 Additive / Sum) it is apparent that there is a unique NE solution and that the solution is iteratively convergent.

Term by term:

$$\frac{\partial}{\partial x_i}\frac{e_{i1}}{2}x_i^2 = e_{i1}x_i$$

Setting $\bar{\mathbf{x}}_{-\mathbf{i}} = \frac{1}{N-1}\sum_{j\neq i}^{N} x_j$ gives:

$$\frac{\partial}{\partial x_i}\frac{e_{i2}}{2}\left(x_i - \frac{1}{N-1}\sum_{j\neq i}^{N} x_j\right)^2 = \frac{e_{i2}}{2}\frac{\partial}{\partial x_i}(x_i - \bar{\mathbf{x}}_{-\mathbf{i}})^2$$

$$= \frac{e_{i2}}{2}\frac{\partial}{\partial x_i}\left(x_i^2 - 2\bar{\mathbf{x}}_{-\mathbf{i}}x_i + {\bar{\mathbf{x}}_{-\mathbf{i}}}^2\right) = e_{i2}x_i - e_{i2}\bar{\mathbf{x}}_{-\mathbf{i}}$$

and finally:

$$\frac{\partial}{\partial x_i}\frac{e_{i3}}{2}(x_i - c_i)^2 = e_{i3}x_i - e_{i3}c_i$$

Then:

$$\frac{\partial TF1_i}{\partial x_i} = e_{i1}x_i + e_{i2}x_i - e_{i2}\bar{\mathbf{x}}_{-\mathbf{i}} + e_{i3}x_i - e_{i3}c_i$$

By virtue of the Environmental Factors being horizontally symmetric, $e_{i1} + e_{i2} + e_{i3} = 1$, we have:

$$\frac{\partial TF1_i}{\partial x_i} = x_i - e_{i2}\bar{\mathbf{x}}_{-\mathbf{i}} - e_{i3}c_i$$

From the strict convexity property of TF1, it is sufficient to check the first order condition for optimality.

Substituting back for $\bar{\mathbf{x}}_{-\mathbf{i}}$:

$$\frac{\partial TF1_i}{\partial x_i} = 0 \Rightarrow x_i^* = \frac{e_{i2}}{N-1}\sum_{j\neq i}^{N} x_j^* + e_{i3}c_i$$

For completeness:

$$\frac{\partial^2 TF1_i}{\partial x_i^2} = 1 > 0$$

Therefore, TF1 is strictly convex in $x_i$and its minimisation admits a unique globally optimal solution and there is a unique Nash equilibrium solution to the trust game with TF1 Trust Function, x_i^*.

Of note, the first Environmental Factor (e_{i1}) is no longer influential (degenerated) in the first derivative of the Trust Function.

The solution can be represented in matrix form:

$$\mathbf{x}^* = \mathbf{A}\mathbf{x}^* + \mathbf{c}$$

where $\mathbf{c}$ in this case is $e_{i3}c_i$ and:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{e_{12}}{N-1} & \cdots & \frac{e_{12}}{N-1} \\ \frac{e_{22}}{N-1} & 0 & \cdots & \frac{e_{22}}{N-1} \\ \vdots & \vdots & \ddots & \frac{e_{i2}}{N-1} \\ \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & 0 \end{bmatrix}$$

We have:

$$\mathbf{x}^* = \begin{bmatrix} 0 & \frac{e_{12}}{N-1} & \cdots & \frac{e_{12}}{N-1} \\ \frac{e_{22}}{N-1} & 0 & \cdots & \frac{e_{22}}{N-1} \\ \vdots & \vdots & \ddots & \frac{e_{i2}}{N-1} \\ \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & 0 \end{bmatrix} \begin{bmatrix} x_1^* \\ x_i^* \\ \vdots \\ x_N^* \end{bmatrix} + \begin{bmatrix} e_{13}c_1 \\ e_{i3}c_i \\ \vdots \\ e_{N3}c_N \end{bmatrix}$$

The solution is:

$$\mathbf{x}^* - \mathbf{A}\mathbf{x}^* = \mathbf{c} = \mathbf{x}^*(\mathbf{I} - \mathbf{A})$$

$$\Rightarrow \mathbf{x}^* = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{c}$$

and:

$$\mathbf{I}-\mathbf{A}=\begin{bmatrix}1 & 0 & \dots & 0\\ 0 & 1 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & 1\end{bmatrix}-\begin{bmatrix}0 & \frac{e_{12}}{N-1} & \dots & \frac{e_{12}}{N-1}\\ \frac{e_{22}}{N-1} & 0 & \dots & \frac{e_{22}}{N-1}\\ \vdots & \vdots & \ddots & \frac{e_{i2}}{N-1}\\ \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & \frac{e_{N2}}{N-1} & 0\end{bmatrix}$$

$$=\begin{bmatrix}1 & \frac{-e_{12}}{N-1} & \dots & \frac{-e_{12}}{N-1}\\ \frac{-e_{22}}{N-1} & 1 & \dots & \frac{-e_{22}}{N-1}\\ \vdots & \vdots & \ddots & \frac{-e_{i2}}{N-1}\\ \frac{-e_{N2}}{N-1} & \frac{-e_{N2}}{N-1} & \frac{-e_{N2}}{N-1} & 1\end{bmatrix}$$

By the same reasoning as (3.7.6.1 Example Continuation), $\mathbf{I}-\mathbf{A}$ is full rank, strictly diagonally dominant and it is therefore non-singular and invertible.

Then:

$$x_i^*=\frac{e_{i2}}{N-1}\sum_{j\neq i}^{N}x_j^*+e_{i3}c_i$$

Setting $\bar{\mathbf{x}}=\sum_i^N x_i$ and $\bar{\mathbf{c}}=\sum_i^N c_i$:

$$\sum_{j\neq i}^{N}x_j^*=\bar{\mathbf{x}}^*-x_i^*$$

$$\Rightarrow x_i^*=\frac{e_{i2}}{N-1}(\bar{\mathbf{x}}^*-x_i^*)+e_{i3}c_i$$

$$\Rightarrow (1+\frac{e_{i2}}{N-1})x_i^*=\frac{e_{i2}}{N-1}\bar{\mathbf{x}}^*+e_{i3}c_i$$

and:

$$\bar{x}=\frac{e_{i2}}{1-e_{i2}}\bar{c}$$

$$x_i^*=\frac{e_{i3}}{N-1+e_{i2}}\left(\frac{e_{i2}}{1-e_{i2}}\bar{c}+(N-1)c_i\right)$$

Substituting back into the Trust Function, gives:

Set:

$$A = \frac{1}{N-1}\sum_{j\neq i}^{N} x_j^*$$

$$\Rightarrow x_i^* = \frac{e_{i2}}{N-1}\sum_{j\neq i}^{N} x_j^* + e_{i3}c_i = e_{i2}A + e_{i3}c_i$$

Then:

$$TF1_i(x_i^*, \mathbf{x}_{-\mathbf{i}}^*) = \frac{e_{i1}}{2}(e_{i2}A + e_{i3}c_i)^2 + \frac{e_{i2}}{2}(e_{i2}A + e_{i3}c_i - A)^2 + \frac{e_{i3}}{2}(e_{i2}A + e_{i3}c_i - c_i)^2$$

$$= \frac{e_{i1}}{2}(e_{i2}A + e_{i3}c_i)^2 + \frac{e_{i2}}{2}((e_{i2}-1)A + e_{i3}c_i)^2 + \frac{e_{i3}}{2}(e_{i2}A + c_i(e_{i3}-1))^2$$

$$= \frac{e_{i1}}{2}\left(e_{i2}\frac{1}{N-1}\sum_{j\neq i}^{N} x_j^* + e_{i3}c_i\right)^2 + \frac{e_{i2}}{2}\left((e_{i2}-1)\frac{1}{N-1}\sum_{j\neq i}^{N} x_j^* + e_{i3}c_i\right)^2$$
$$+ \frac{e_{i3}}{2}\left(e_{i2}\frac{1}{N-1}\sum_{j\neq i}^{N} x_j^* + c_i(e_{i3}-1)\right)^2$$

Since matrix $\mathbf{I} - \mathbf{A}$ has $|a_{ii}| > \sum_{j\neq i}|a_{ij}| \ \forall\, i$, it is strictly diagonally dominant, it is therefore non-singular and invertible by Levy–Desplanques theorem (Horn and Johnson 1990) (equivalent to the Gerschgorin Circle Theorem (Olver 2008)), and the determinant of $\mathbf{A}$ is not equal to zero, $det(\boldsymbol{A}) \neq 0$.

Theorem (Levy–Desplanques): A strictly diagonally dominant matrix is non-singular. In other words, let $\mathbf{A} \in \mathbb{C}^{\boldsymbol{n,n}}$ be a matrix satisfying the property (Taussky 1949) (Schneider 1977):

$$|a_{ii}| > \sum_{j\neq i}|a_{ij}| \ \forall\, i$$

Moreover, $\mathbf{I} - \mathbf{A}$ is full rank and therefore, the linear System converges.

□

3.9.3 Summary

Finally, we have derived a well-constructed Trust Function as candidate for the experimental analysis of the framework.

The function is described in terms of the mathematical framework, proved to uniquely converge by virtue of its convex classification, and influenced by well-defined Environmental Factors.

While the function derived is quadratic, there is no restriction on the class of function that could be used as long as it can be shown to converge. Divergent functions will not exhibit an equilibrium and other functions may exhibit multiple.

3.10 Conclusion

The formal mathematical description of the framework is established in this chapter. Using graph theoretical techniques, we are able to represent an Emerging System and examine the trust relationships between its constituent nodes. The set of these descriptive mathematical entities (graphs, association matrices, trust spaces, Reputation Profiles and trust profiles) allow us to represent and examine a trust System.

The framework is then extended to use the convergence of node cost functions to establish consensus of trust. We have been able to determine final values for trust and reputation based on the opinions of neighbouring nodes in the System, and initial Trust Values.

A complete representation of a System's trust relationships needs to be multidimensional so that all components of trust are present. Components of trust can influence each other, a relationship that within represented within the Trust Function itself, often as a constant.

Within any System, there are factors that universally affect the constituent nodes. Represented as Environmental Factors within the framework, they can present themselves as administrative configuration controls or any other common influencing factor.

Limiting the classification of Trust Functions to convex assures the uniqueness of equilibrium solutions. The framework can accommodate any class of function but convergence and uniqueness cannot be guaranteed. Convex functions can be constructed from other convex functions under their additive property and scaled, and convexity can be determined for a function can be determined through derivative analysis.

The role of Game Theory is identified within the framework. We have been able to determine the characteristics of a non-cooperative game and accommodate them within the framework. The primary result is that the game type suits the characteristics of Emerging Systems.

To experimentally examine the behaviour of determining consensus trust within the trust framework, we consider iterative approaches to solutions of Trust Function Systems and determine JOR as a suitable method. The iterative algorithm will be used to compute final consensus Trust Values within a testing environment. When deployed, the computation will take place on each node within the Emerging System.

For the purposes of experimental analysis and to collaboratively apply the theoretical analysis and results from this chapter, we describe, derive and prove a well-defined Trust Function, TF1. It is in multidimensional trust spaces, influenced by multidimensional environmental spaces, convex and quadratic, and convergent for the JOR iterative method.

In this chapter, we have been able to:

1. **demonstrate the formulation of mathematical constructs can define a trust nomenclature as a foundation for a trust framework;**
2. **prove the suitability of rigorous applications of non-cooperative game theoretical techniques to establish stability and equilibrium applied to the constructs;**
3. **prove the suitability of iterative methods and algorithms as the computational mechanics of these techniques for a trust framework, and;**
4. **derive a well-constructed cost function as a candidate for the experimental analysis of the trust framework.**

3.10.1 Conceptual Model

We are able to represent the conceptual model augmented with the mathematical concepts that have been established in this chapter:

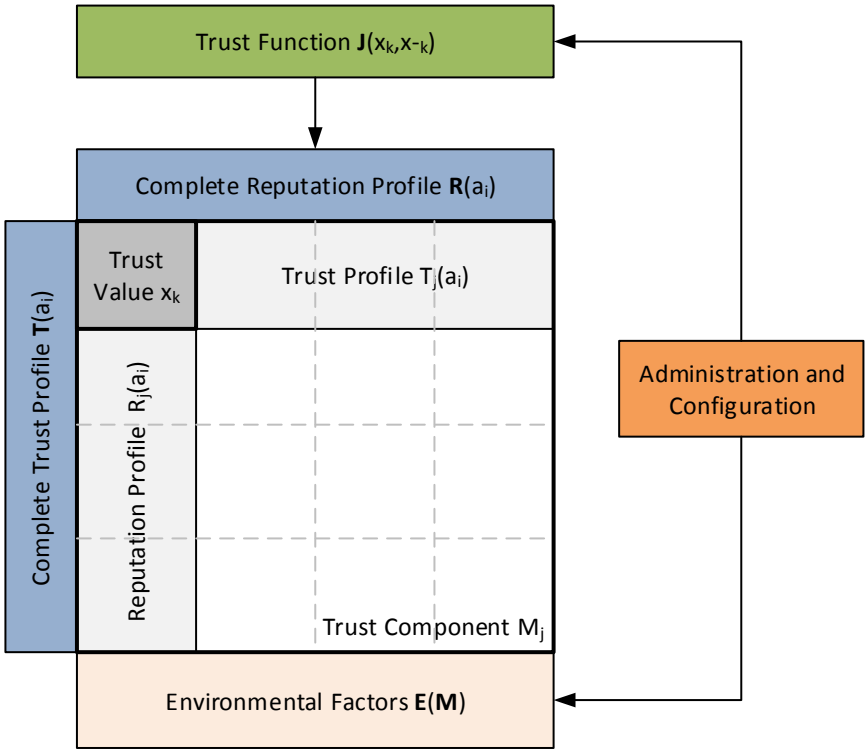


*Figure 26 Conceptual Model of the trust framework for Emerging Systems with dim (**M**) = n and I,j,k=1,2,…*

4 Experimental Analysis

4.1 Introduction

Forgotten often are elementary rules of logic, that extraordinary claims require extraordinary evidence and that what can be asserted without evidence can also be dismissed without evidence – “Quod gratis asseritur, gratis negatur” (Hitchens 2003).

Abstaining to be tied to any subjectivism, this chapter describes the results and conclusions of the experimental analyses of the trust functions derived in the previous chapter within the trust NPOST framework and how they support the research objectives. We explore research methods in general, extending the discussion specifically to experimental methods upon the theory of which, the experiments are based.

From the literature, we identified the importance of examining the framework experimentally to establish how it behaves when exposed to conditions characteristic of Emerging Systems, particularly high topological volatility, variations in node volume, and under different environmental factors.

If a game has a unique Nash equilibrium and is played among agents under certain conditions, then the Nash equilibrium strategy set will be adopted. Sufficient conditions to guarantee that a Nash equilibrium game is played are (Aumann and Brandenburger 1995) (Nash 1951):

1. The agents all will do their utmost to maximise their expected payoff as described by the game;
2. The agents are flawless in execution;
3. The agents have sufficient intelligence to deduce the solution;
4. The agents know the planned equilibrium strategy of all of the other agents;
5. The agents believe that a deviation in their own strategy will not cause deviations by any other agents, and;
6. There is common knowledge that all agents meet these conditions, including this one. So, not only must each agent know the other agents meet the conditions, but also they must know that they all know that they meet them, and know that they know that they know that they meet them, and so on.

For Emerging Systems, we are questioning change in the first condition and somewhat, the second. Each iteration of the algorithm is a complete game. Topological volatility occurs between and during games where a game’s nodes are best fit to what is locally available at a point in time.

The Non-cooperative Programmable Open System Trust (NPOST) framework is simulated here and the results considered against our contribution and significance statements of claim.

4.1.1 Contribution and Significance

The contribution and significance of this chapter is the results showing the robustness of the Non-cooperative Programmable Open System Trust (NPOST) framework under scale, partitioning and with changing environmental influence for the previously derived trust functions, when iteratively calculated. These variables are controlled to simulate conditions within an Emerging System.

The contribution and significance of this chapter is to support:

1. **proof of the suitability of the NPOST framework for Emerging Systems;**
2. **proof of the practical implementation potential of the NPOST framework;**
3. **proof of the robustness of the NPOST framework:**
    a. **when scaled;**
    b. **when partitioned, and;**
    c. **under changing environmental influencing factors.**

By "robustness", we mean the ability of the framework to continue to establish final reputation profiles effectively without exceeding reasonable bounds of computational effort or simply failing (Pakazad, Hansson et al. 2015).

4.1.2 Roadmap

This chapter covers five main topic areas:

- Research Methods;
- Experimental Research Methods;
- Hypotheses;
- Method, and;
- Results.

4.1.2.1 Research Methods

The *Research Methods* section describes traditions in research methods and associated reasoning styles. It discusses positivism and Postpositivism, Interpretivism and mixed methods.

4.1.2.2 Experimental Research Methods

The *Experimental Research Methods* section specialises the research method discussion to determining if a specific treatment influences an outcome in the positivist or scientific method.

4.1.2.3 Hypotheses

From the General Hypothesis, the *Hypothesis* section establishes an Operational Hypothesis that provide highly testable cases that can be simulated.

4.1.2.4 Method

The experimental *Methods* section describes how the experiments are to be carried out in terms of participants, variables, instruments and materials, threats to validity, procedures and analysis of the data.

4.1.2.5 Results

In the final section *Results*, the results of the experiments are explicated through collation, analysis, and interpretation.

4.2 Research Methods

4.2.1 Introduction

Research is an inquiry process that has clearly defined parameters and has as its aim, the (McClure and Hernon 1991):

- Discovery or creation of knowledge, or theory building;
- Testing, confirmation, revision, refutation of knowledge and theory; and/or
- Investigation of a problem for local decision making.

4.2.2 Traditions

There are two major traditions of research, *positivist* and *interpretivist*. The positivist tradition primarily concerns itself with the collection of qualitative data collected by instruments of measure. While the emphasis for the interpretivist (or interpretive) tradition, is meaning derived from qualitative techniques. Invariably neither traditional is adopted exclusively, rather both types of data and data collection approaches are used in conjunction to accomplish the aims of the research, though they might appear dichotomous in principle (mixed methods) (Williamson, Bow et al. 2002).

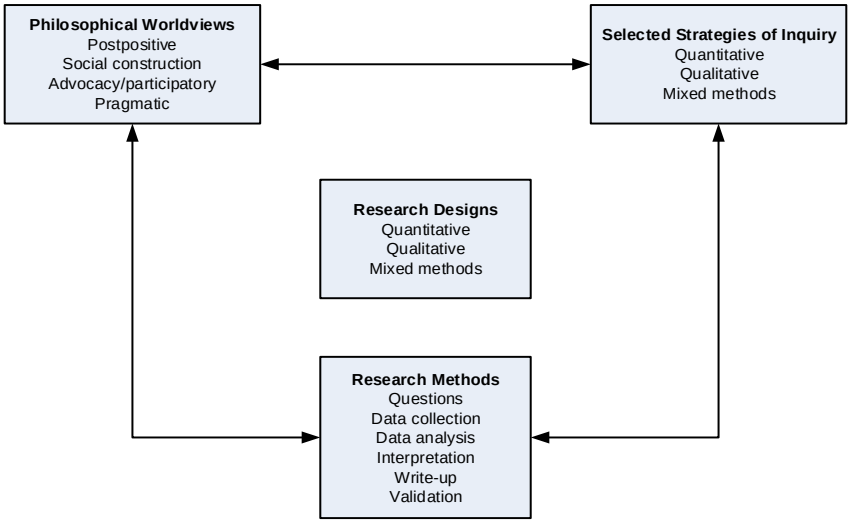


Figure 27 Framework for Design - The interconnection of Worldviews, Strategies of Inquiry and Research Methods (Creswell 2003)

Creswell (Creswell 2003), appeals to a principle of a "world view" as a basic set of beliefs that guide action (Lincoln 2000) for a philosophical research approach as part of a framework for research design.

The framework associates Philosophical Worldviews with Research Strategies, Methods and Designs. Central to the framework are the two dominant research traditions and designs.

4.2.3 Reasoning Styles

Associated with the two schools of research are different reasoning styles. *Deductive* reasoning is associated with the scientific, positivist approach and *inductive* reasoning, with the interpretivist approach (Williamson, Bow et al. 2002) (Creswell 2003).

- Deductive reasoning exhibits a hypothesis testing approach to research, where the argument moves from general principles to particular instances;
- Alternatively, inductive reasoning begins with particular instances and ends with general statements or principles.

4.2.4 Positivism and Postpositivism

A positivist / postpositivist approach to research considers the world as a collection of observable facts and events that can be measured. Research designs more commonly associated with this approach are experimental or surveys.

Central to the positivist approach is:

- Seeking to link cause and effect, and;
- Empiricism such that all scientific knowledge is based on objectively observed impressions.

Postpositivism is associated with quantitative research which is a means for testing objective theories by examining the relationship among variables. These variables, in turn, can be measured, typically on instruments, so that numbered data can be analysed using statistical procedures (Creswell 2003).

Assumptions are made about testing theories deductively, building in protections against bias, controlling for alternative explanations, and being able to generalise and replicate findings (Creswell 2003).

4.2.4.1 Research Design

For a postpositivist research design, deductive styles of reasoning determine a hypothesis against which, the relationships between variables selected for study are tested. Observations on random samples attempt to draw nomothetic conclusions that corroborate or refute (actively in the deductivist case (POPPER 2002)) the hypothesis.

The research design, deductive approach is linear in nature:

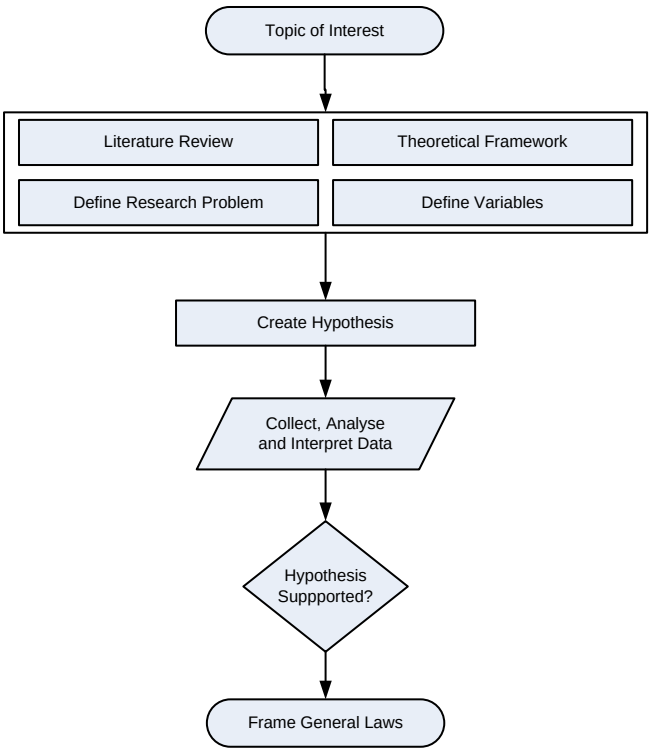


Figure 28 Positivist research design (Williamson, Bow et al. 2002)

Quantitative methods tend to be characterised by data collection methods (Creswell 2003):

- Pre-determined;
- Instrument based questions;
- Performance data, attitude data, observational data and census data;
- Statistical analysis, and;
- Statistical interpretation.

4.2.5 Interpretivism

Associated with qualitative research methods, the interpretivist school of research ostensibly encompasses:

- Constructivism;
- Grounded Theory;
- Phenomenology;
- Narrative;
- Case Study;
- Critical Theory;
- Symbolic Interactionalism, and;
- Ethnography.

The central interpretivist tenet is that people are constantly evolving their interpretation of their social constructions and of themselves. People construct their own perception of reality and develop their own meanings that could differ radically from others. They are constantly involved in making sense of and interpreting their world (Williamson, Bow et al. 2002).

Interpretivists regard their research task as coming to understand how the participants in social setting interpret the around them. Their concern is with the beliefs, feelings and interpretations of participants, and recording these perspectives as accurately as possible (Creswell 2003).

4.2.5.1 Research Design

Interpretivist research design is more interactive than the linear positivist approach. Hypotheses are not explicitly formed but rather, propositions are posed which are grounded in the perspectives of the participants. Idiographic studies tend not to yield generalisations and consequently, do not make demands on replication or randomisation. Samples are selected purposefully for investigation of a specific problem.

Validity and reliability as measures are still important to an interpretivist approach. Rigour can still be assured through the consistency checks and triangulation approaches, for instance.

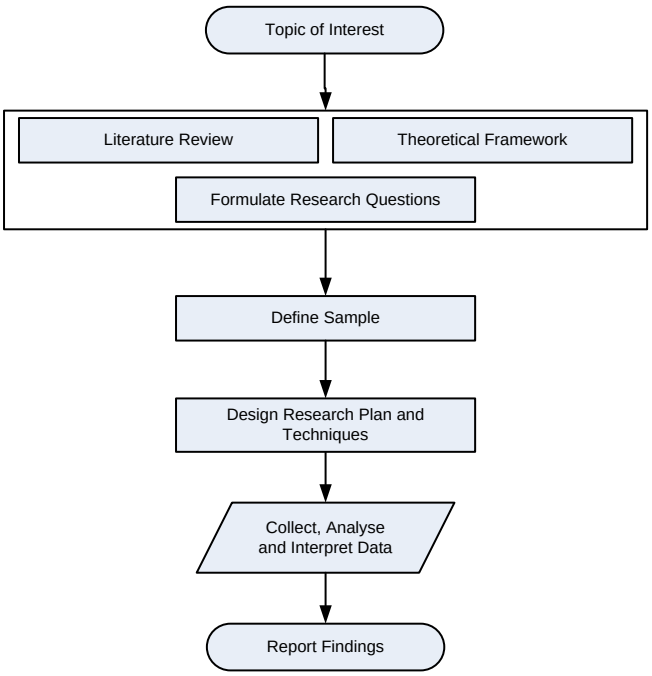


Figure 29 Qualitative research design (Williamson, Bow et al. 2002)

Qualitative methods tend to be characterised by data collection methods (Creswell 2003):

- Emerging;
- Open-ended questions;
- Interview data, observation data, document data and audio-visual data;
- Text and image analysis, and;
- Themes, patterns interpretation.

4.2.6 Mixed Methods

Mixed methods is an approach to inquiry that combines or associates both qualitative and quantitative forms. It involves philosophical assumptions, the use of qualitative and quantitative approaches, and the mixing of both approaches in a study (Creswell 2003).

Matching the overall approach to the nature of the research question appears sound reasoning. Triangulation is an approach that validates results between data collection methods. These methods could be qualitative and quantitative.

4.2.7 Summary

The two major approaches to research are broadly labelled “positivist” and “interpretivist”. The former attempts to apply scientific methods and is most usually associated with deductive reasoning and quantitative data collection. Because of their use of natural settings and greater emphasis on qualitative data collection, post-positivists have some characteristics in common with interpretivists, although they still believe that there is a reality which can be measured. Interpretivists, on the other hand, are concerned with meanings constructed by individuals and groups, use principally inductive reasoning and naturalistic inquiry, constructivism and phenomenology (Williamson, Bow et al. 2002).

4.3 Experimental Research Methods

Experimental research exemplifies a classical positivist approach or scientific method. This research tradition is based on hypothesis testing, on a deductive process of logical inference, where reasoning proceeds from general principles to particular instances (Williamson, Bow et al. 2002).

Experimental research seeks to determine if a specific treatment influences an outcome. This impact is assessed by providing a specific treatment to one group and withholding it from another and then determining how both groups scored on an outcome (Creswell 2003).

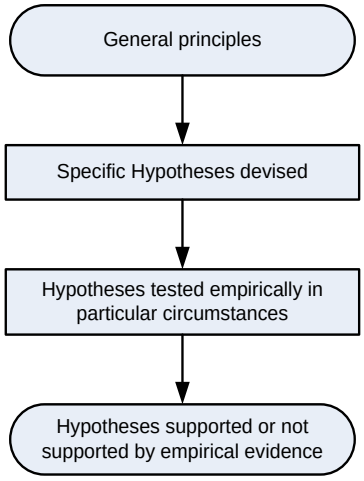


Figure 30 Deductive reasoning process for experimental research methods (Williamson, Bow et al. 2002)

The *true experiment* is a hypothetico-deductive research model and is a classic example of the scientific method. It is very well suited to laboratory controlled cause-and-effect relationships where isolation and control of variables is possible. Experimental groups experience some condition of the independent variable and are then measured on the dependent variable to establish a relationship.

Possible experimental designs include:

- True:
    - Randomised two group;
    - Pre-test/post-test control group;
    - Post-test only control group;
    - Solomon four-group;
    - A-B-A single-subject, and;
    - Factorial.
- Pre:
    - One-shot;
    - One-group pre-test/post-test
    - Static group comparison;
    - Alternative treatment post-test only with non-equivalent groups, and;
    - Randomised two group.
- Quasi:
    - Pre-test/post-test non-equivalent control group;
    - Single group interrupted time series;
    - Control group interrupted time series, and;
    - Regression-discontinuity.

Experiments include true experiments, with random assignment of subjects to treatment conditions, and quasi-experiments that use non-randomised designs (Keppel 1991). Included with quasi-experiments are single-subject designs (Creswell 2003).

For this work, we have a Postpositivist worldview, an experimental strategy of inquiry, and pre- and post-test (factorial) measurements of statistical significance (Creswell 2003).

For Emerging Systems (the Topic of Interest), we have established the General Hypothesis, that there is a need for an open system trust framework of reprogrammable nodes (an Emerging System), from a review of the current and classical literature.

4.4 Hypotheses

We have posited that the distributed and open nature of these systems is well suited to a non-cooperative game theoretical approach to the framework.

Further, we have explored suitable Research Hypotheses through a rigorous analysis of an underlying mathematical trust framework and established its theoretical suitability for Emerging Systems.

Now we can interrogate Operational Hypotheses for highly testable cases within a simulated implementation of the NPOST framework.

4.5 Method

4.5.1 Participants

The participants in the experiments are N simulated nodes a_{-k} in an Emerging System and their associated initial trust values x_i within a range $q \leq x_i \leq r$, that form a reputation profile of a node a_k in the system, $R_j(a_k)$ in some trust space component, M_j:

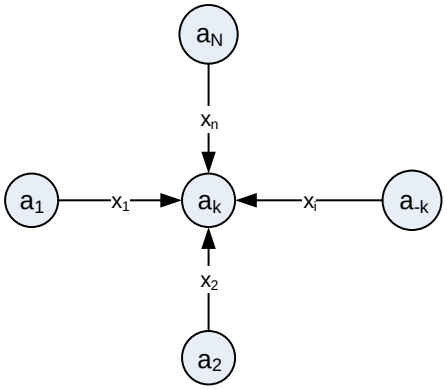


Figure 31 Experiment participants

with $q, r, x_i \in \mathbb{Z}, i = 1,2, \dots, n \in \mathbb{N}, k = 1,2, \dots, N \in \mathbb{N}$ and $n = N - 1$.

We select randomly uniformly generated node participants for the experiment and associated trust values, hence, we ensure that each node has equal probability of being selected and is representative of some whole population. This is a necessary condition for the selection of a true, pure scientific experimental design. We can disregard quasi and pre-designs on this sufficient basis.

Moreover, the participants are all simple, generated numeric integer values in a range and so, for all practical purposes, identical. This eliminates any possibility of systematic differences between them

that could affect outcomes so that any differences in outcomes can be attributed to experimental treatment with a high level of certainty.

4.5.1.1 Sample Size

The number of participating nodes will vary between experiments as this is an experimental variable used to determine the behaviour of the framework simulation when scaled.

The sample size is given by:

$$\dim\left(R_j(a_k)\right) = N - 1 = n$$

A power approach (Lenth 2001) or similar to determining a statistically significant sample size is not appropriate here since we are not conducting strictly significance statistical test experiments. There are not physical limitations on the size of the sample as the participants are not human and resources are not scare. There may be constraints due to limitations of the simulation environment to be determined as an experiment in itself. A suitable sample size then, is best determined by the context of the experiment, which is the Emerging Systems we are simulating. It is important that the framework be tested on small and large size systems for it to be established suitable for Emerging Systems in general.

The sample size range from 10 (primarily a control) to 10,000,000 nodes depending on the hypotheses of the experiment, reflecting small localised or fragmented communities, and very large communities of nodes.

30,000,000 is approximately the number of mobile devices (not necessarily highly reprogrammable) in use in Australia (population: 22,700,000, mobile telephones: 30,200,000, penetration percentage: 133, year: 2011- 2010 (Chapman 2012)). An Emerging System of this size would be substantial and highly volatile.

There are approximately 13.5 billion connected devices, according to Ericsson (Ericsson 2015), including (in order of size) mobile phones, PC / laptop / tablets / routers, connected Consumer Electronics, machine-to-machine (M2M) and fixed phones.

Ericsson (Ericsson 2015) forecast growth in connected devices and the Internet of Things (IoT). Cisco have also said there will be 50 billion “connected things” by 2020, while Huawei forecast there will be 100 billion terminals interconnected by the internet by 2025 (Costello 2015), as Emerging Systems become more embedded in everyday life and across all industries. ZTE forecast that there will be 100 billion connected devices by 2020 (Costello 2015).

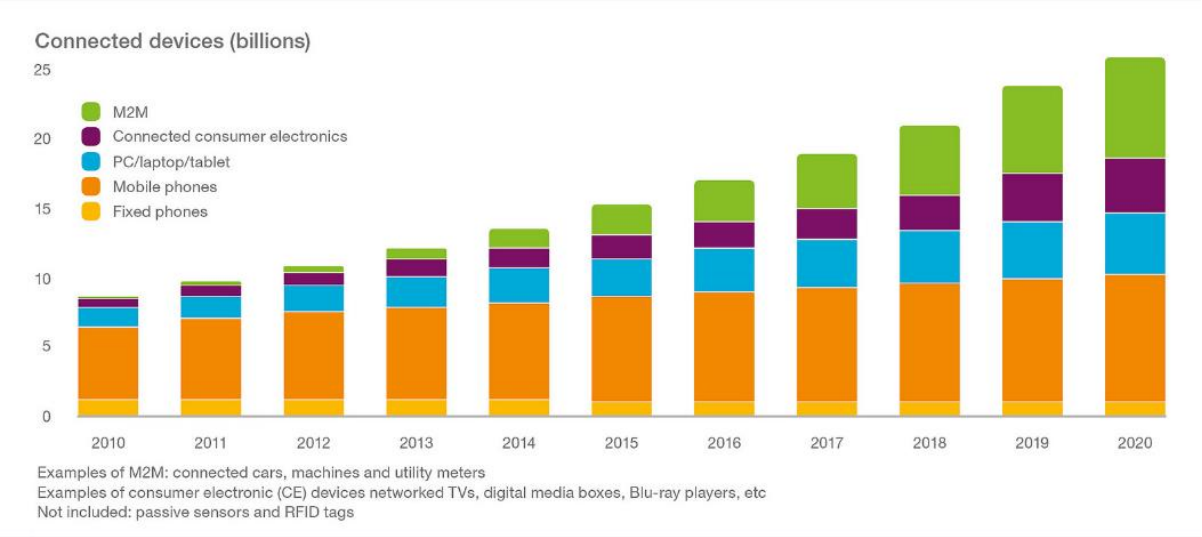


Figure 32 Connected devices

4.5.2 Variables

We need to experimentally understand the behaviour of the framework under certain circumstances to establish a cause-and-effect relationship between defined variables (Creswell 2003) (Williamson, Bow et al. 2002).

For each independent experiment, we define:

- Null Hypothesis against which we will determine significant difference - (H_0);
- Alternate hypotheses ($H_1, H_2, \dots, H_n$);
- Experimental variables ($X_1, X_2, \dots, X_n$);
- Observation variables ($O_1, O_2, \dots, O_n$);
- Any Extraneous variables represented by experimental variables, and;
- Any Moderating variables represented by experimental variables.

We will establish a control (Not-X_i) and experimental state (X_i) by setting initial values for experimental variables. Initial trust values will be establish through a process of randomisation (R_i) to ensure the states are equivalent in composition.

The experiments determine the effects of variations in the following variables which also parameterise the control state:

4.5.2.1 Experimental Variables

Experimental variables can be divided into two groups with a common concern – variables that directly affect the Non-cooperative Programmable Open System Trust (NPOST) framework and those

that serve to calibrate the iterative method algorithm. One or many of the experimental variables is identified for each independent experiment as the independent variable.

4.5.2.1.1 NPOST Framework

X_1	=	Trust Function ($TFn_i(x_i, \mathbf{x_{-i}})$) TF1
X_2	=	Complete Trust Space dimension ($\dim(\boldsymbol{M})$)
X_3	=	Initial Trust Value range ($q \leq x_i \leq r$)
X_4R_1	=	Initial Reputation Profile Trust Values ($R_j(a_k) = \mathbf{x}_i, R_1$)
X_5	=	Initial Reputation Profile dimension ($\dim(R_j(a_k))$)
X_6R_2	=	Environmental Factors ($\boldsymbol{E}$, R_2): • horizontally symmetric ($\sum_{j=1}^{n} e_{kj} = 1$) and horizontally non-symmetric ($\sum_{j=1}^{n} e_{kj} \neq 1$), • dominant ($e_i^* \geq e_{-i}$) and strictly dominant ($e_i^* > e_{-i}$), • vertically symmetric ($e_{kj} = e_{lj}$) or vertically non-symmetric (e_{kj} does not necessarily equal e_{lj}), and; • uniformly, pseudorandom

4.5.2.1.2 Iterative Method (JOR) Algorithmics

X_7	=	Stability strategy and readjustment scheme (termination of node after s unresponsive requests and reinstatement criteria – accumulative / consecutive and correction)
X_8R_3	=	Determinant of a node's availability in the system for an iteration, R_3: • static non-responsive (0), • static responsive (1), and; • uniformly pseudorandom (2)
X_9	=	Convergence condition or error tolerance: ($\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon$ for all Δx or for any Δx)
X_{10}	=	Upper iteration bound, ($\sup(t) \leq N$)
X_{11}	=	Relaxation parameter, (ω)

Interpretation is corroborated with reference to the observational variables:

4.5.2.2 Observational Variables

O_1	=	Number of algorithm iterations ($N \geq \sup(t)$)
O_2	=	Norm of the differences between subsequent iterations over time ($\varepsilon^{(t)}$)
O_3	=	Computation real execution time (seconds)
O_4	=	Final Reputation Profile Trust Values ($R_j^*(a_k) = \mathbf{x}_i^*$)
O_5	=	Stability (After $\sup(t) \leq N$ iterations, $\left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon \Rightarrow$ convergence otherwise, divergence)
O_6	=	Final Reputation Profile dimension ($\dim(R_j^*(a_k))$)

4.5.2.3 Random Variables

A necessary condition for applying true testing procedures in experiments is that experimental group members must be randomly assigned. This ensures the groups are of equivalent composition (Williamson, Bow et al. 2002).

It is reasonable to assume that nodes within an Emerging System will hold opinions that can be described by a normal statistical distribution since certainly, nodes in close proximity will experience similar interactions with a common node. Since all information is complete and the multi-act games non-cooperative, we can assume uniformity or normality in the distribution of opinions and consequently, trust values.

We will utilise the simulation environment's own pseudorandom value generation capabilities (Awad 2010) to provide the initial values and "shuffle" the generator with each testing session to ensure a high randomisation integrity, although this approach is not considered of statistical importance generally (MathWorks 2015).

Experiments conducted where there is a requirement for some nodes within the system to become unresponsive or leave the system all together, will be modelled in a randomly selective manner. Uniformly distributed random values will be used.

4.5.3 Instrumentation and Materials

The experiments are conducted in MathWorks Matlab (Strum and Kirk 1999) – the *simulation environment*. They are carried out as procedural, algorithmic functions based JOR iterative method. A complete script listing can be found in, 7.1 Appendix: NPOST Simulation Matlab Script Listings.

The hardware is consistent for all experiments (carried out on the same computer). Running software is limited to the simulation environment, operating system and any absolutely necessary supporting system software. A system resource monitoring application will be used to assure the hardware state, identify inconsistent behaviour and monitor performance during experiments. While it is most important that the simulation environment be consistent so that relative significance can be established and attributed to the experimental treatment, for reference however, the pertinent simulation environment hardware and software specifications are:

- Hardware:
    - Intel Core i7-3720QM CPU @ 2.60GHz (4 cores, 8 threads);
    - 16GB DDR3 PC-12800 1600MHz SODIMM RAM;
    - 256GB SSD HDD.
- Software:
    - Microsoft Windows 7 Professional 64-bit;
    - MathWorks Matlab R2012a (7.14.0.739) 64-bit.

To eradicate anomalistic results directly attributable to the simulation environment, each experimental will be repeated five times while physical resources are monitored and mean results used for analysis.

Pilot tests on the physical environment were carried out to establish its suitability for supporting the simulation environment and operating environment effectively. It was determined that there was a very low variability between system responses for repeated elementary large matrix calculations.

The hardware specifications more than adequately fulfil the recommended system requirements for the operating (Microsoft 2013) and simulation (MathWorks 2012) environments.

The MATLAB.EXE process is assigned a single CPU affinity and priority nominated as “high”. Since the simulation environment is single-threaded, this dedicates a single CPU to the purpose of running it and assures that the highest priority is given to the simulator. This allocation will help to ensure that all experiments have similar resources available to them, limiting the prospect of an instrumental threat to validity. CPU affinity allocation can also take advantage of more frequent CPU cache hits, making the simulation more efficient (TechNet 2015).

The JOR iterative method algorithm was pilot tested successfully against known results and analysed rigorously as part of the mathematical Non-cooperative Programmable Open System Trust (NPOST) framework previously, to ensure the integrity of the simulation environment implementation. Refinements were made to increase speed of calculation, reduce reporting overhead, increase

control and reduce resource footprint. As with the hardware and software specifications, it is most important that the implementation be consistent for all independent experiments.

4.5.4 Threats to Validity

The experiments are high in internal validity as the independent variables are controllable and easily malleable. We can be significantly certain that the effects observed are attributable to the independent variable.

This also means some typical rival hypotheses can be rejected fairly easily, without much further consideration (Creswell 2003) (Shadish, Cook et al. 2002) (Tuckman 1988):

Threat to Internal Validity	**Mitigation**
History	Participants are not influenced by events over time. Each group of participants is generated as required for each independent experiment.
Maturation	Participants do not mature over time.
Regression	Extreme participants have no opportunity to regress towards mean values.
Selection	The selection process is definitively pseudo-random.
Mortality	The participants are not mortal. Removal of some participants from the experiments is part of the experimental variable variations – controlled by the variable, $\boldsymbol{u}$.
Diffusion of treatment	There is no possible communication between participants.
Compensation / resentful demoralisation	There are no benefits to the participants.
Compensatory rivalry	Participants are incapable of experiencing devaluation.
Testing	Participants are ignorant of the testing process.
Instrumentation	The most applicable of the threats – instruments have been pilot tested, validated and are monitored throughout the experiments for any anomalistic behaviour (4.5.3). Pseudo-random variables are shuffled each experimental session reduce the chance of repeated results (4.5.2.3).

While the experimental approach adopted here is true experimental in nature, there is potential for a further *information systems* approach for establishing causation and enabling generalisation outside of laboratory conditions. This is largely dependent on the application of the framework – whether it is a purely machine-to-machine system or it includes human nodes, for instance.

Further, while the experimental results established are within a single system domain, the topology of open systems is by definition, heterogeneous in nature transcending multiple system boundaries. This limits the external validity of the experiments outside of machine-to-machine systems.

Consequently, an extensive assessment of external validity for the experiments is not necessary.

4.5.5 Procedures

The readily testable nature of the operational hypotheses and in fact, their existence at all, suggests that within the research design, there is a clear wish to trace cause-and-effect relationships between defined variables and consequently, the research approach should be experimental.

Experimental and observational variables can be reliably well-defined and empirically measured based on objectively observed impressions within the simulation environment. Through the experimental approach, we aim to establish a statistically significant difference between control and experimental groups, through the collection, analysis and interpretation of the data.

The simulation environment ensures a high level of internal validity, allowing a high level of confidence in the results and that they can be attributed to the impact of the independent variable. External validity can be considered by generalising the findings further to other systems, outside of the simulation environment.

Pre-experimental designs are well suited to exploratory exercises but are prone to compromise by rival explanations for results. They are useful when the possibility for a high level of internal integrity is not there.

Quasi-experimental designs are a compromise between pre and true experimental designs but do not take full advantage of the simulation environment conditions available for the experiments.

Due to the rigorous controls and ability to rule out rival explanations, true experimental methods ensure a much higher internal integrity than other methods. They are also best suited to the testing of causal relationship hypotheses, as is required here.

For inferring causation with the highest level of certainty then, a true experimental design is both viable and desirable in this case. Specifically, a combination of n-group pre-test / post-test control-group and factorial design, while often difficult to statistically interpret and methodologically complex, provides a suitably flexible approach appropriate for this experimental analysis.

A true experimental approach offers the greatest potential for inferring causal relationships since we are able to carefully control experimental conditions and ensure the equivalent composition of experimental groups.

A factorial model involves two or more independent variables under study. Both the independent and interactive effects of these variables are studied.

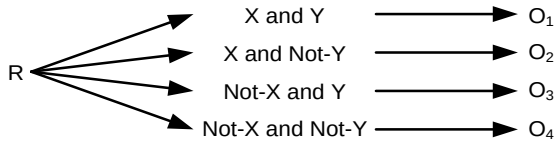


Figure 33 Basic Factorial experimental design (Williamson, Bow et al. 2002)

The experimental approach will be of n-group pre-test / post-test control-group and factorial design, declared explicitly for each independent experiment.

4.5.5.1 n-Group Pre-Test / Post-Test Control-Group

Group A	R	----------	O	----------	X	----------	O
Group B	R	----------	O	----------		----------	O

The main weakness of this research design is the internal validity is questioned from the interaction between such variables as selection and maturation or selection and testing. In the absence of randomisation, the possibility always exists that some critical difference, not reflected in the pre-test, is operating to contaminate the post-test data. For example, if the experimental group consists of volunteers, they may be more highly motivated, or if they happen to have a different experience background that affects how they interact with the experimental treatment - such factors rather than X by itself, may account for the differences (Montgomery 2012).

4.5.5.2 n-Group Post-Test Only Control-Group

Group A	R	----------	X	----------	O
Group B	R	----------		----------	O

The advantage here is the randomisation, so that any differences that appear in the post-test should be the result of the experimental variable rather than possible difference between the two groups to start with. This is the classical type of experimental design and has good internal validity. The external validity or generalisability of the study is limited by the possible effect of pre-testing. The Solomon Four-Group design accounts for this (Montgomery 2012).

4.5.5.3 Solomon Four-Group

A special case of the two-by-two factorial design and overcomes the external validity weakness in the design caused when pre-testing affects the subjects in such a way that they become sensitized to the experimental variable and they respond differently than the unpre-tested subjects (Montgomery 2012):

Group A	R	----------	O	----------	X	----------	O
Group B	R	----------	O	----------		----------	O
Group C	R	----------		----------	X	----------	O
Group D	R	----------		----------		----------	O

4.5.5.4 Steps

Based on the approach of Borg and Gall (Borg and Gall 1989), each Non-cooperative Programmable Open System Trust (NPOST) framework experiment will follow the procedural steps:

1. Calibrate environmental variables in accordance with the null and alternative hypotheses – apply the procedure to the trust functions, TF1.
2. Shuffle pseudo-random number generator in the simulation environment.
3. Initiate control and experimental groups according to established environmental variables, in step 1.
4. Generate (randomly or otherwise, depending on the hypotheses) initial state reputation profiles of significant sample sizes.
5. Administer measures of the independent variables to the experimental groups and none to the control group.
6. Generate final state reputation profile results through execution of the JOR algorithm in the simulation environment for all groups.
7. Repeat the experiment, with the number of repetitions dependent on the practicalities imposed by the experiment type, from step 3, creating a “batch” of experiments of similar configuration and motivation.
8. Compare the performance of the experimental and control groups on the post-test observational variables, and establish any statistical and observational significance.
9. Accept or refute the hypotheses in step 1.

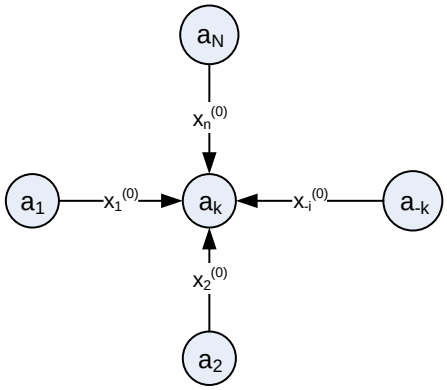


Figure 34 Initial state reputation profile

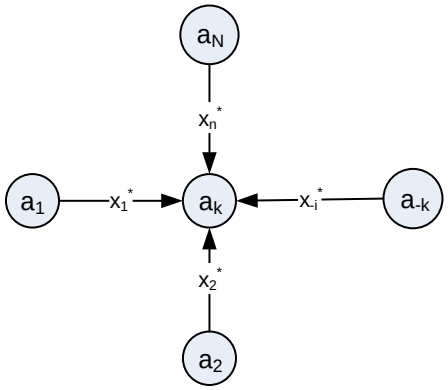


Figure 35 Final state equilibrium reputation profile

4.5.6 Statistical Analysis

Each independent experiment's data will be analysed in terms of some or all of the descriptive statistical measures:

- Mean;
- Median;
- STandard Deviation (STD);
- Variance;
- Balanced one-way ANalysis Of Variance confidence internals (ANOVA1) (MathWorks 2015) (Creswell 2003);
- Percentile rank – 25th and 75th percentiles (1st and 3rd quartiles);
- Range, and;
- Effect Size – mean and range percentage difference.

The reporting of *effect size* and *confidence intervals* is intended to be indicative of the practical significance of the findings. A confidence interval is an interval estimate of the range of upper and lower statistical values that are consistent with the observed data and are likely to contain the population mean. An effect size identifies the strength of the conclusions about group differences or the relationships among variables (Creswell 2003). Unless we experience divergence, we can expect the final reputation profile trust values to be convergent to values within the range of the initial reputation profile trust values. Intervals are descriptive of this change.

ANOVA1 performs balanced one-way ANOVA for comparing the means of two or more independent samples containing mutually independent observations. The function returns the p-value under the null hypothesis that all samples are drawn from populations with the same mean.

If the p-value is near zero, it casts doubt on the null hypothesis and suggests that at least one sample mean is significantly different from the other sample mean. Common significance levels are 0.05 and 0.01 (MathWorks 2015). The p-value describes the significance of the change of reputation profile trust values from control group to experimental group, and between initial and final reputation profiles within the same group.

The ANOVA1 test makes the following assumptions about the data being compared (MathWorks 2015):

- All sample populations are normally distributed;
- All sample populations have equal variance, and;
- All observations are mutually independent.

The ANOVA test is known to be robust with respect to modest violations of the first two assumptions. The validity of the test could be called in question on those grounds for these experiments.

We apply statistical measures to describe the change in trust profile from initial ($R_j(a_k)$) to final trust values ($R_j^*(a_k)$), as well as the differences between control and experimental group outcomes.

4.5.7 Reporting

Each Non-cooperative Programmable Open System Trust (NPOST) Framework experiment is reported in the following form:

NPOST Experiment Reference:	Unique experiment identifier class ($TF1.n$)		
Stamp:	Unique experiment identifier ($yyyymmddhhss$)		
Procedure:	From the experimental research methodology (4.3), most all experiments take the form of a 2-group pre-test / post-test control-group design: Group A R ---------- O ---------- O Group B R ---------- X_i ---------- O		
Type:	Experiments are of one or more of the types: 1. Scale; 2. Topology and Stability, and; 3. Environmental Factors. The experiment Type is consistent with the contribution and significance, and directly addresses the goals of this chapter (4.1.1).		
Operational Null hypothesis:	H_0 : Statement of the hypothesis that there is no significant difference between specified populations, any observed difference being due to sampling or experimental error, defined in terms of the experimental variables.		
Operational Alternate hypotheses:	H_1 : Statement of alternative hypothesis that sample observations are influenced by some non-random cause, defined in terms of the experimental variables.		
Experimental variables:	Framework: $X_1, X_2, X_3, X_4R_1, X_5 \; and \; X_6R_2$ Iterative Method (JOR) Algorithmics: $X_7, X_8R_3, X_9, X_{10} \; and \; X_{11}$ As defined (4.5.2.1).		
Extraneous variables:	Any undesirable variables that influence the relationship between the variables that the experiment is examining. These will be mainly concerned with experimental	**Moderating variables:**	Any strong contingent variable that has an effect on the independent variable-dependent variable relationship, which influences the general observed result of the experiment.

	instrumentation and materials (4.5.3).		
Control State:	Reference to the control experiment (Group A) (experiment reference or stamp) and experimental environmental variables being tested (X_n).		
Results			
Group B observational variables: $O_1, O_2, O_3, O_4 \; and \; O_5$			
Data Analysis			
Refute or accept H_0. Statistical analysis of the observational variables will be considered in terms of the statistical measures (4.5.6).			
Interpretation			
Interpretation in the form of a discussion of primarily of the observational variables relative to the contribution and significance goals of this chapter (4.1.1).			

All experimental data and figures available from 7.2 Appendix: NPOST Experimental Data and Figures

4.6 Results

4.6.1 Scale

4.6.1.1 Summary

These experiments are designed to determine two characteristic scale bounds of the NPOST framework simulation; *volume* and *stress*:

1. *Volume* – establish a capacity threshold for reasonable operation of the simulation, and;
2. *Stress* – determine the absolute capacity of the simulation before it no longer functions.

These are the constituent parts of the scale experiments. The volume capacity is used as an experimental control threshold for the all subsequent experiments.

These experiments contribute to supporting:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**

3. Proof of the robustness of the NPOST framework when scaled.

The scale experiments will determine the capacity of the framework in two dimensions; the number of nodes in the system and the range of initial reputation trust values. By varying these dimensions, we are able to establish the scale capacity of the simulation and its suitability as a framework for use in Emerging Systems.

The principal experimental variables are:

X_3	=	Initial Trust Value range ($q \leq x_i \leq r$)
X_5	=	Initial Reputation Profile dimension ($\dim(R_j(a_k))$)

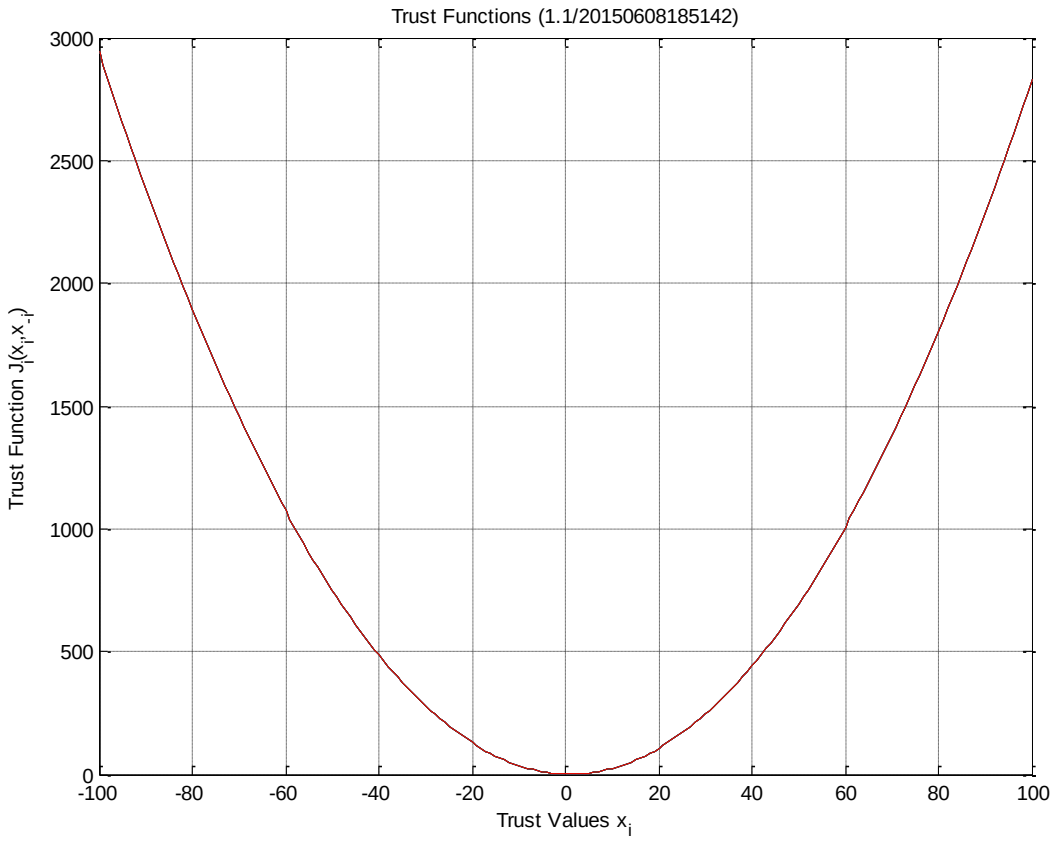


Figure 36 Typical Trust Function system of quadratic equations plot for Experiment Batch 1.1 and Experiment Batch 1.2

The system of Trust Functions describes a single line quadratic since the environmental factors are vertically (and horizontally) symmetric:

$$X_6R_2 \text{ with } e_{kj} = e_{lj}, \forall\, k, l, j$$

and:

$$\sum_{i=1}^{n} e_i = 1$$

4.6.1.2 Experiments

4.6.1.2.1 Experiment Batch 1.1

4.6.1.2.1.1 Operational Hypothesis

Ipsissima verba, the experimental operational hypotheses are:

H_0 : No significantly variation of O_1 and O_3 for variable X_5

H_1 : Significant variation of O_1 and O_3 for variable X_5

for experimental variable:

- $X_5 = \{10, 100, 1000, 10{,}000, 100{,}000, 1{,}000{,}000, 10{,}000{,}000\}$.

That is, that the Trust Values are expected to converge within the interaction upper-bound but the execution time and number of iterations of the algorithm is not expected to change significantly, despite an increased volume of nodes in the Emerging System.

4.6.1.2.1.2 Simulation Configuration

Initial Reputation Profile (X_4R_1) and environmental factors (X_6R_2) pseudo-randomly generated using a "multFibonacci" generator.

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	[0, 10] ∈ $\mathbb{Z}$
X_4R_1	=	multFibonacci
X_5	=	Experimental variable
X_6R_2	=	Horizontally and vertically symmetric, and uniformly pseudorandom.

Table 6 Simulation framework configuration for Experiment Batch 1.1

The algorithm was configured as follows:

X_7	=	$x_{-i}^{(S_k)}$
X_8R_3	=	Static responsive (1)
X_9	=	0.0001
X_{10}	=	100
X_{11}	=	1

Table 7 Simulation JOR algorithm configuration for Experiment Batch 1.1

4.6.1.2.1.3 Results

1,090 independent experiments were conducted. All nodes in the Emerging System responded for every iteration ($X_8R_3 = 1$), yielding the following results:

X_5 Initial Reputation Profile dimension	**O_1 Number of algorithm iterations (to 2 decimal places)**	**O_3 Computation real execution time (seconds to 9 decimal places)**	**Ratio (O_3:X_5) (to 9 decimal places)**
10	14.92	0.000421245	4.21245E-05
100	10.52	0.002909711	2.90971E-05
1000	13.32	0.05139402	5.1394E-05
10,000	14.24	2.931579509	0.000293158
100,000	19.80	364.1896626 (~6 minutes)	0.003641897
1,000,000	9.60	34,643.72607 (~9.6 hours)	0.034643726
10,000,000	Fail		

Table 8 Experiment Batch 1.1 results

4.6.1.2.1.4 Discussion

The volume threshold is determined under the condition that after $X_{10} = \sup(t) \leq 100$ iterations, $O_5 = convergence$ such that:

$$\left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq 0.0001$$

Every ten-fold increase in Initial Reputation Profile dimension, exerted a disproportionate increase in computational execution time. Between 10,000 and 100,000 nodes, computational time increased by a factor of over 124.

However, the number of iterations until convergence was achieved, does not appear influenced by the Initial Reputation Profile dimension. With $\dim(R_j(a_k) = 1{,}000{,}000$, the highest dimension

tested before the stress threshold was breached, converged in the lowest mean number of iterations of all the experiment batches. As should be expected from the mathematical analysis of the Trust Function, there should be no discernible correlation between these two experimental variables, but the computational approach of the simulation could have exerted some influence.

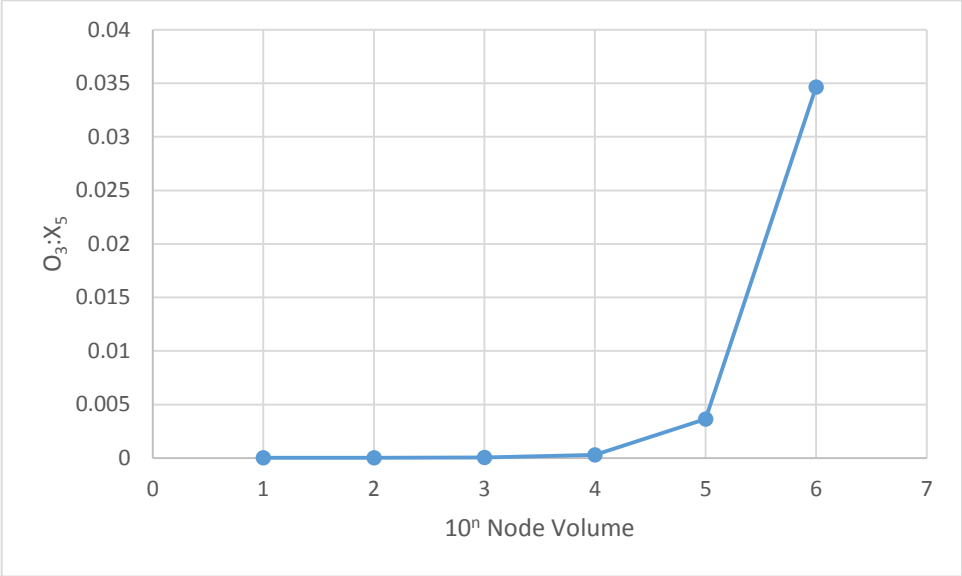


Figure 37 Ratio of computation real execution time against Reputation Profile Dimension incremental change plot

In all experiments, convergence was achieved except at the stress threshold level, $\dim(R_j(a_k) = 10{,}000{,}000$ where the experimental instruments failed to respond.

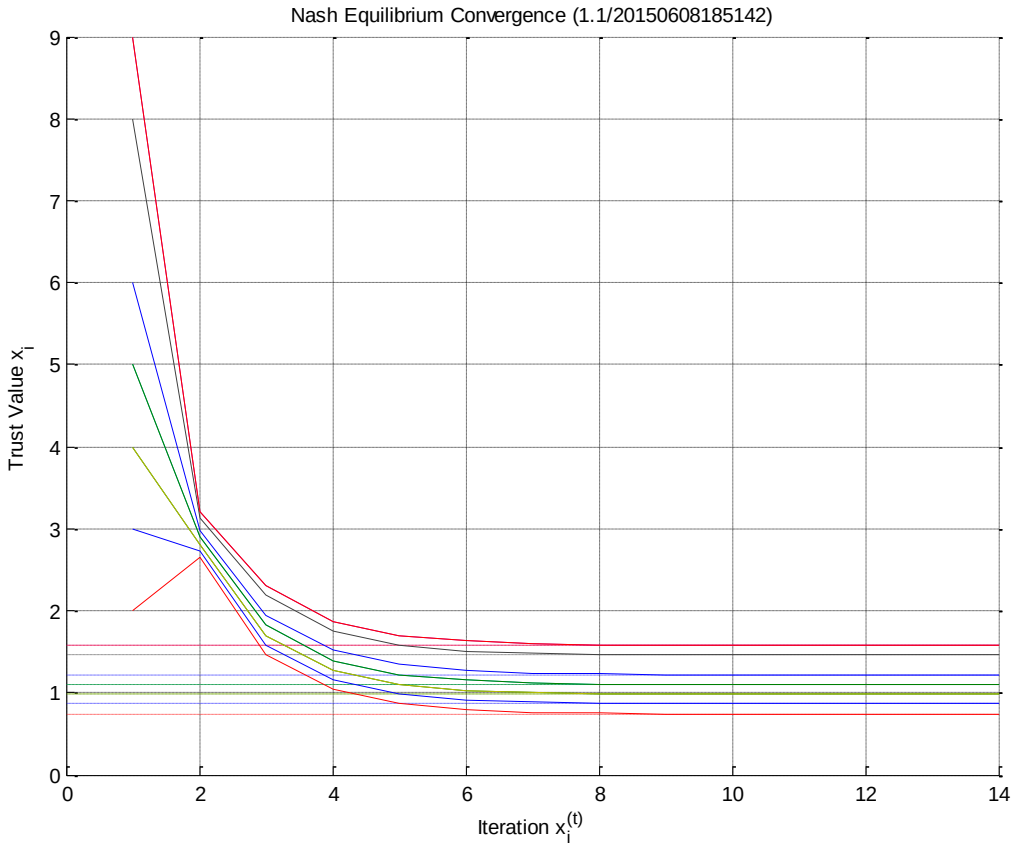


Figure 38 Nash Equilibrium convergence against iteration plot

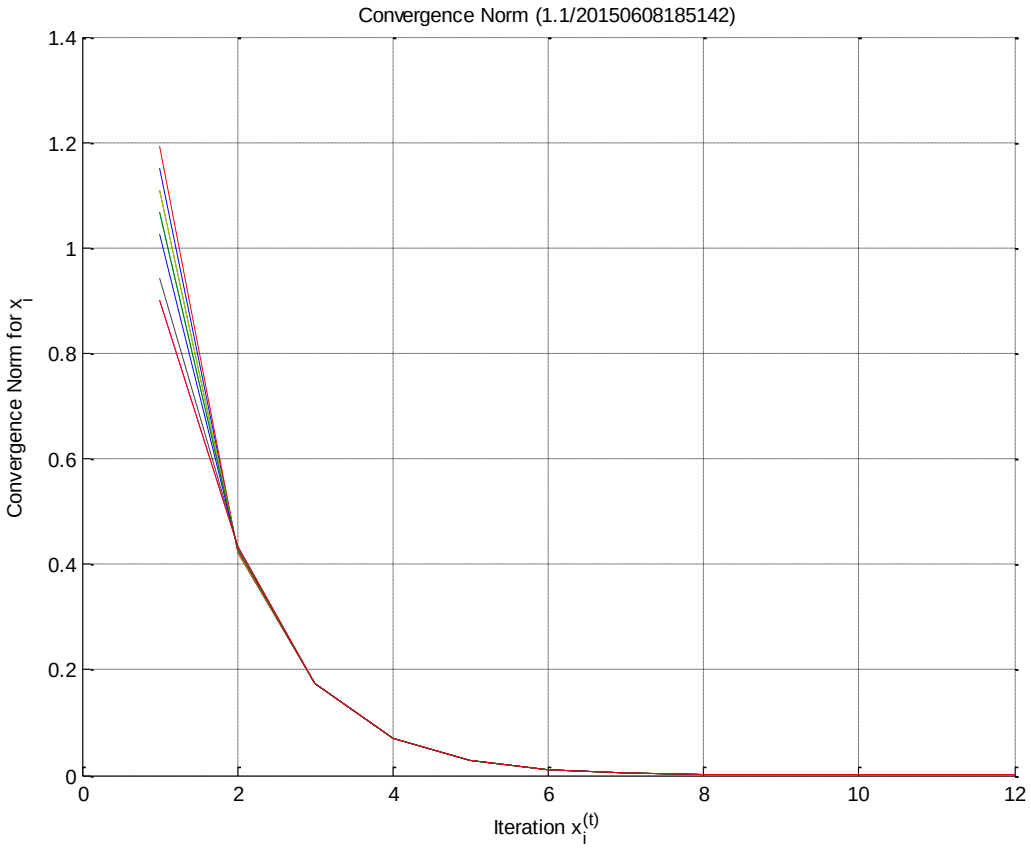


Figure 39 Convergence norm of Trust Values between computation iterations plot

Experiments of the 1.1 batch reference are the control for the remaining dimensional scale experiments. For experiment 1.1/20150608185142, Trust Values converged quickly (within two iterations) and uniformly to a Nash Equilibrium stability. There is an arbitrarily small disparity between convergence norms between nodes, comparable to a natural logarithmic decay to stability.

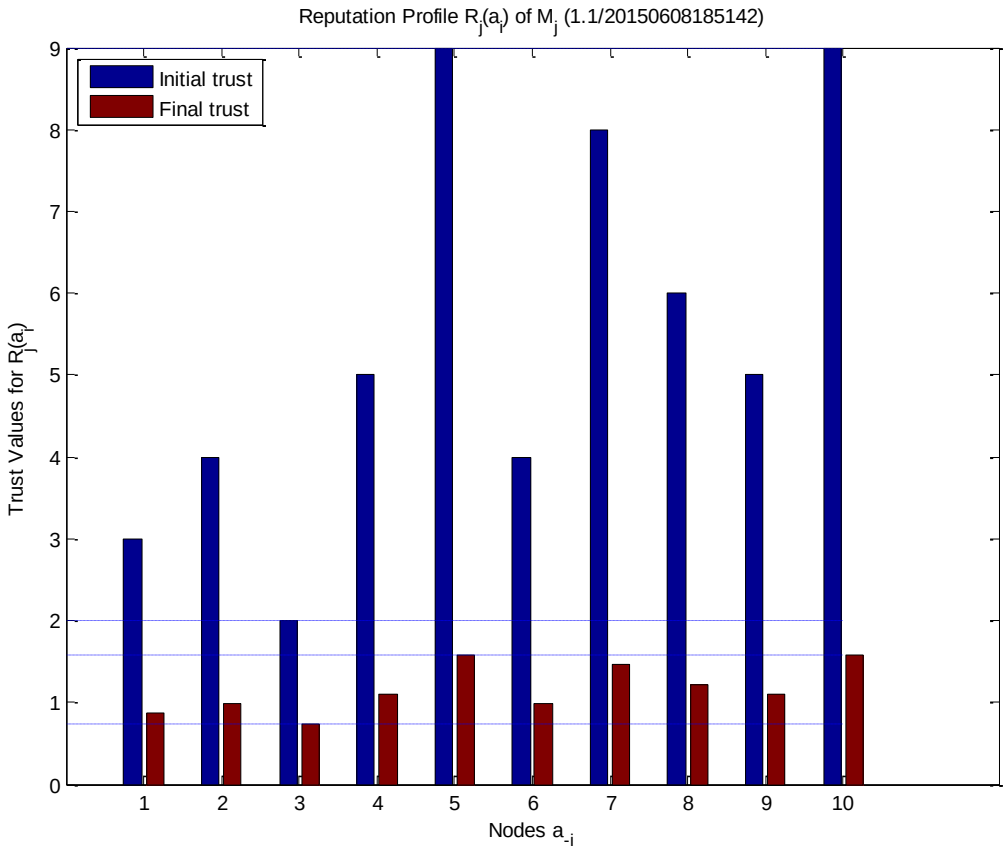


Figure 40 Initial Reputation Profile against final Reputation Profile after simulation analysis

For experiment 1.1/20150608185142, the initial Trust Value norm was 7 before simulation with a range from 2 to 9, with final Trust Value norm of 0.837 (to 3 decimal places) and range 0.743 to 1.580, a significant 88.04% change. The mean for the scale 1.1 experiment reference is a ~65.30% change. ANOVA1 indicates a significant variation in mean and range for initial and final Trust Values.

At the point where the simulation is no longer able to respond reasonably to the scale of the experiment, we consider that the nodes in the system are no longer able to meet the sufficient conditions for Nash Equilibrium, specifically, the agents' execution was flawed (3.7.4 Sufficient Conditions).

The capacity of the simulation was limited by the analysis functions used to report the results of the experiments. The experimental instruments have introduced extraneous variables into the experiment.

There is an inherent overhead to observing the experiments that needs to be reduced in order for the experiment to complete that is not directly required for the simulation to run. By restricting the

observational variable set, statistical analysis measures, matrix analysis and graphical generation of results, the simulation performed better but less observational data was able to be collected. Matrix analysis for instance, determines for the system matrix properties:

- Symmetric;
- Full rank;
- Diagonally dominant;
- Strictly diagonally dominant;
- Positive definite, and;
- All non-zero diagonal.

This requires the calculation of eigenvalues which can add substantial effort to the computation (0

Spectral Radius). The default Matlab algorithm depends on the properties the matrix, but generally uses the QZ algorithm otherwise Cholesky factorization (Chadwick and Bindel 2015). Invariably, the two algorithms return the same result. The QZ algorithm can be more stable for certain problems, such as those involving badly conditioned matrices (MathWorks 2015). The selection of algorithm does not directly affect the result of the simulation experiment but it could potentially, influence the performance of the result analysis.

The main cause of the performance degradation was the need to store all of the results for every iteration of the experiment, in computation memory (RAM). This resulted in a matrix of results of $X_5 \times number\ of\ interations$ in size.

- For large $X_5 \geq 10{,}000$, the simulation was not able to store the result matrix and perform calculations on it so observational variables had to be restricted.
- For larger $X_5 \geq 10{,}000{,}000$ the NPOST simulation failed – this is deemed the stress threshold.

It would be possible to refine these results to determine exactly (to a single node), when the sufficient conditions for Nash Equilibrium are breached, but these results are sufficient for the purposes of these experiments as they are representative of an Emerging System in principle.

The convergence condition was not breached in any experiment. Convergence was consistently achieved, except at the stress threshold.

4.6.1.2.1.5 Conclusion

From the experimental results the conclusion is drawn that:

> *For Experiment Batch: 1.1, we must refute H_0 and accept H_1. While it is not the case that the volume of nodes in the System significantly alters the number of iterations before convergence of the simulation, it increases the computational time, disproportionally to the increase in volume of nodes.*

4.6.1.2.2 Experiment Batch 1.2

4.6.1.2.2.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : No significantly variation of O_1 and O_3 for variable X_3

H_1 : Significantly variation of O_1 and O_3 for variable X_3

for experimental variables:

- $X_3[min, max] = \{[-10, 10], [-100, 100], [-1{,}000, 1{,}000], [-10{,}000, 10{,}000]\}$ with $X_5 = 1{,}000$ and $X_4R_1 \in \mathbb{Z}$,
- $X_3[min, max] = \{[-1{,}000, 1{,}000], [-10{,}000, 10{,}000]\}$ with $X_5 = 1{,}000$ and $X_4R_1 \in \mathbb{Q}$, and;
- $X_3[min, max] = [-1{,}000, 1{,}000]$ with $X_5 = \{1{,}000, 10{,}000\}$ and $X_4R_1 \in \mathbb{Z}$.

It is expected that the Trust Values converge within the interaction upper-bound but the execution time and number of iterations of the algorithm is not expected to vary significantly, despite an increased range and variable sign of initial Trust Values.

4.6.1.2.2.2 Simulation Configuration

Again, Initial Reputation Profile (X_4R_1) and environmental factors (X_6R_2) pseudo-randomly generated using a “multFibonacci” generator.

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	Experimental variable
X_4R_1	=	multFibonacci
X_5	=	Experimental variable
X_6R_2	=	Horizontally and vertically symmetric, and uniformly pseudorandom.

Table 9 Simulation framework configuration for Experiment Batch 1.2

The algorithm was configured similarly to Experiment Batch 1.1 for control:

X_7	=	$x_{-i}^{(S_k)}$
X_8R_3	=	Static responsive (1)
X_9	=	0.0001
X_{10}	=	100
X_{11}	=	1

Table 10 Simulation JOR algorithm configuration for Experiment Batch 1.2

4.6.1.2.2.3 Results

1,620 independent scale experiments were conducted. All nodes in the Emerging System responded for every iteration ($X_8R_3 = 1$).

X_3 Initial Trust Value range ($X_4R_1 \in \mathbb{Z}$)	O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)	Ratio (O_3:X_5) (to 9 decimal places)
-10 to 10	6.45	0.024830175	0.003849640
-100 to 100	8.65	0.032391006	0.003744625
-1,000 to 1,000	10.10	0.039990661	0.003959471
-10,000 to 10,000	15.70	0.060872745	0.003877245

Table 11 Experiment Batch 1.2 for integer initial Trust Value range

X_3 Initial Trust Value range ($X_4R_1 \in \mathbb{Q}$)	O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)	X_3 Initial Trust Value range O_3 delta ($X_4R_1 \in \mathbb{Z}$) (to 9 decimal places)
-1,000 to 1,000	11.80	0.046807060	0.006816399
-10,000 to 10,000	16.05	0.063363608	0.002490863

Table 12 Experiment Batch 1.2 results for rational initial Trust Value range against computation real execution time for integer initial Trust Value range

X_3 Initial Trust Value range ($X_4R_1 \in \mathbb{Z}$)	O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)	X_5 Initial Reputation Profile dimension (to 9 decimal places)
-1,000 to 1,000	9.90	2.413678627	1,000
-1,000 to 1,000	8.90	117.495331100 (1.96 minutes)	10,000

Table 13 Experiment Batch 1.2 results for integer initial Trust Value range against computation real execution time higher Initial Reputation Profile dimensions

4.6.1.2.2.4 Discussion

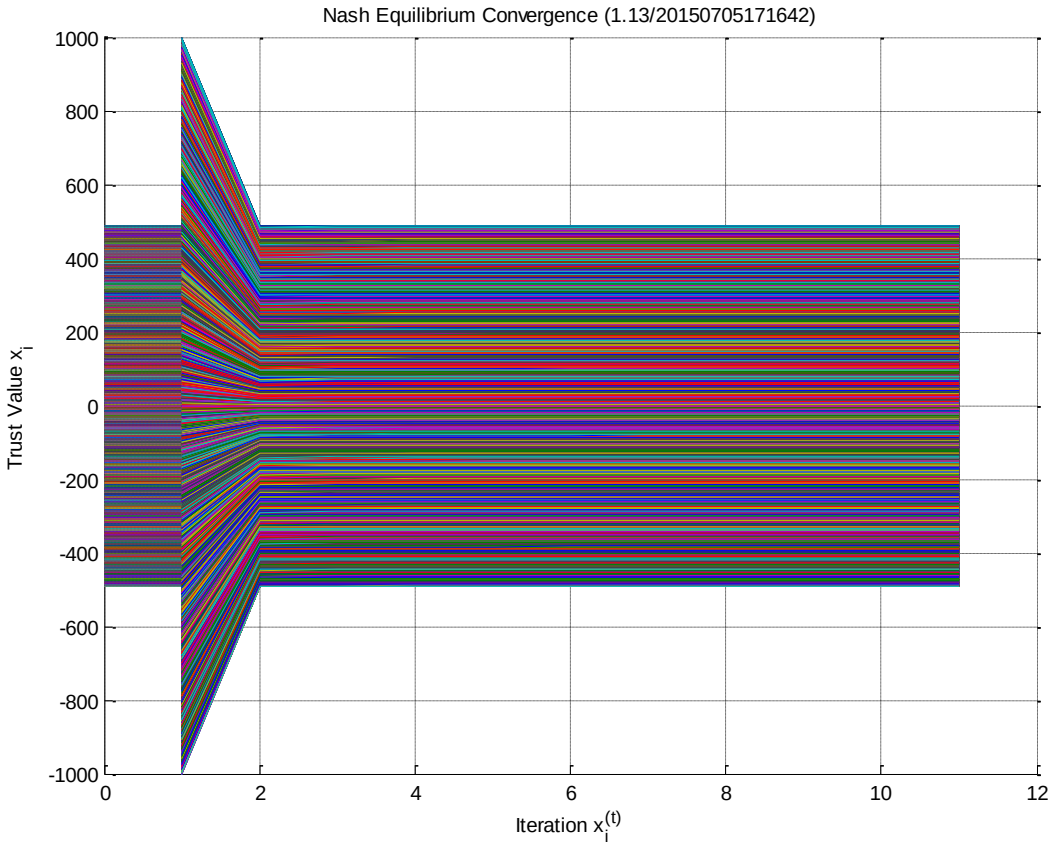


Figure 41 Nash Equilibrium convergence against iterations plot

Convergence to Nash Equilibrium is similar to the erstwhile plot (Figure 38), but clearly with far greater numbers of nodes represented and a much larger range of Initial Trust values. However, the number of iterations is fewer – 11 in this case and 14 in Figure 38. At this scale and above, this graphical representation because increasingly general.

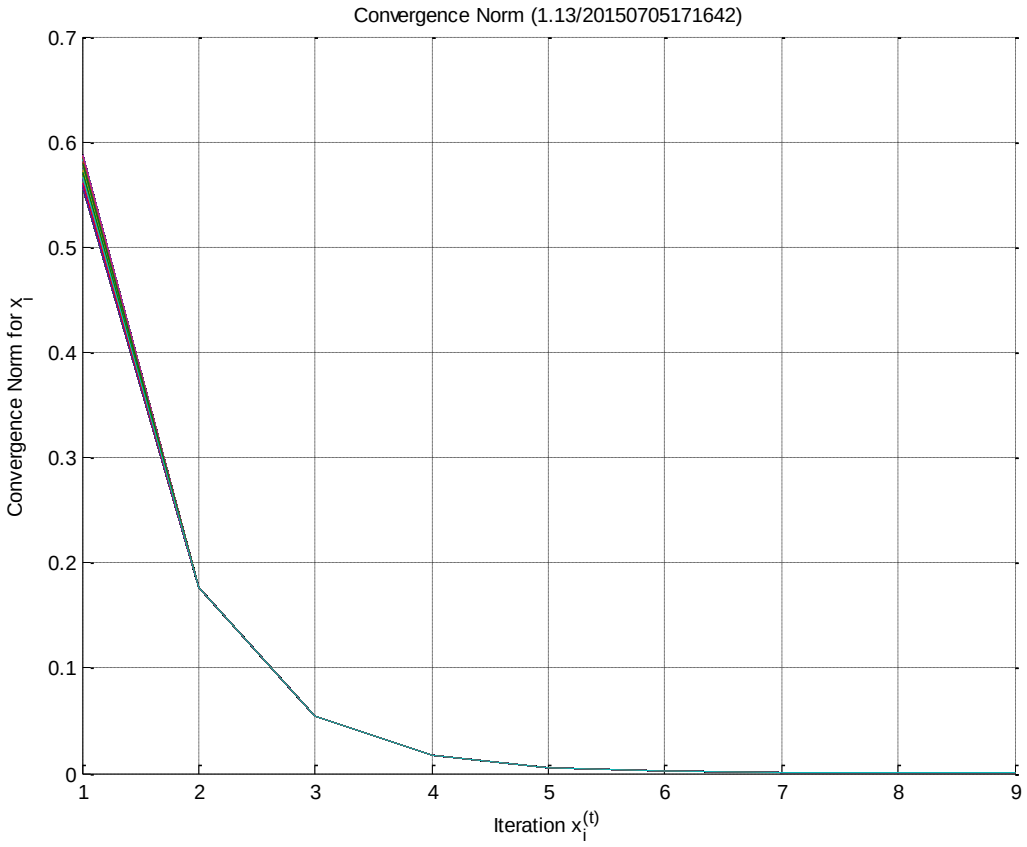


Figure 42 Convergence norm of Trust Values between computation iterations

The convergence norm for higher dimension and domain range experiments retains the similar logarithmic decay profile of Figure 39. All convergence is similar after two iterations.

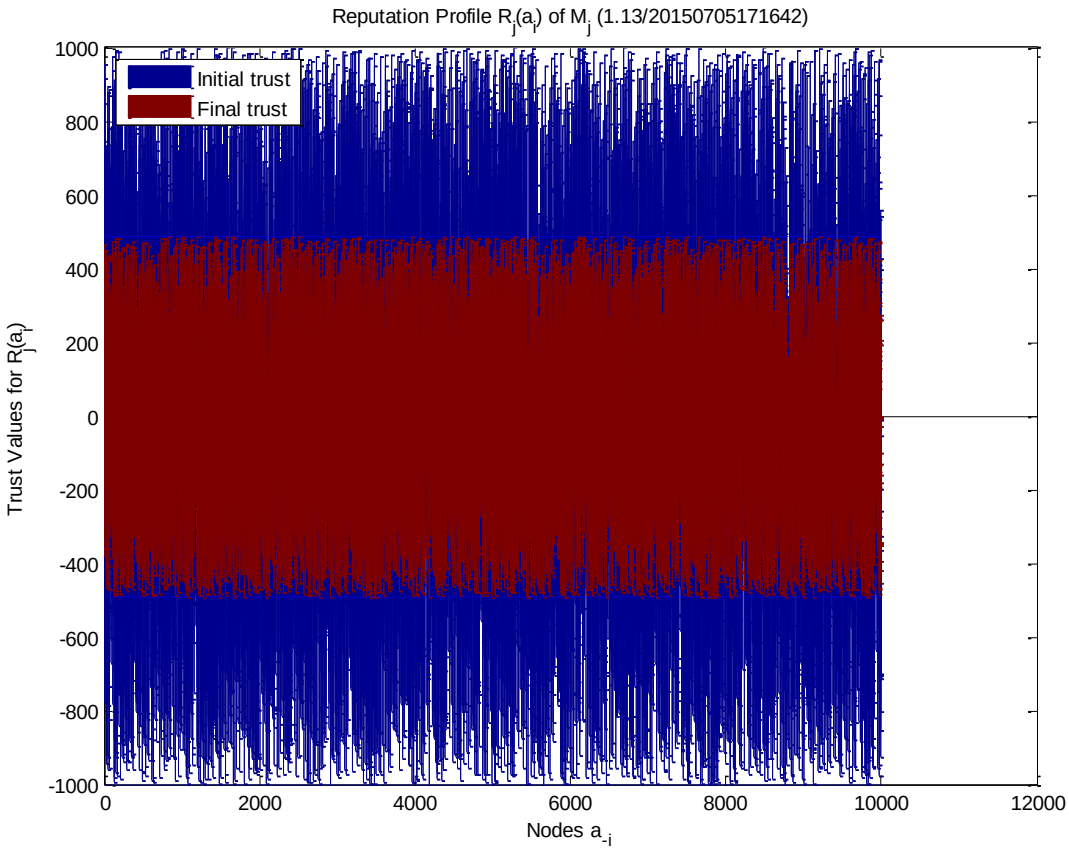


Figure 43 Initial Reputation Profile against final Reputation Profile after simulation analysis

As with the Nash Equilibrium Convergence plot, Initial Reputation against Final Reputation Profile because increasingly difficult to discern individual values as the experiments scale.

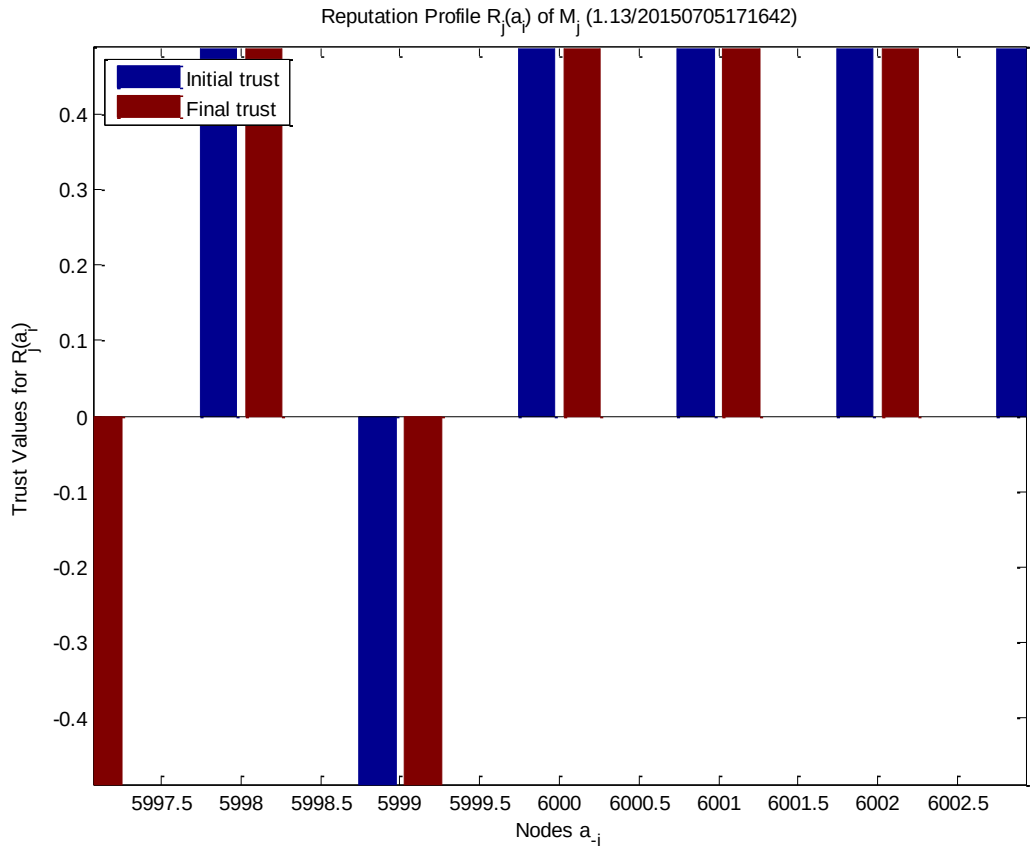


Figure 44 Magnified Initial Reputation Profile against final Reputation Profile after simulation analysis

For experiment 1.13/20150705171642, the initial Trust Value norm was exactly 2,000 before simulation with a range from -1,000 to 1,000, with final Trust Value norm of 979.331 (to 3 decimal places) and range -491.676 to 487.655, a significant 51.33% change. The mean for the scale 1.13 experiment reference is a ~64.360% change. ANOVA1 indicates a significant variation in mean and range for initial and final Trust Values. These results are consistent with the 1.1 batch of experiments.

There is no observable difference between executions of the simulation with different integer Trust Value ranges. The ratio of iterations to real execution time remains consistent to over three decimal places.

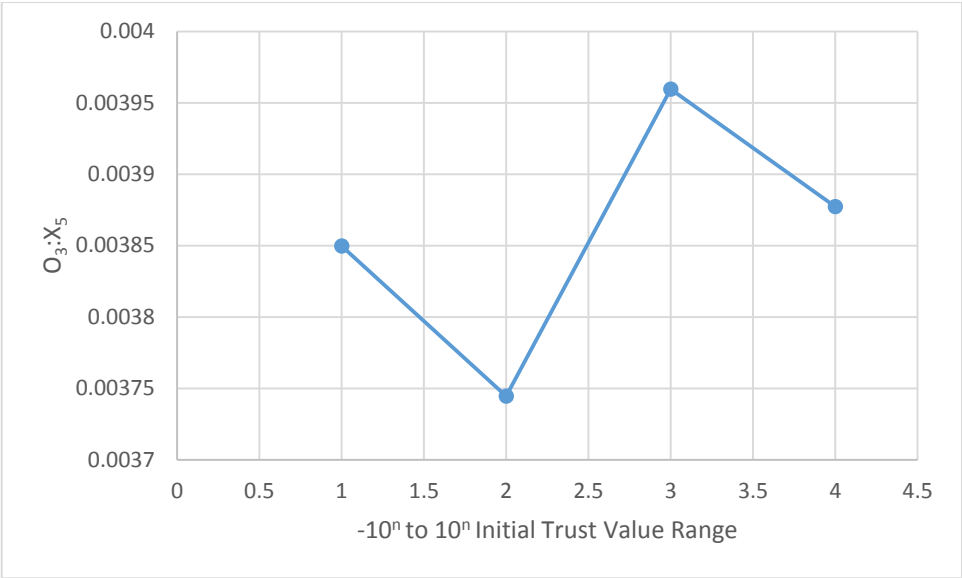


Figure 45 Ratio of computation real execution time against initial Trust Value range incremental change plot

There is no significant observable difference between executions of the simulation with integer or rational initial Trust Values and ranges. The difference in iterations is on mean approximately one and the real time execution time is similar to three decimal places.

4.6.1.2.2.5 Conclusion

The experimental results are adduced to conclude the following:

> *For Experiment Batch 1.2, we must accept H_0 and refute H_1. There is no evidence to support the hypothesis that either type or range of initial Trust Value influences the computation time of the simulation or number of iterations before convergence is reached.*

4.6.1.3 Conclusion

These experiments contribute to:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**
3. **Proof of the robustness of the NPOST framework when scaled.**

Dependent on the specific application of the framework and the sample size necessary to support it (4.5.1.1 Sample Size), there is sufficient capacity in the simulation to establish meaningful results for considerable volume and range Emerging Systems.

For Emerging Systems that require almost real-time trust propagation data, processed within milliseconds so that it is available virtually immediately (O_3 < 0.01 seconds) as response to the process from which it was requested, from the experimental analysis, factors that should be considered include:

- Size of X_5 – the local propagation of trust contiguous to other Trust Spaces, and;
- Implementation instrument capacity.

An Emerging System established over a longer period of time (O_3 > 6 minutes or O_3 > 9.6 hours) does not require such refinements (though should do, as Rahman, Maksud-Ul-Alam et al. (2015) assert, as a matter of "best-practice").

These results are consistent between mathematically convergent Trust Functions, $TFn_i(x_i, \mathbf{x_{-i}})$.

Consequently, we fix 1,000 as a reasonable control volume threshold of nodes in the System for all experiments and consider the range and numerical type of trust values as inconsequential, and therefore we arbitrarily adopt $X_3[min, max] = [-100, 100] \in \mathbb{Z}$ as the control for all experiments.

This experimental analysis supports the conjecture that the NPOST framework scales suitably, and is robust enough to support different and changing volume Emerging System nodes and initial Trust Value ranges for different practical applications.

4.6.2 Topology and Stability

4.6.2.1 Summary

Topology experiments were designed to test the response of the simulation when an Emerging System is partitioned. The experiments test the convergence of the system under *topological volatility* with alternative *Stability Strategies*. Topological volatility occurs when nodes in the system fail to respond to requests for Trust Values and do not contribute to convergence (or potentially, divergence) persistently. It is possible that one or many nodes may join the System and then depart multiple times between iterations partitioning the initial System either temporarily or permanently, requiring the simulation to compensate for the changes. Stability Strategies define how the simulation responds to these changes.

These experiments contribute to:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**
3. **Proof of the robustness of the NPOST framework when partitioned.**

With $X_5 = 1{,}000$ as a reasonable control volume threshold of nodes in the System and considering the range and numerical type of trust values as inconsequential, and therefore arbitrarily adopting $X_3[min, max] = [-100, 100] \in \mathbb{Z}$ from the scale experiments, these experimental variables are fixed as control for the topology and stability experiments, unless explicitly stated otherwise.

The principal experimental variables are:

X_7	$=$	Stability strategy and readjustment scheme (termination of node after s unresponsive requests and reinstatement criteria – accumulative / consecutive and correction)
X_8R_3	$=$	Determinant of a node's availability in the system for an iteration, R_3: • static non-responsive, • static responsive, and; • uniformly pseudorandom
X_9	$=$	Convergence condition or error tolerance: ($\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon$ for all Δx or for any Δx)

For the scale experiments, a stability strategy was not employed since all nodes in the system were configured to respond to all requests for updated Trust Values.

4.6.2.2 Experiments

4.6.2.2.1 Experiment Batch 1.3

4.6.2.2.1.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : O_1 will decrease as more nodes fail to respond (X_8R_3) and O_3 will not significantly vary

H_1 : O_1 will increase or remain the same as more nodes fail to respond (X_8R_3) and O_3 will not significantly vary

for experimental variables:

- X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $=$
  $\{[0.0, 100.0, 0.0], [0.1, 99.9, 0.0], [1.0, 99.0, 0.0], [10.0, 90.0, 0.0], [25.0, 75.0, 0.0], [50.0, 50.0, 0.0],$
  $[75.0, 25.0, 0.0], [90.0, 10.0, 0.0], [100.0, 0.0, 0.0]\}$
  with $X_9 \leq 0.0001$ for all Δx and for any Δx;
- X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $=$
  $\{[0.0, 99.9, 0.1], [0.0, 99.0, 0.1], [0.0, 90.0, 10.0], [0.0, 75.0, 25.0], [0.0, 50.0, 50.0],$
  $[0.0, 25.0, 75.0], [0.0, 10.0, 90.0], [0.0, 0.0, 100.0]\}$
  with $X_9 \leq 0.0001$ for all Δx, and;
- X_7, termination of node after s iterations and reinstatement criteria.

It is expected that the more nodes that fail to respond in the System, will decrease the number of iterations before the convergence condition is reached. This is due to the Stability Strategy producing zero Trust Value differences between iterations for non-responsive nodes.

The Stability Strategy and Readjustment Scheme (X_7) reuses the previous iteration Trust Value response a node admitted before it became unresponsive, until convergence is reached or the node becomes responsive again.

4.6.2.2.1.2 Simulation Configuration

Initial Reputation Profile (X_4R_1) and environmental factors (X_6R_2) pseudo-randomly generated.

The framework was configured as follows:

X_1	$=$	TF1
X_2	$=$	1
X_3	$=$	[-100, 100] $\in \mathbb{Z}$
X_4R_1	$=$	multFibonacci
X_5	$=$	1,000
X_6R_2	$=$	Horizontally and vertically symmetric, and uniformly pseudorandom.

Table 14 Simulation framework configuration for Experiment Batch 1.3

The algorithm was configured as follows:

X_7	=	Experimental variable
$X_8 R_3$	=	Experimental variable
X_9	=	0.0001
X_{10}	=	100
X_{11}	=	1

Table 15 Simulation JOR algorithm configuration for Experiment Batch 1.3

4.6.2.2.1.3 Results

3,600 independent experiments were conducted, yielding the following results:

X_8R_3			X_9 Convergence condition	O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)
0	**1**	**2**			
0.0	100.0	0.0	Arbitrarily, All	8.65 (control from scale experiments)	0.032391006 (control from scale experiments)
0.1	99.9	0.0	All	9.45	0.036313577
0.1	99.9	0.0	Any	1.00	0.003733117
1.0	99.0	0.0	All	9.45	0.035205079
1.0	99.0	0.0	Any	1.00	0.003746814
10.0	90.0	0.0	All	8.35	0.030856273
10.0	90.0	0.0	Any	1.00	0.003575341
25.0	75.0	0.0	All	9.10	0.030481629
25.0	75.0	0.0	Any	1.00	0.003393782
50.0	50.0	0.0	All	6.45	0.019271858
50.0	50.0	0.0	Any	1.00	0.003329736
75.0	25.0	0.0	All	4.65	0.012128874
75.0	25.0	0.0	Any	1.00	0.002804717
90.0	10.0	0.0	All	4.45	0.010670443
90.0	10.0	0.0	Any	1.00	0.002758020
100.0	0.0	0.0	All	1.00	0.002228778
100.0	0.0	0.0	Any	1.00	0.002322508

Table 16 Results for Experiment Batch 1.3

2,400 additional independent experiments were conducted, with the following results:

X_8R_3			O_1 Number of algorithm iterations with random X_8R_3 (to 2 decimal places)	O_3 Computation real execution time with random X_8R_3 (seconds to 9 decimal places)	O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)
0	**1**	**2**				
0.1	99.9	0.0	9.60	0.036952824	9.45	0.036313577
1.0	99.0	0.0	8.15	0.029947473	9.45	0.035205079
10.0	90.0	0.0	7.30	0.025748900	8.35	0.030856273
25.0	75.0	0.0	6.10	0.019530156	9.10	0.030481629
50.0	50.0	0.0	5.40	0.015262603	6.45	0.019271858
75.0	25.0	0.0	4.55	0.011675987	4.65	0.012128874
90.0	10.0	0.0	3.40	0.008034167	4.45	0.010670443
99.0	1.0	0.0	3.45	0.008210179	3.40	0.008269469
99.9	0.1	0.0	2.00	0.004581090	2.00	0.004622675

Table 17 Results for Experiment Batch 1.3 with random X_8R_3

A further 1,800 independent experiments were conducted, with the following results:

X_8R_3 Determinant of node availability			O_1 Number of algorithm iterations with random X_8R_3 (to 2 decimal places)	O_3 Computation real execution time with random X_8R_3 (seconds to 9 decimal places)
0	1	2		
0.0	100.0	0.0	8.65	0.032391006
0.1	99.9	0.0	9.60	0.036952824
0.0	99.9	0.1	8.85	0.038334526
1.0	99.0	0.0	8.15	0.029947473
0.0	99.0	1.0	14.05	0.061038759
10.0	90.0	0.0	7.30	0.025748900
0.0	90.0	10.0	19.65	0.084838746
25.0	75.0	0.0	6.10	0.019530156
0.0	75.0	25.0	22.95	0.094951722
50.0	50.0	0.0	5.40	0.015262603
0.0	50.0	50.0	29.50	0.115535242
75.0	25.0	0.0	4.55	0.011675987
0.0	25.0	75.0	30.35	0.109903097
90.0	10.0	0.0	3.40	0.008034167
0.0	10.0	90.0	31.45	0.108081945
99.0	1.0	0.0	3.45	0.008210179
0.0	1.0	99.0	31.35	0.105304399
99.9	0.1	0.0	2.00	0.004581090
0.0	0.1	99.9	30.30	0.100897678
100.00	0.0	0.0	1.00	0.002228778
0.0	0.0	100.0	34.35	0.116528544

Table 18 Results for Experiment Batch 1.3 with pseudorandom, X_8R_3 System positioning and response

4.6.2.2.1.4 Discussion

A pertinent consideration for these experiments is the implementation of the Convergence Condition (X_9) in the simulation. The simulation can enforce the condition in two ways:

$\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon = 0.0001$, for *all* Δx or for *any* Δx. That is, the Convergence Threshold can be achieved when all Trust Values between iterations, are below the conditional value or any one.

The results show that with a consistent execution time, for the Convergence Condition $\left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon = 0.0001$, for any Δx there is always exactly one iteration before convergence is reached. This can be explained by a non-responsive achieving the Convergence Threshold in a single iteration, since its value remains the same between iterations.

For the Convergence Condition $\left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon = 0.0001$, for all Δx, the percentage of nodes designated unresponsive increases, the number of iterations to convergence decreases. The choice of ε in these cases is arbitrary since convergence is actually being established with $\varepsilon = 0$ for non-responsive nodes.

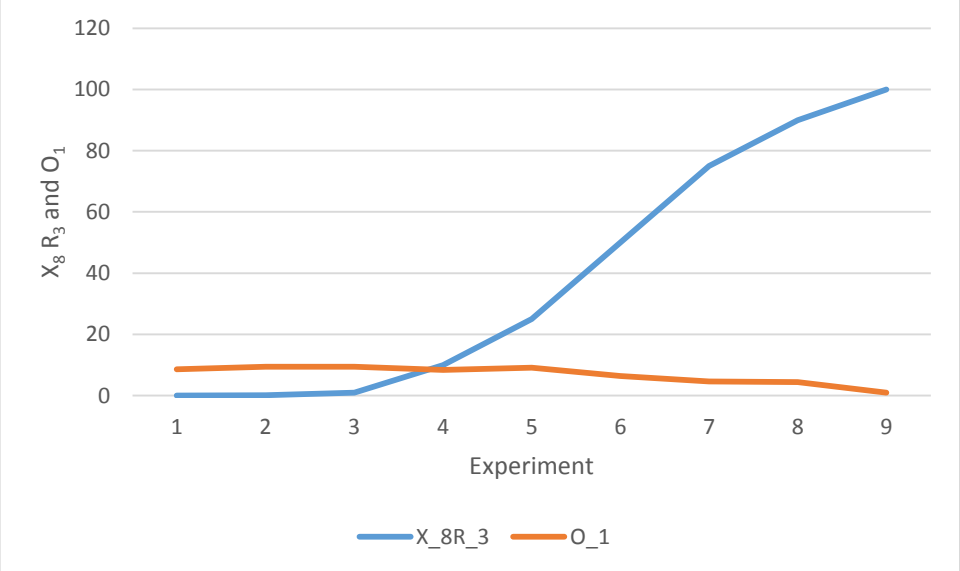


Figure 46 Comparison of unresponsive node percentage to simulation iteration count plot

The mean iteration execution time of 0.003175623 seconds with a standard deviation of 0.000599065, suggests that there was very little difference between the performances of how each experiment executed. This is consistent with the scale experiment results.

At $X_8R_3 = \{[0,100,0],[100,0,0]\}$, the results are similar for all Δx as for any Δx configurations.

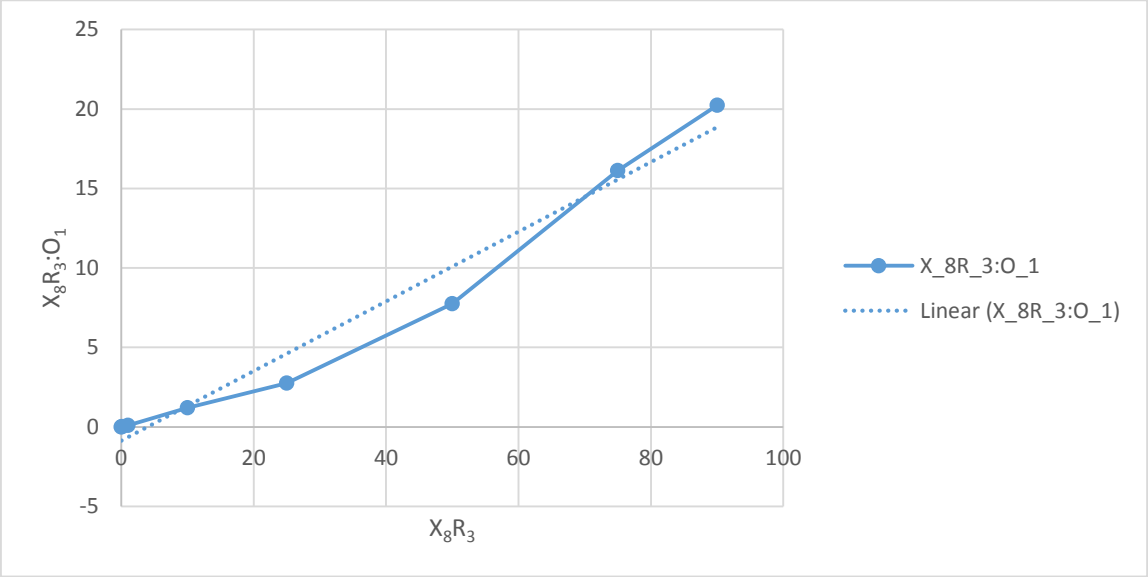


Figure 47 Comparison of unresponsive node percentage to ratio of unresponsive node percentage and iteration count plot

There can be observed a linearly increasing association between the percentage of nodes that do not respond in the system, and the ratio of percentage to number of iterations until convergence. This is again, indicative of decreasing iteration count as a result of unresponsive nodes converging immediately.

For previous experiments in this batch, the non-response nodes were configured to be the first nodes interrogated for their Trust Values. To establish if this exerts influence on the number of iterations the simulation completes before convergence is reached, we consider the case where the nodes are randomly allocated for non-responsiveness.

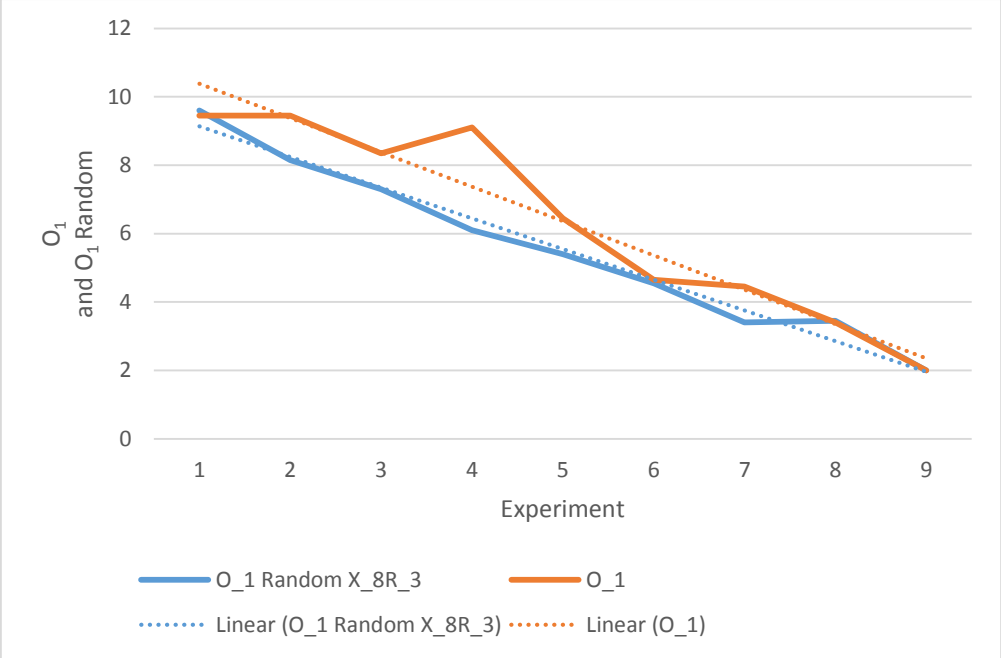


Figure 48 Comparison of unresponsive node percentage to simulation iteration count plot

The results indicate that the number of iterations reduces in this case since it is more likely that some convergence has occurred in previous iterations with persistent responsive nodes, before an unresponsive node is encountered. The experimental results support this, as attested by Figure 48. The mean iteration count with random allocation is 5.55 (with standard deviation of 2.34) and without, 6.37 (with standard deviation of 2.69).

When randomly responsive nodes are introduced into the System, that is, nodes that have a uniformly distributed chance of being responsive or non-responsive to Trust Value requests, we observe a linearly increasing iteration count, compared to the same percentage of nodes allocated as wholly non-responsive. The randomly responsive nodes are randomly allocated in the System, with previous iteration Trust Values adopted when non-responsive stability strategy and all nodes are subject to an "all" convergence condition.

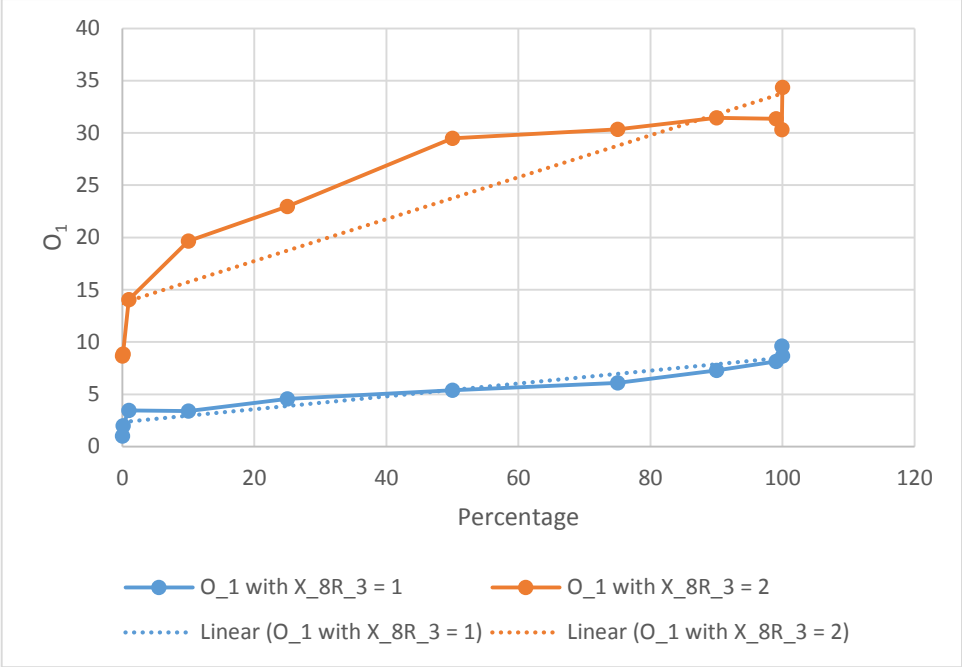


Figure 49 Comparison of percentage responsive against randomly responsive nodes, iteration count to convergence

As similar result is observed for convergence execution times with the mean iteration time remaining consistent.

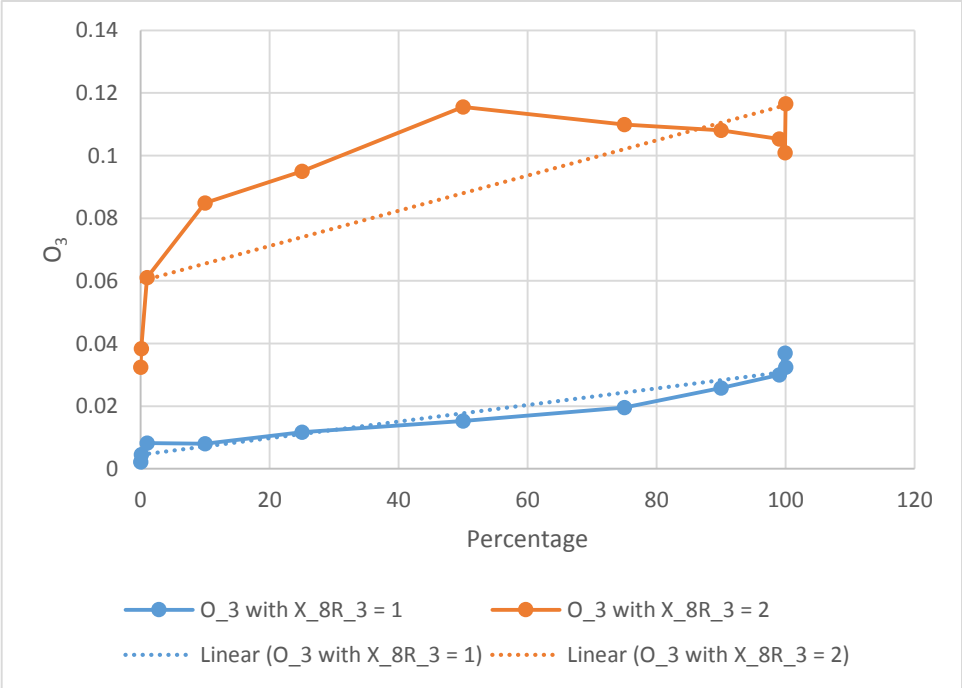


Figure 50 Comparison of percentage responsive against randomly responsive nodes, completion time to convergence

Maximum $O_1 = 68$ and $O_3 = 0.26638046$ (means being $O_1 = 25.28$ and $O_3 = 0.0935541466$) which are substantially higher results than have been observed previously, in the case where the percentage of pseudorandom ($X_8 R_3 = 2$) was allocated at 50. Both results occurred in the same experiment and produced an extremely erratic convergence, with some nodes failing to respond up to 45 times during the simulation, but the simulation continued to achieve convergence.

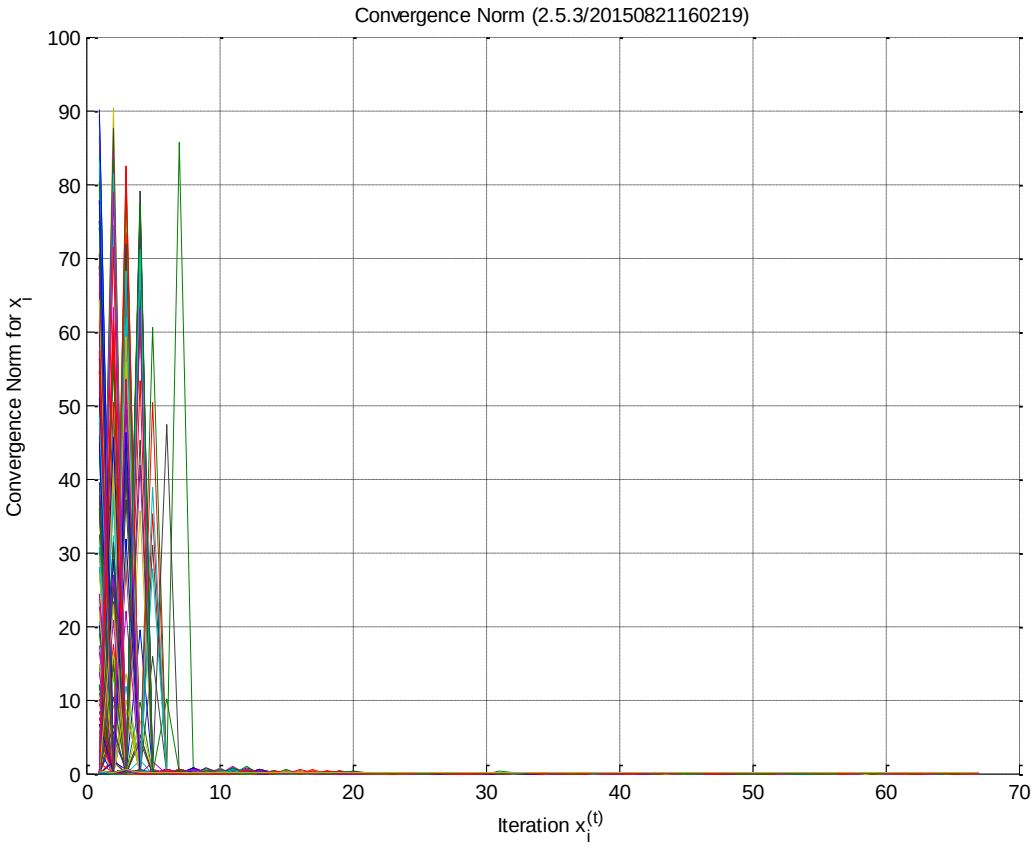


Figure 51 Convergence norm plot for 50% allocated randomly responsive nodes

The convergence norm for the experiments with randomly non-responsive nodes describe a mercurial convergence progression resulting in protracted total convergence time and increased iterations to convergence. This is a significantly different plot from the results observed in Figure 42, where convergence is achieved in a linearly decreasing fashion for all nodes. Convergence is substantially more volatile for randomly responsive nodes and is drawn out over more iterations.

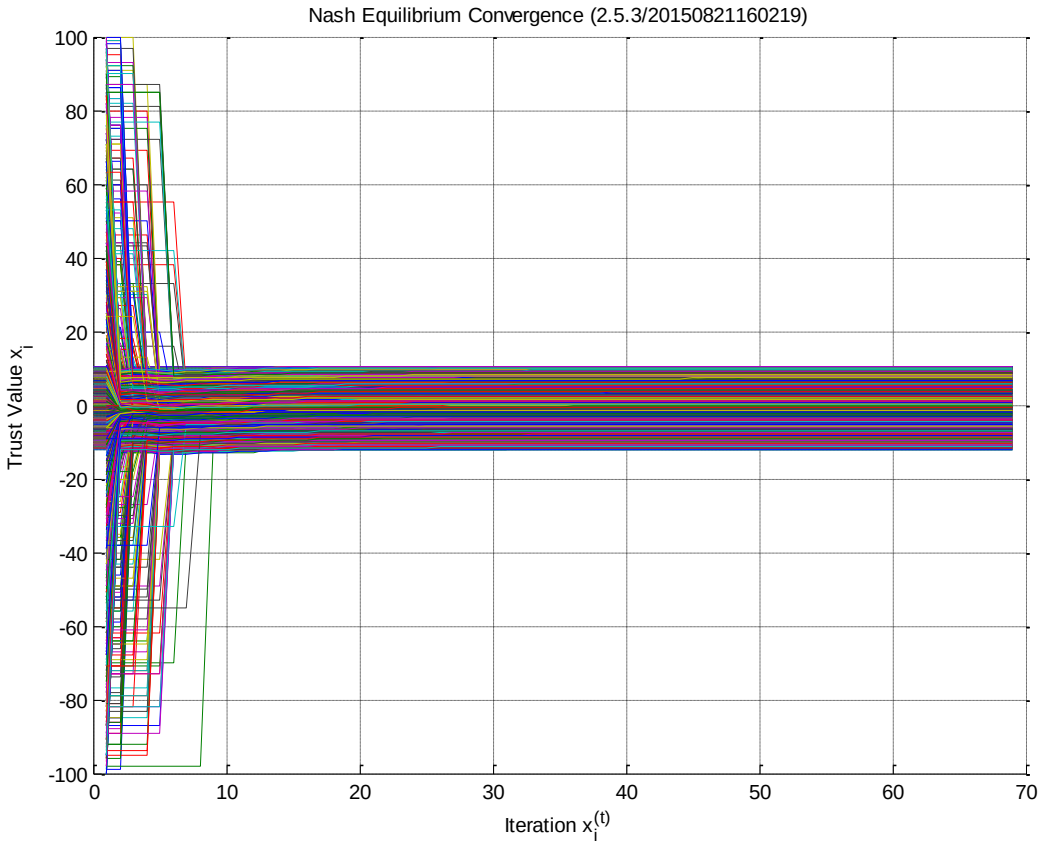


Figure 52 Comparison of Trust Values against iteration count for plot for 50% allocated randomly responsive nodes

Consistent results with Figure 51 can be observed for the Nash Equilibrium convergence plot (Figure 52). Again, there is a stark contrast between Figure 52 and Figure 41. It is possible to visually distinguish the convergence of individual node's Trust Values in Figure 52 because their behaviour differs so much as they randomly respond, where as in Figure 41, the nodes describe similar, "overlapping" convergence paths.

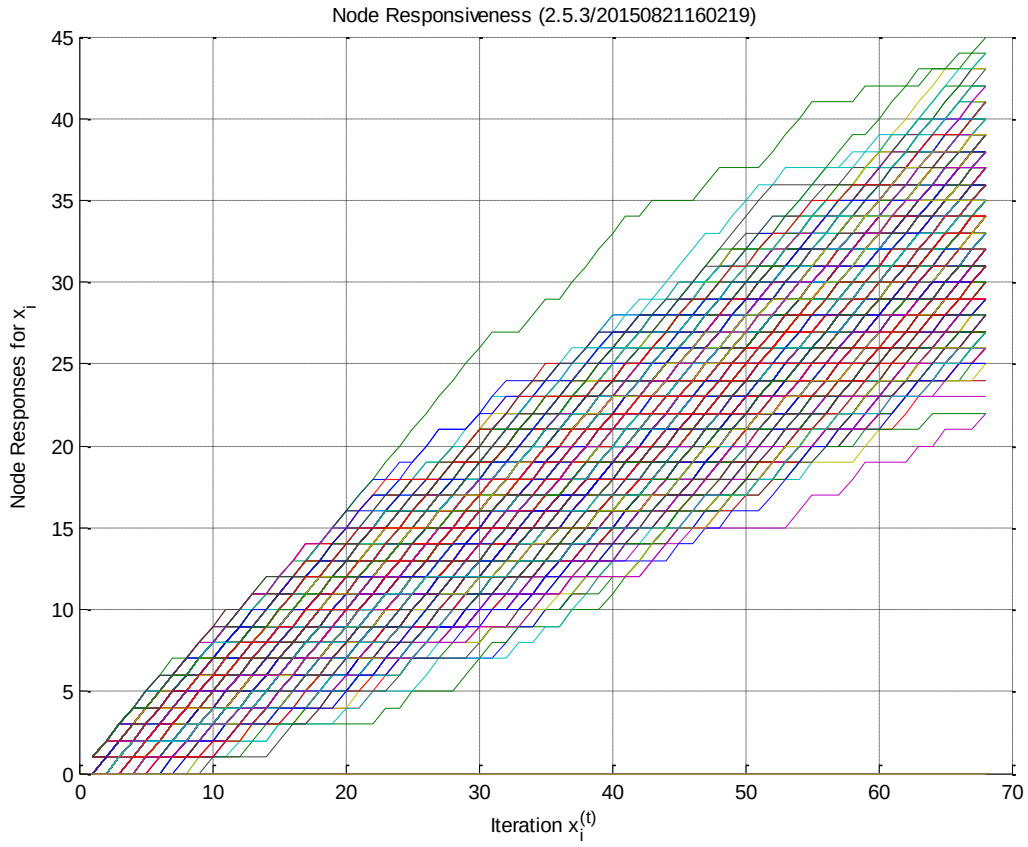


Figure 53 Comparison of random node responses against iteration plot

Each plateau represents a node responding between iterations, where each incline indicates an iteration where the node did not respond. The allocated responsive nodes in the System continue to respond randomly until convergence is reached. Non-responsiveness increments account for each node that fails to respond for an iteration.

Trivial results were observed when the convergence condition is configured to “any” and is therefore excluded from further experimentation where perfect responses are not received from all nodes in the System. Consistent with the scale experiments, it does not matter which Trust Value number type or value range is selected for experimentation and therefore we can assume no degradation to performance.

Nodes are randomly selected to be non-responsive as is most representative of an Emerging System where nodes trivially fail to respond under changing circumstances. This approach will be adopted for all following experiments. The number of nodes that fail is completely dependent on where the Emerging System is applied. More static Systems will have much lower non-responsive rates than dynamic. We take, 90 percent guaranteed response from nodes in a System as reasonable for the establishment of consensus trust and consider a persistently non-responsive node as an extreme

case of a randomly (non-)responsive node, and use this as a basis for further experimentation. This is reflective of a more static implementation but should be adequate for the reasonable bounds of experimental investigation.

4.6.2.2.1.5 Conclusion

The experimental results attest the conclusions that, for Experiment Batch 1.3:

> *in the fixed non-responsive case where the convergence condition is required to be met by any single node that are not randomly distributed in the System, we must strictly refute* H_0 *and accept* H_1*. However, the result is trivial. As more non-responsive nodes are introduced into the System, the number of iterations to convergence remains the same and is always one;*
>
> *in the fixed non-responsive case where the convergence condition is required to be met by all nodes that are not randomly distributed in the System, we must accept* H_0 *and reject* H_1*. As more non-responsive nodes are introduced into the System, the number of iterations to convergence decreases;*
>
> *in the fixed non-responsive case where the convergence condition is required to be met by all nodes that are randomly distributed in the System, we must accept* H_0 *and reject* H_1*. As more non-responsive nodes are introduced into the System, the number of iterations to convergence decreases, and more quickly than the fixed case;*
>
> *in the randomly non-responsive case where the convergence condition is required to be met by all nodes that are randomly distributed in the System, we must refute* H_0 *and accept* H_1*. As more randomly responsive nodes are introduced into the System, the number of iterations to convergence increases.*

Further, O_3 did not significantly change for all experiments. Time per iteration remains consistent, as supported by the scale experiments.

4.6.2.2.2 Experiment Batch 1.4

4.6.2.2.2.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : O_1 and O_3 will decrease with the different Convergence Conditions (X_9) and Stability Strategies (X_7) for non-responsive nodes (X_8R_3)

H_1 : O_1 and O_3 will increase or remain the same with the different Convergence Conditions (X_9) and Stability Strategies (X_7) for non-responsive nodes (X_8R_3)

for Stability Strategy (X_7), for randomly distributed, non-responsive and randomly non-responsive nodes:

- Use previous iteration Trust Value, $x_i = x_{i-1}$;
- Modify current iteration Trust Value by a configured factor or $\frac{1}{\dim(R_j(a_k)) - 1}$, of x_{-i} Trust Value standard deviation;
- Exclude node after $\max(s)$, where $s \in \mathbb{N}$, failed responses, with a corrective criteria:
    - Accumulative ($s_i, (s+1)_j, \ldots, \max(s)_l$, for any i, j, l, where i, j, l are not necessarily consecutive iterations) with;
    - Correction ($s = 0$, or $s = \max(s) - r$, for some configured $r \leq \max(s)$ or 0 if $r > \max(s)$, once or for all subsequent responsive iterations), or;
    - Consecutive ($s_i, (s+1)_{i+1}, (s+2)_{i+2}, \ldots, \max(s)_{\mathrm{i+max(s)}}$, strictly consecutive iterations) with;
    - Correction ($s = 0$).

with Convergence Conditions (X_9):

- $\varepsilon = \{0.0001, 0.0010, 0.0100, 0.1000, d\}$ for some dynamic variable $d \in \mathbb{Q}$, and;
- $\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon$ for any Δx;

and for experimental variable (X_8R_3):

- X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] = $\{[0.0, 99.9, 0.1], [0.0, 99.0, 1.0], [0.0, 90.0, 10.0]\}$.

It is expected that an overarching stability strategy can be identified that not only improves the performance of the simulation by reducing the number of iterations but also establishes a better equilibrium reflection of the System's consensus trust that is not skewed by non-responsive nodes.

4.6.2.2.2.2 Simulation Configuration

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	[-100, 100] ∈ $\mathbb{Z}$
X_4R_1	=	multFibonacci
X_5	=	1,000
X_6R_2	=	Horizontally and vertically symmetric, and uniformly pseudorandom.

Table 19 Simulation framework configuration for Experiment Batch 1.4

The algorithm was configured as follows:

X_7	=	Experimental variable
X_8R_3	=	Experimental variable
X_9	=	Experimental variable
X_{10}	=	100 per phase
X_{11}	=	1

Table 20 Simulation JOR algorithm configuration for Experiment Batch 1.4

Initial Reputation Profile (X_4R_1) and environmental factors (X_6R_2) pseudo-randomly generated.

4.6.2.2.2.3 Results

500 independent experiments were conducted. With $X_8R_3 = \{0.1,99.9,0.0\}$ and an arbitrary Stability Strategy, a single node static non-responsive system with two phase (Uninhibited and Readjustment) execution of the simulation produced the following results:

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations Uninhibited phase (to 2 decimal places)	O_3 Computation real execution time Uninhibited phase (seconds to 9 decimal places)	O_1 Number of algorithm iterations Readjustment Phase (to 2 decimal places)	O_3 Computation real execution time Readjustment Phase (seconds to 9 decimal places)
1	2.00	0.001833014	13.00	0.047483253
2	3.09	0.006397625	14.18	0.052543686
3	4.00	0.009436880	12.20	0.044653424
4	5.00	0.013085895	11.20	0.040923055
5	6.00	0.016528465	12.30	0.045030155

Table 21 Results for two phase for Experiment Batch 1.4

A further 500 independent experiments were conducted. With $X_8R_3 = \{0.0,99.9,0.1\}$, a consecutive Stability Strategy (correction to zero and arbitrary repetition), a single node randomly non-responsive system with two phase (Uninhibited and Readjustment) execution of the simulation produced the following results:

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations Uninhibited phase (to 2 decimal places)	O_3 Computation real execution time Uninhibited phase (seconds to 9 decimal places)	O_1 Number of algorithm iterations Readjustment Phase (to 2 decimal places)	O_3 Computation real execution time Readjustment Phase (seconds to 9 decimal places)
1	2.50	0.003552503	11.50	0.042466139
2	5.20	0.014767884	11.00	0.040963555
3	8.67	0.032311299	5.11	0.020119717
4	13.30	0.051773942	4.20	0.015739476
5	13.40	0.053147449	0.00	0.000000000

Table 22 Results for two phase for Experiment Batch 1.4

500 independent experiments were conducted. With $X_8R_3 = \{0.0,99.9,0.1\}$, an accumulative Stability Strategy, correction to -1 (without repetition), a single node randomly non-responsive

system with two phase (Uninhibited and Readjustment) execution of the simulation produced the following results:

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations Uninhibited phase (to 2 decimal places)	O_3 Computation real execution time Uninhibited phase (seconds to 9 decimal places)	O_1 Number of algorithm iterations Readjustment Phase (to 2 decimal places)	O_3 Computation real execution time Readjustment Phase (seconds to 9 decimal places)
1	2.80	0.005685275	9.20	0.035891021
2	8.30	0.030280587	7.60	0.025431984
3	9.40	0.032606896	12.10	0.047846761
4	9.80	0.035258543	9.50	0.037329856
5	10.33	0.039308789	6.89	0.026854578

Table 23 Results for two phase for Experiment Batch 1.4

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)
1	15.00	14.00	12.00	0.049316267	0.046018642	0.041576296
2	17.27	16.20	15.90	0.058941311	0.055731439	0.055712571
3	16.20	13.78	21.50	0.054090304	0.052431016	0.080453657
4	16.20	17.50	19.30	0.054008950	0.067513418	0.072588399
5	18.30	13.40	17.22	0.061558620	0.053147449	0.066163367

Table 24 Combined total results for two phase for Experiment Batch 1.4

300 independent experiments were conducted. With $X_8R_3 = \{0.0,90.0,10.0\}$ (10% randomly non-responsive nodes), an accumulative Stability Strategy with no correction, 1,000 node system execution of the simulation yielded the following results:

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	Number of Phases (to 2 decimal places)
1	22.33	0.006538337	15.33
2	26.02	0.011901505	19.67
3	57.75	0.046996091	51.00

Table 25 Stability Threshold effect on convergence for Experiment Batch 1.4

100 independent experiments were conducted. With $X_8R_3 = \{0.0,90.0,10.0\}$, a consecutive Stability Strategy (with zeroing correction by definition), 1,000 node system, with a Stability Threshold of 3, execution of the simulation produced the following results:

X_7 Readjustment Scheme (Stability Threshold)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	Number of Phases (to 2 decimal places)	X_7 Stability Strategy
3	75.25	0.062006952	65.05	Consecutive (zeroing correction)
3	*57.75*	*0.046996091*	*51.00*	*Accumulative (no correction)*

Table 26 Consecutive and Accumulative Stability Strategy comparison for Experiment Batch 1.4

190 independent experiments were conducted. With $X_8R_3 = \{0.0,90.0,10.0\}$, a consecutive Stability Strategy (with zeroing correction by definition), 1,000 node system, with a Stability Threshold of 3, execution of the simulation produced the following results:

X_9 Convergence Condition (to four decimal places)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	Number of Phases (to 2 decimal places)
0.0001	*65.05*	*0.062333869*	*76.37*
0.0010	62.20	0.054506979	55.20
0.0100	86.00	0.068393735	76.00
0.1000	24.86	0.028690086	35.00
1.0000	3.00	0.014610265	1.00

Table 27 Effects of Convergence Conditions on simulation performance for Experiment Batch 1.4

140 independent experiments were conducted. With $X_8R_3 = \{0.0,90.0,10.0\}$, a consecutive Stability Strategy (with zeroing correction by definition), 1,000 node system, with a Stability Threshold of 3 and Convergence Condition 0.0001, execution of the simulation produced the following results:

X_7 Readjustment Scheme Trust Value Adjustment (to 2 decimal places)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)	Number of Phases (to 2 decimal places)
None	*65.05*	*0.062333869*	*76.37*
0.01	114.00	0.141937412	100.00
0.10	115.00	0.141082680	99.67
0.25	116.33	0.151991560	100.33
0.50	116.33	0.162256365	100.67
1.00	118.00	0.169225006	101.00
2.00	117.00	0.155779168	101.00
$\frac{1}{\dim\left(R_j(a_k)\right)-1} = 0.0010$	114.00	0.159791643	100.00

Table 28 Trust Value adjustment on simulation performance for Experiment Batch 1.4

4.6.2.2.2.4 Discussion

A *Stability Strategy* attempts to reduce the influence non-responsive nodes have on the final Reputation Profile. The Stability Threshold is a component of the Stability Strategy that specifies the number of iterations for which a node can be deemed sufficiently non-responsive before it is removed from the System. The Stability Threshold can be configured in different ways, primarily dictated by the application of the Emerging System, and serve as a corrective influence on convergence.

If the Stability Threshold is configured to be *consecutive*, a non-responsive node will increment a non-responsive count uniquely attributed to it, initially at zero, for each consecutive iteration it fails to respond. If the count breaches the Stability Threshold for any iteration, then the node will be removed from the System. If the node responds again for a single iteration before the threshold is breached, its non-responsive count will be reset to zero again.

If the Stability Threshold is configured to be *accumulative*, a non-responsive node will increment a non-responsive count uniquely attributed to it, initially at zero, for each iteration it fails to respond. The non-responses need not be consecutive, which is the primary distinction between the accumulative and consecutive configurations. If the count breaches the Stability Threshold for any iteration, then the node will be removed from the System, in both configurations. Otherwise, if the node responds again for a single iteration before the threshold is breached, its non-responsive count can either be:

- Reset to zero (as in the consecutive case), or;
- Decreased by some configured amount once, or;
- Decreased by some configured amount for each subsequent, consecutive responses.

The non-responsive count is always greater than or equal to zero – it is never negative. Any incremental decrease will therefore cease when the non-response count is zero. The decrease amount can be fixed or derived and is a whole number. This is a *correction*.

The consecutive configuration can be consider a specialisation of the accumulative configuration, with a fixed (zeroing) correction.

When a node fails to respond, a correction can be applied to its Trust Value. This can be derived from its previous iteration response Trust Value, other responsive nodes in the System or could be a fixed value. The correction Trust Value is then used in the next iteration and reapplied for each subsequent iteration, until the node responds or the Stability Threshold is breach and the node is removed from the System. The correction further attempts to reduce the influence the non-responsive node has on the final Reputation Profile.

Other components of the Stability Strategy include the Convergence Threshold. This threshold configures the modulus change of Trust Values between iterations that needs to be achieved in order for convergence to be considered established. This can be configured as being required by all or any nodes. It can be fixed or dynamic depending on the application of the Emerging System.

By way of control and to test the principles of topologic change for the simulation, we consider the single node, static non-responsive case. During topological change, the simulation undergoes multiple combinations of one, some or all of three phases of convergence:

1. *Uninhibited* – where the Emerging System retains a stable number of nodes;
2. *Readjustment* – where a node is sufficiently unresponsive, that it is no longer considered a part of the Emerging System and is excluded, and;
3. *Expansion* - where an additional node or cluster of nodes join the Emerging System (explored in for Experiment Batch 1.5).

A transition through one or many of these phases, represents the changing topology in an Emerging System and how it becomes partitioned as convergence is attempted. To simulate topological change experimentally, we identify the specific phases for an experiment and what the thresholds and stability strategies are for each. From the scale experiments, we know that the range and numeric type of initial Trust Values does not affect the experimental results.

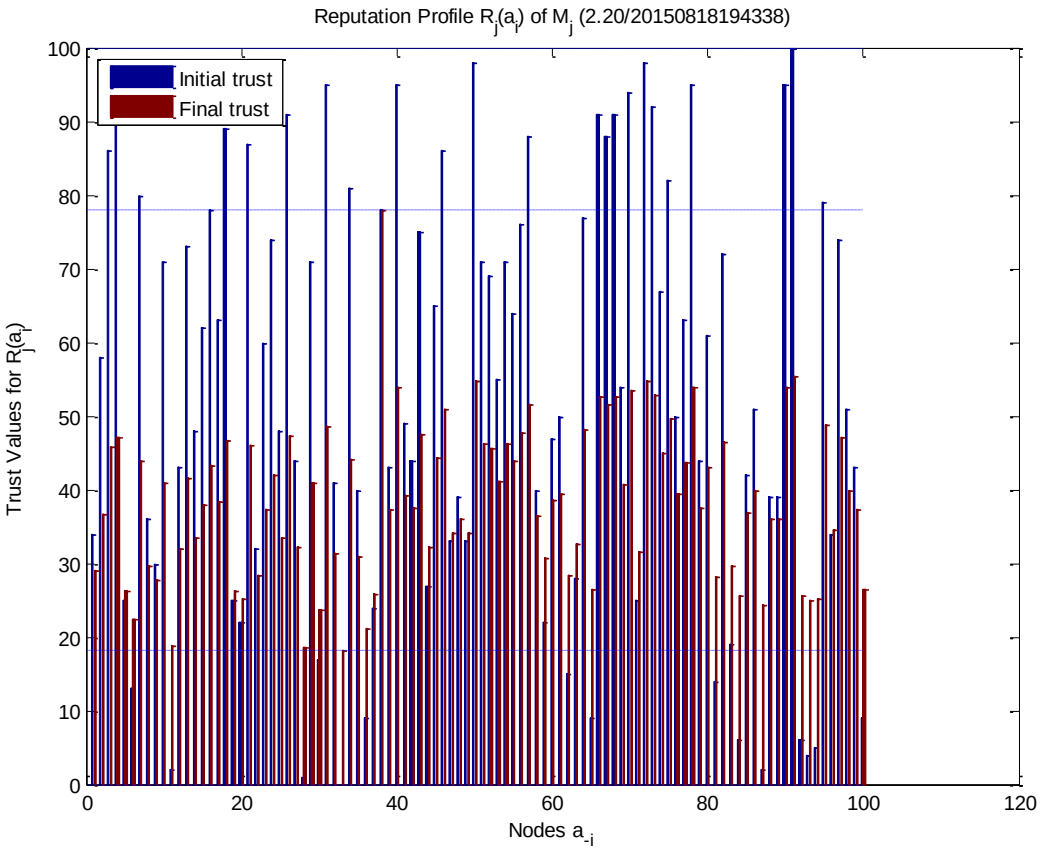


Figure 54 Initial and final Trust Value plot for an Uninhibited phase of experiment 20150818194338

For this experiment, X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $= [0.1, 99.9, 0]$ so a single node in the System of $\dim(R_j(a_k)) = 100$ nodes, is static unresponsive.

Taking the configuration of Experiment Batch 1.3 as the point of departure for these experiments, convergence condition (X_9) is enforced for all nodes in the Emerging System such that:

$$\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq \varepsilon = 0.0001, \text{ for all } \Delta x$$

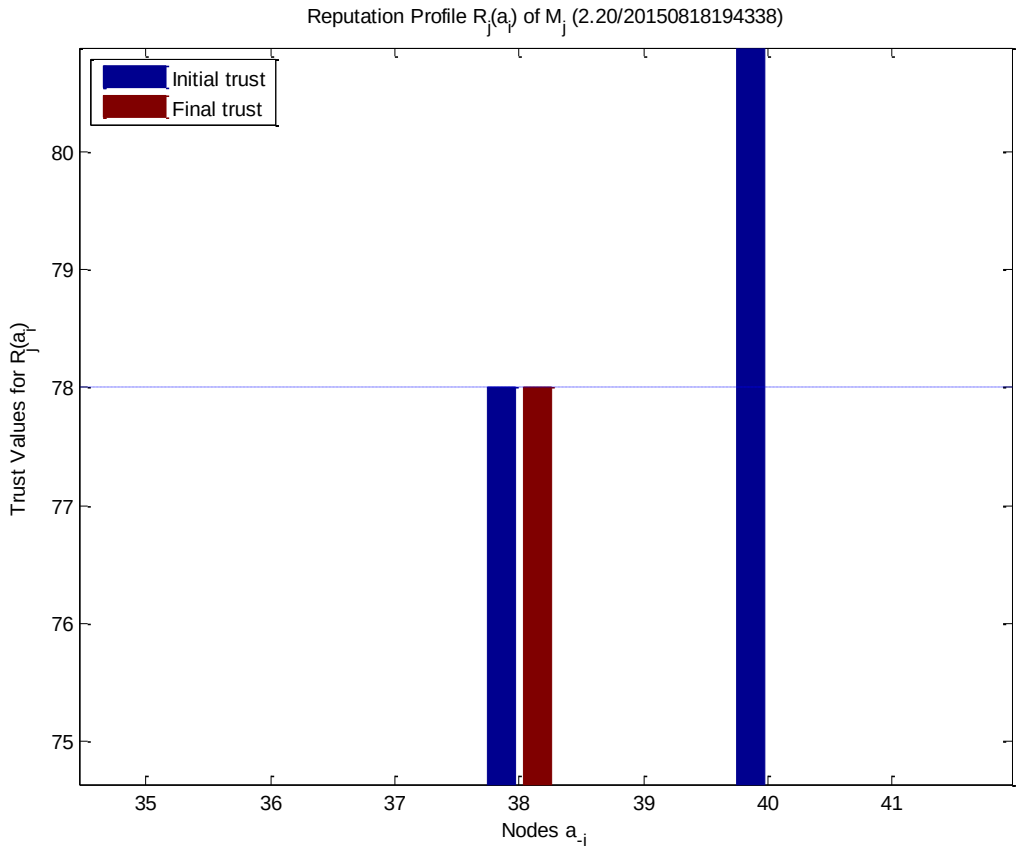


Figure 55 Unresponsive node initial and final Trust Value plot for an Uninhibited phase of experiment 20150818194338

From observation, node a_{38} is the single randomly allocated unresponsive node. This is also verified by the simulation variables. Consequently, the initial and final Trust Values for a_{38} are the same, 78. Since the stability strategy is that when a node is non-responsive, its previous iteration Trust Value is used for the current iteration and in this case, the node never responds (static non-responsive) so we can expect the Trust Value not to change throughout this simulation phase.

The simulation phase is recorded by the convention that its number is included between the experiment reference and stamp, delimited by a colon (":") - <reference>:<phase>/<stamp>.

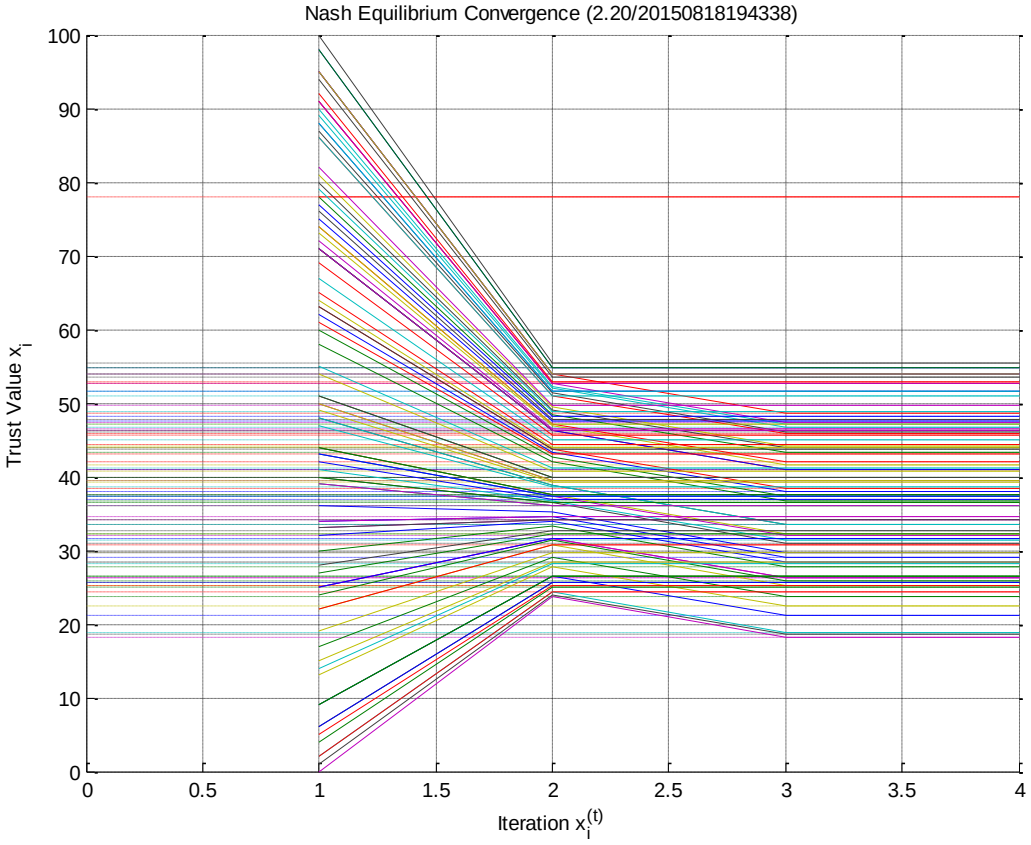


Figure 56 Nash Equilibrium convergence plot for an Uninhibited phase of experiment 20150818194338

The simulation continues in an Uninhibited phase with a_{38} failing to respond (represented by the outlier in Figure 56), until the readjustment scheme (X_7) threshold is breached. This occurs after 3 iterations (O_1) in 0.001494464 seconds (O_3). This breach could also occur based on other factors such as the convergence change of the other nodes in the System.

In Experiment Batch 1.3, the readjustment scheme did not influence the experiments. In this case, the failure of a node to respond twice, causes the termination of the current Uninhibited phase with the following variations in initial and final Trust Values (to 2 decimal places):

Variable	Minimum	Maximum	Norm	Mean	Median	Standard Deviation
X_4R_1 Initial Reputation Profile Trust Values	0.00	100.00	100.00	52.69	50.50	28.95
O_4 Final Reputation Profile Trust Values	18.19	78.00	59.81	38.74	29.42	10.50

Table 29 Initial and final Reputation Profile Trust Values for experiment Uninhibited phase of experiment 20150818194338

There is clearly some convergence to equilibrium although the convergence condition X_9 is not reached and it is skewed by the static non-responsive node a_{38}. The median and mean are indicative of a convergence result as they get smaller, with a smaller standard deviation, indicating that the mean difference between each node's Trust Value and the mean, is decreasing. The maximum for the final Reputation Profile (O_4) is, by configuration, a_{38}.

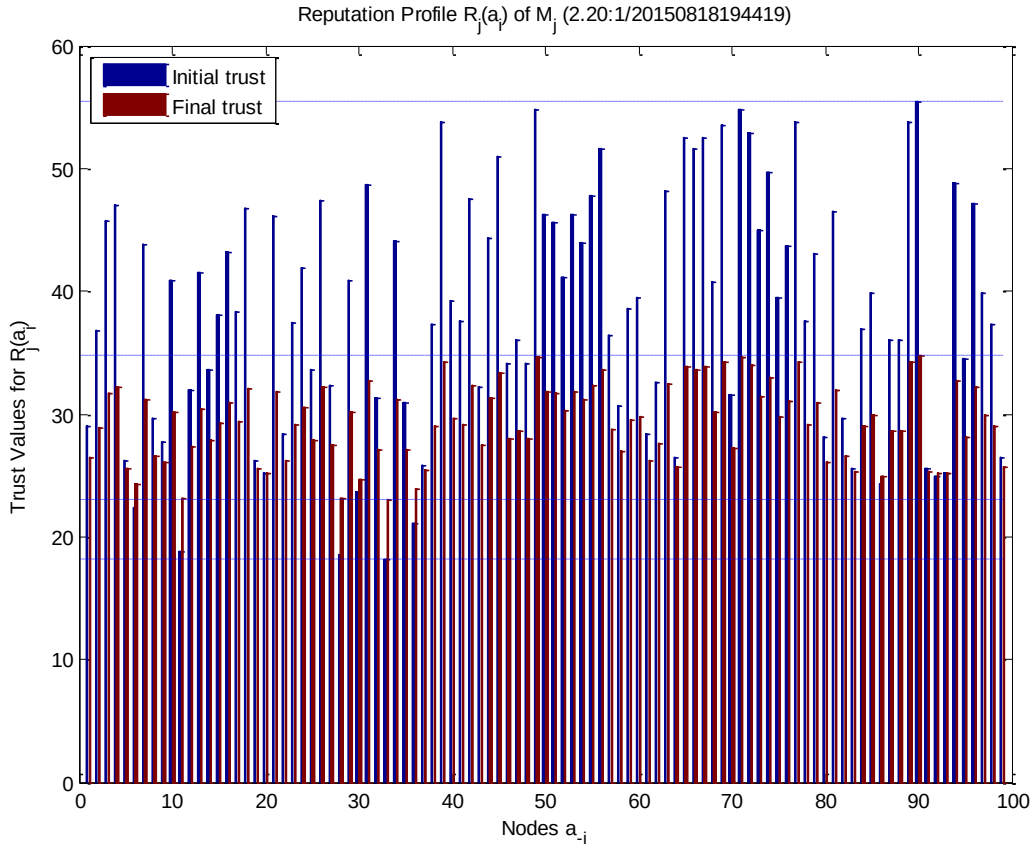


Figure 57 Initial and final Trust Value plot for a Readjustment phase of experiment 20150818194338

The simulation enters a Readjustment phase for a single iteration (O_1) with the non-responsive node, a_{38} removed from the System, so now $\dim(R_j(a_k)) = 99$. The iteration completes in 0.000260525 seconds (O_3) and the Trust Values continue to converge.

In the case that several nodes or cluster of nodes leave the System at the same time, the simulation enters a Readjustment phase for each node that breaches the readjustment schema threshold. The readjustment scheme could be set to any positive value for removal of a node (in the case here, it is two) including one with the result that any node that fails to respond a single time, will be removed. Resumption from a readjustment takes place at the next iteration.

In the case considered here, the distinction between an accumulative or consecutive readjustment thresholds is arbitrary since the non-responsive node is static, by definition it will never respond.

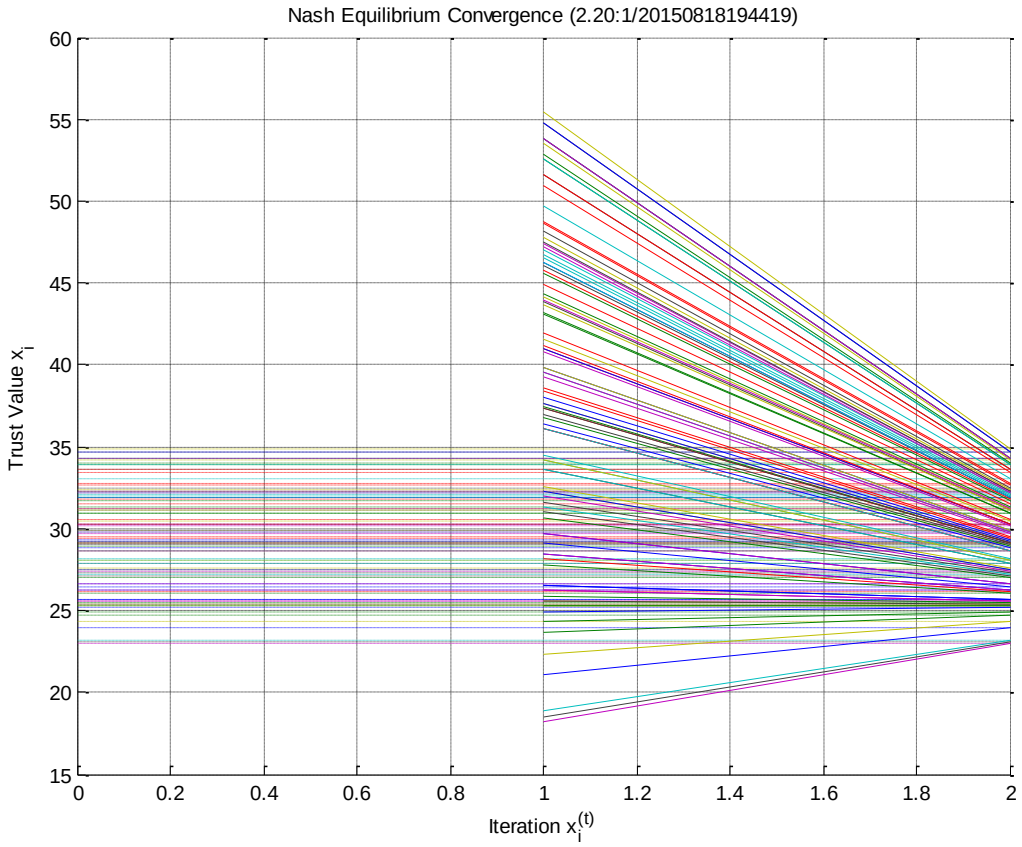


Figure 58 Nash Equilibrium convergence plot for a Readjustment phase of experiment 20150818194338

After the Readjustment phase, the Trust Values continue to converge with the non-responsive node removed from the System:

Variable	Minimum	Maximum	Norm	Mean	Median	Standard Deviation
X_4R_1 Initial Reputation Profile Trust Values	18.19	55.44	37.25	38.34	38.39	9.77
O_4 Final Reputation Profile Trust Values	23.01	34.84	11.83	29.41	29.42	3.10

Table 30 Initial and final Reputation Profile Trust Values for experiment Readjustment phase of experiment 20150818194338

The previous maximum final Trust Value has significantly reduced from the previous Uninhibited phase since it was attributed to the static non-responsive node, while the minimum initial Trust Value remains unchanged. The norm has reduced by almost 48% which is indicative of how much of

an outlier the unresponsive node was. The standard deviation has reduced significantly even after a single iteration which indicates that convergence is occurring more rapidly than in the Uninhibited phase which is again attributable to the outlying, unresponsive node skew.

Subjectively, the removal of nodes elicits a truer representation of consensus trust because it assures that the contributions to the final Trust Values are legitimately derived. Nodes that fail to respond nonetheless continue to exert influence on the final Trust Values, if they are not removed. Readjustment strategies that artificially modify a nodes response are small compensation for an actual response. Restarting the trust analysis each time a node does not respond however, ensuring that only nodes that are fully responsive are considered is not a strategy that supports the topological volatility of an Emerging System well. This approach would result in protracted execution iterations and time as the consensus evaluation is repeated, potentially without resolution. The best approach then, appears to be a contextually well-suited stability and readjustment strategy.

All of the stability and readjustment strategies tested here should be considered in combination to provide the best over-arching strategy for a particular manifestation of an Emerging System. The strategies should be "tuned" to ensure that they are well suited to their specific application.

At the point where nodes no longer respond reasonably within an experiment as dictated by the readjustment condition, we consider that the node is no longer able to meet the sufficient conditions for Nash Equilibrium, specifically, the agents' execution is flawed (3.7.4 Sufficient Conditions).

For the initial experiments in this batch, the single node static non-responsive case ($X_8R_3 = \{0.1,99.9,0.0\}$) always exhibits an initial Uninhibited phase iteration count equal to the Stability Threshold. The higher the Stability Threshold, the longer the execution of the Uninhibited phase takes. The mean execution time for each iteration remains consistent throughout all experiments (mean 0.002143616 seconds for these experiments) borne out by a small standard deviation for iteration execution time (0.000656563). There is a linear correlation between the number of iterations and the execution for the Uninhibited phase.

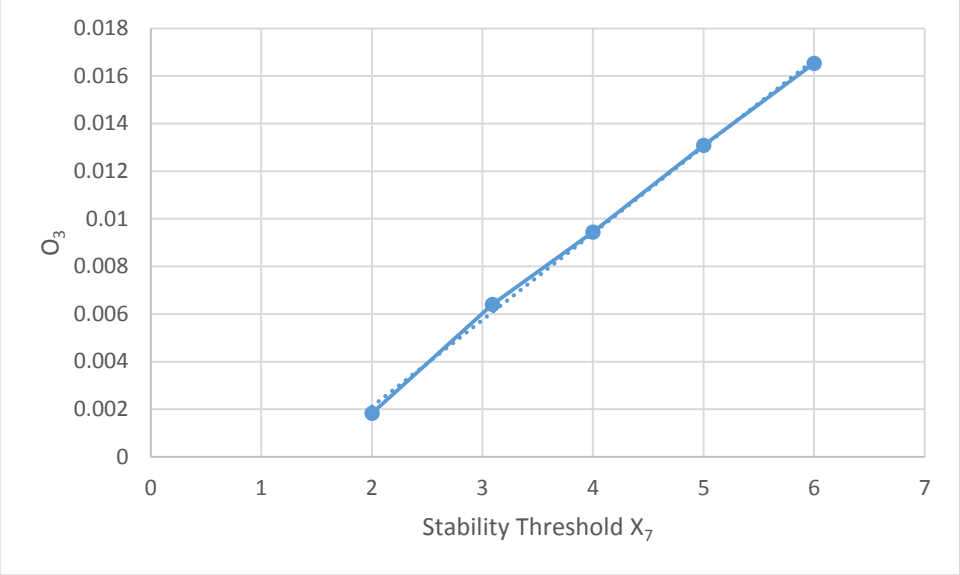


Figure 59 Iteration count and computation execution time for first phase, Uninhibited static non-responsive single node Emerging System plot

These are unremarkable results since this behaviour is common to all single node static non-responsive cases. As the Stability Threshold increases, it is more likely that convergence is achieved before the threshold is breached. If this occurs, then we have a similar case to the single phase experiments. In the high Stability Threshold case, the results converge to the single phase case. This reinforces the consistency of the results of the experiments.

The second, Readjustment phase, appears to converge consistently regardless of the initial Uninhibited phase. Iteration count and execution time are remarkably consistent for this phase while continuing to correlate directly to each other, as we have seen in every other experiment. The second phase should not be affected by the choice of Stability Threshold since, by configuration, no nodes are able to cause a breach since they all respond for every iteration. The only node that does not respond, has been removed. There appears to be no significant influence on the second phase from the first. Considered as a separate experiment, the second phase reflects the results seen in previous experiments where all nodes are configured to respond.

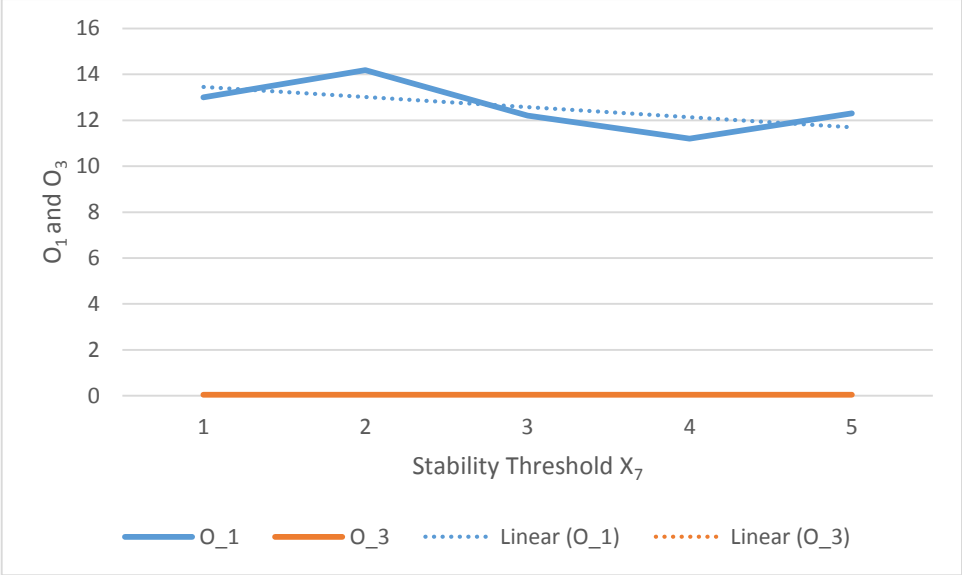


Figure 60 Iteration count and computation execution time for second phase, Readjustment static non-responsive single node Emerging System plot

This type of two phase behaviour might be seen in a highly fault tolerant, stable Emerging System where non-responsive nodes are anomalistic. An unresponsive event might trigger an alert within the System to indicate that some type of administrative intervention is necessary (Young 2008).

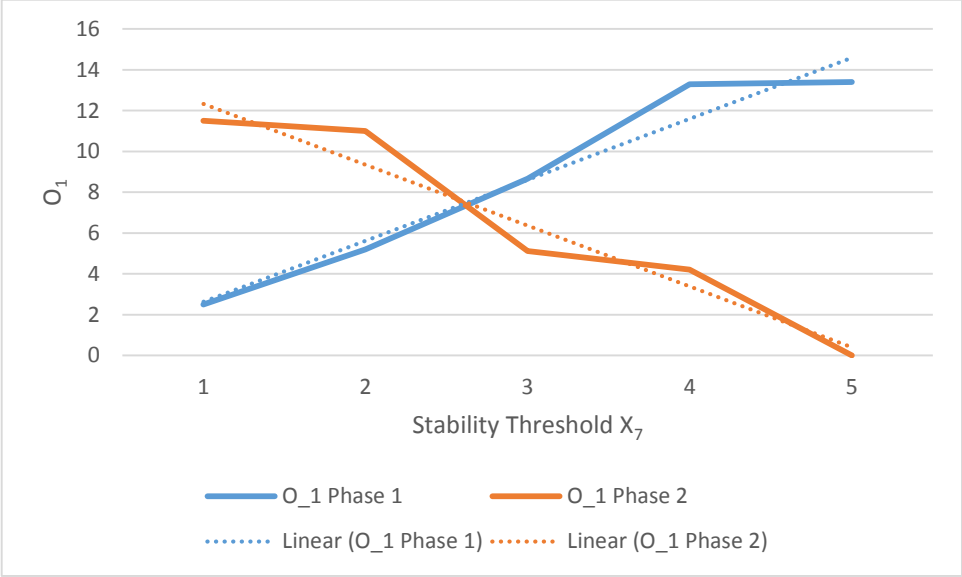


Figure 61 First and second phase iteration count for increasing Stability Thresholds for single node randomly non-responsive zeroing Readjustment Scheme Emerging System plot

In the case with a single randomly responsive node ($X_8R_3 = \{0.0,99.9,0.1\}$), the number of initial Uninhibited phase iterations increases with the Stability Threshold, but there is no longer a direct correlation as we have seen before. Computation time per iteration remains constant.

Unlike the previous static case, the number of iterations for the second Readjustment phase decreases until the threshold is never breached before convergence is achieved. This is the case consistently for a Stability Threshold of five. Convergence is reached wholly within the first phase.

The Readjustment Scheme enforced resets the non-responsive count for a node to zero if the node becomes responsive again.

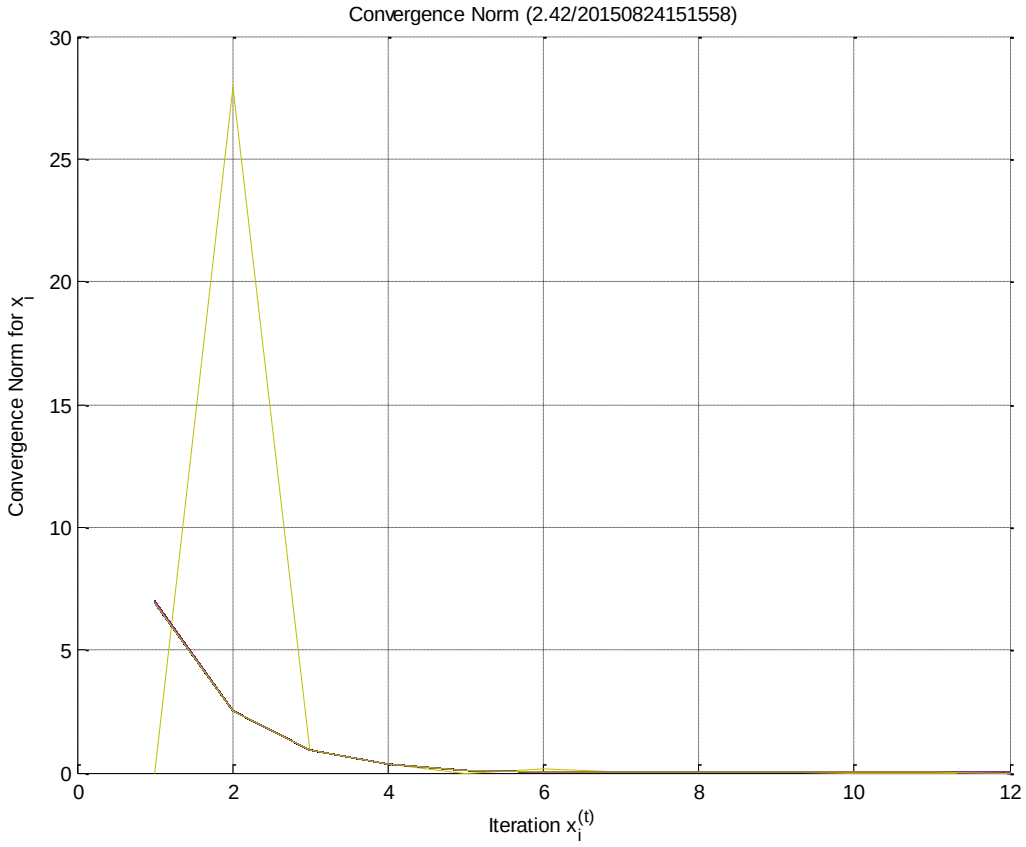


Figure 62 Typical convergence norm plot for a zeroing Readjustment Scheme

A typical convergence norm plot for a zeroing Readjustment Scheme shows the non-responsive node's convergence relative to the other responsive nodes in the System. Once the node is removed (or the System converges), the convergence norm plot exhibits a familiar smooth convergence.

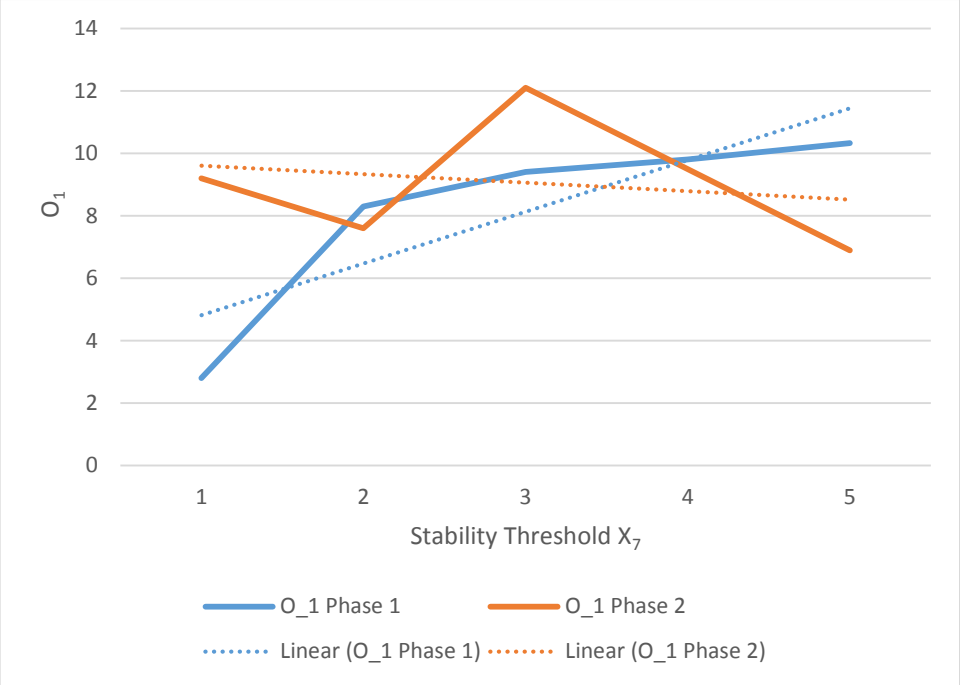


Figure 63 First and second phase iteration count for increasing Stability Thresholds for single node randomly non-responsive -1 Readjustment Scheme Emerging System plot

For a similar System under the same conditions, but with the Readjustment Scheme reducing the non-responsive count for a non-responsive node by one each iteration, if it later becomes responsive, (rather than reducing it to zero) we no longer observe convergence being achieved within the first phase. The number of iterations in the initial Uninhibited phase continues to rise as the Stability Threshold does but the correlation is less consistent. There appears to be little observable pattern to the number of iterations in the second, Readjustment phase of the experiment.

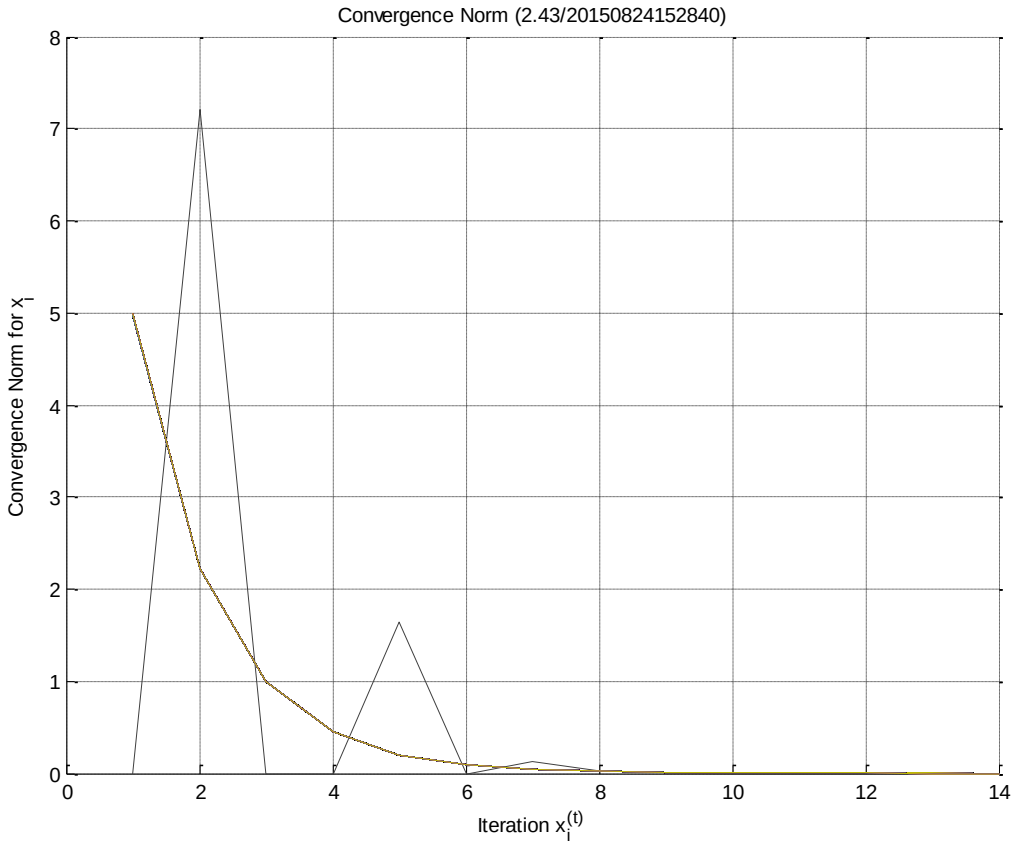


Figure 64 Typical convergence norm plot for a -1 Readjustment Scheme

Unlike the zeroing Readjustment Scheme, a typical convergence norm plot for a -1 Readjustment Scheme clearly indicates the gradual decay in non-responsiveness before the non-responsive node is removed or convergence is achieved. Three peaks can be observed in the plot, iteratively progressively decreasing in magnitude as convergence is approached.

We consider the overall effect over the two phases of the different Stability Thresholds and Readjustment Schemes in these simple cases to determine suitable experiments for the 90% randomly non-responsive Emerging Systems.

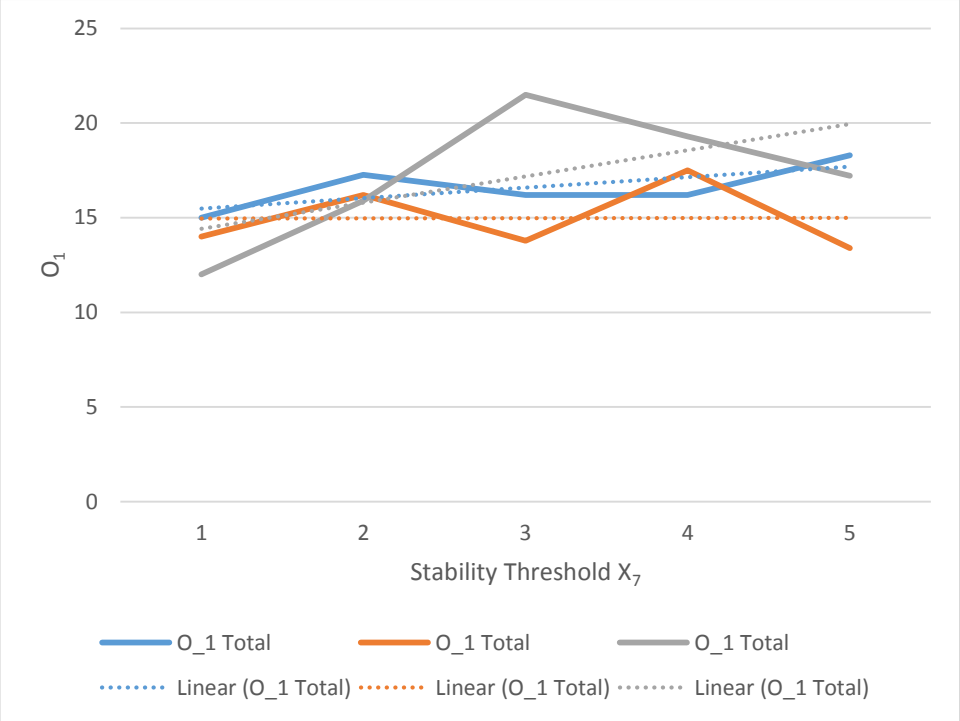


Figure 65 Total two phase iteration count plot

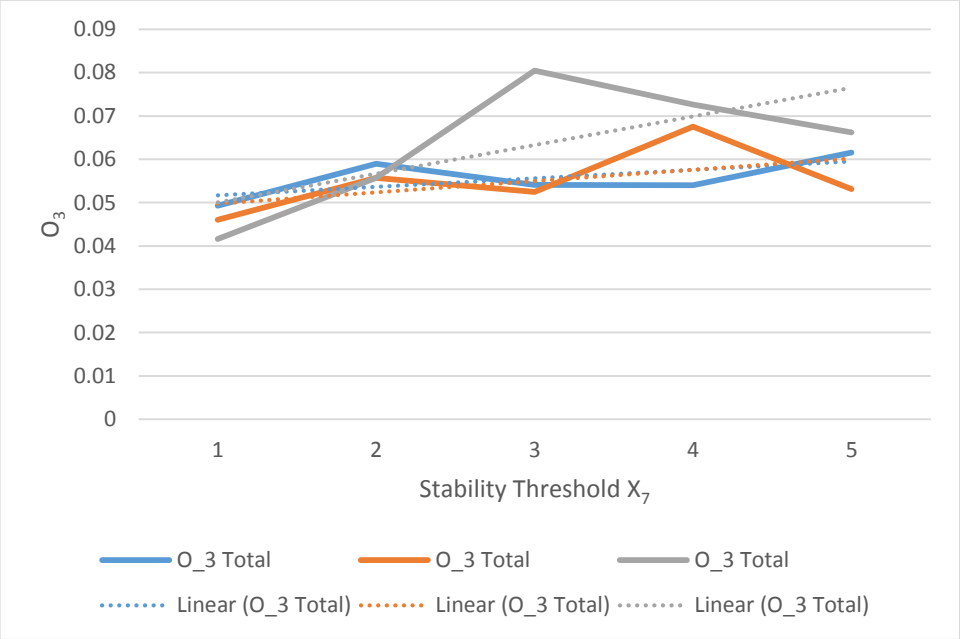


Figure 66 Total two phase execution time plot

The results attest that there is little difference in iteration count between the Readjustment Schemes for low Stability Thresholds. For Stability Thresholds at one and two, a -1 Readjustment Scheme correction is marginally superior. However, above the Stability Threshold of two, the zeroing Readjustment Scheme appears to be most effective in reducing iteration count to convergence. The zeroing Readjustment Scheme correction is most effective when the Stability

Threshold is three. It is also possible with the zeroing Readjustment Scheme correction to achieve convergence in the first phase. Execution time remains directly correlated to iteration count in all cases, consistent with all other findings.

Following on from these results, a Stability Threshold of three with a zeroing Readjustment Scheme correction is selected for 90% non-responsive node experiments. It is not clear how these parameters influence the Emerging System at higher volumes and topological volatility but from these baseline results, they appeared suitable for further experimentation. That is:

- Use previous iteration Trust Value, $x_i = x_{i-1}$;
- Exclude node after $\max(s)$, where $s \in \mathbb{N}$, failed responses, with a corrective criteria:
    - Accumulative ($s_i, (s+1)_j, \dots, \max(s)_l$, for any i, j, l, where i, j, l are not necessarily consecutive iterations) with;
    - Correction ($s = 0$, for all subsequent responsive iterations);

for experimental variable (X_8R_3):

- X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $= \{0.0, 90.0, 10.0\}$.

At higher volumes of nodes with a higher non-responsive percentage, it is reasonable to observe that a less generous Readjustment Scheme will result in faster convergence. The sooner a node is identified as unresponsive and removed from the System, the more quickly the System becomes stable.

Supporting this, with $X_8R_3 = \{0.0,90.0,10.0\}$ and a Stability Threshold of 3, the experimental results indicate between consecutive (zeroing correction) and accumulative (no correction) Stability Strategy (X_7), a ~23% reduction in:

- algorithm iterations from 75.25 to 57.75 (O_1);
- ~0.062 to ~0.047 seconds execution time (O_3), and;
- number of phases, from 65.05 to 51.00;

the convergence experiences to stable conclusion.

Moreover, varying the Stability Threshold with a consistent 10% randomly non-responsive node System, produces comparable results:

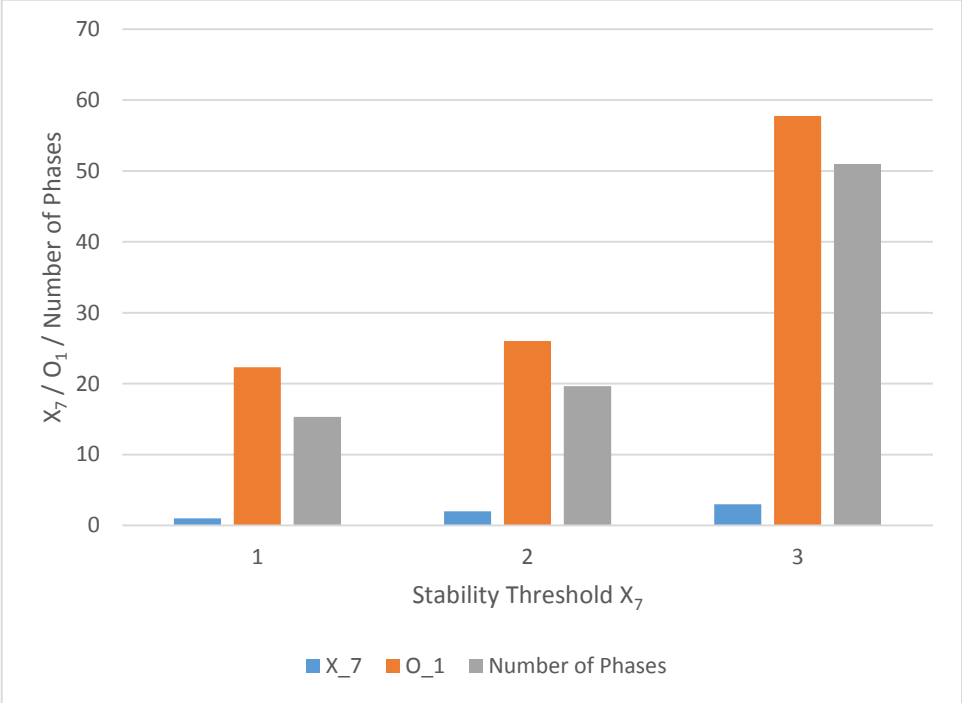


Figure 67 Relative increase in iteration count, execution time and number of phases against Stability Threshold plot

As the Stability Threshold increases, the number of iterations, execution time and number of phases increases relatively. Single iteration time has remained consistent for all experiments, as before.

Removing a node from the System too quickly may prevent it from providing valid future contributions to the establishment of final consensus trust. A node could be non-responsive initially but then be fully responsive later. Consensus trust is more reliably determined from higher volume populations of contributing nodes because the influence of malicious or atypical, outlier responses are reduced.

With $X_8R_3 = \{0.0, 90.0, 10.0\}$, 90% random non-responsive nodes in a 1,000 node System, the iteration count is greatly protracted from the responsive cases. Though the execution time per iteration remains relatively constant between experiments, the simulation actually reports a minimum of a single iteration per phase, when a phase may be less than a complete iteration. This occurs when a several nodes are identified as breaching the Stability Threshold within the same iteration. It is therefore, appropriate to consider the total execution time for the experiment in preference to the iteration count, and further consider the number of phases and their type.

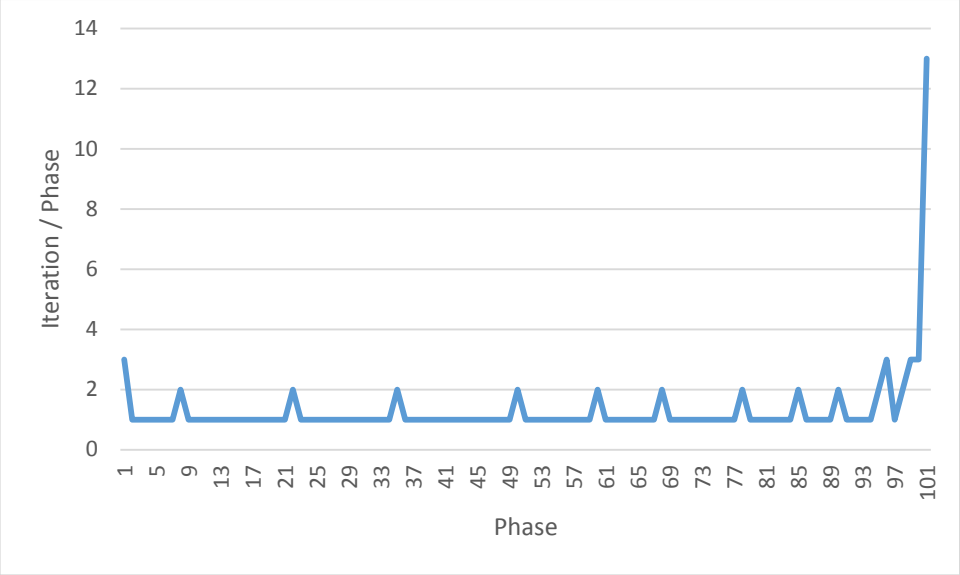


Figure 68 Iteration count per Phase plot

A typical plot for the iteration count per phase to convergence, illustrates the initial Uninhibited phase consisting of a higher than mean iteration count approximately proportional to the Stability Threshold as the initial non-responsive nodes are identified but before any accumulative breach, then the vast majority of phases are Readjustment as non-responsive nodes are eliminated, and then a final Readjustment phase consisting of substantially more iterations after there are no more non-responsive nodes in the System and convergence is attained. With the completion of each readjustment phase, the number of nodes in the System is reduced by one.

Alternative Stability Strategies were tested; decreasing the Convergence Threshold to allow convergence to occur with a statically or dynamically lower consensus agreement, or rather than removing nodes that are non-responsive from the System, augmenting their responses by an amount derived from the current consensus Trust Values:

- Increase ε either statically or dynamically and;
- Modify current iteration Trust Value by a configured factor or, $\frac{1}{\dim(R_j(a_k)) - 1}$, of x_{-i} Trust Value standard deviation.

For $\varepsilon = \{0.0001, 0.0010, 0.0100, 0.1000, d\}$ for some dynamic variable, $d \in \mathbb{Q}$, there is a linear relation between the number of phases before convergence and convergence threshold. The experimental results describe some anomalistic behaviour for $\varepsilon = 0.0100$. The range of number of phases was unusually large with not apparent determinant (range of 80 (from 36 to 116) phases between all experiments with a standard deviation of 33.03). It is reasonable to suggest that the results are within the bounds of expectation for the random components of the simulation.

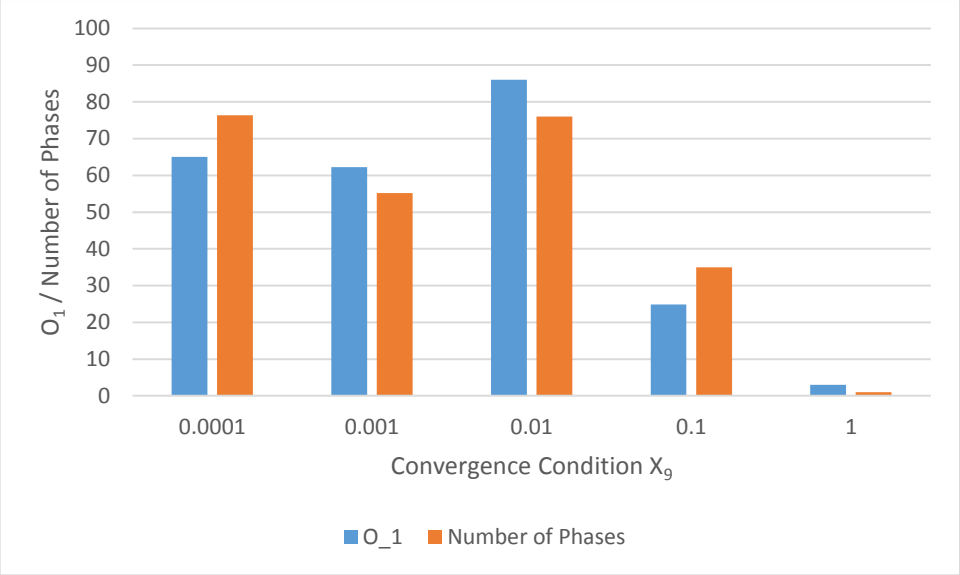


Figure 69 relative increase in iteration count, execution time and number of phases against Stability Threshold plot

With consideration for the anomalistic results for $\varepsilon = 0.0100$, increasing the Convergence Threshold appears to linearly reduce the number of phases to convergence until at $\varepsilon = 1.0000$, there are singles phases of 3 iterations to convergence. Higher ε leads to convergence before nodes have been removed from the System during any Readjustment phase.

Changes to the volume of nodes in the System could be used as a factor in the selection of ε for each iteration. As nodes are added or removed from the System during Readjustment or Expansion phases, ε could be adjusted formulaically. For instance:

$$d = 1/\dim\left(R_j(a_k)\right) = \varepsilon$$

For these experiments, this would be the case the, $\varepsilon = \frac{1}{1000} = 0.0010$ for which, we have results (Table 27).

The experiments in Experiment Batch 1.5 make use of two ε values, first ε_1 to trigger the introduction of additional nodes into the system and then a smaller value, ε_2 as a final Convergence Threshold, $\varepsilon_1 > \varepsilon_2$.

An alternative method for coping with non-responsive nodes is to modify the current iteration Trust Value by a configured factor or function of $\boldsymbol{x}_{-i}$. Adopting a factor of the Trust Value standard deviation at each iteration, modifies the Trust Value of the non-responsive node to the mean of the other Trust Values. The approach reduces the impact that the node has on the final Reputation Profile consensus trust. The non-responsive node's Trust Value undergoes a correction.

For $X_7 = \{None, 0.01, 0.10, 0.25, 0.50, 1.00, 2.00, \frac{1}{\dim\left(R_j(a_k)\right)-1} = \frac{1}{999} = 0.0010\}$, and non-responsive a_k:

$$x_i = x_{i-1} \pm X_7 \sqrt{\frac{1}{\dim\left(R_j(a_k)\right)-1} \sum_{i=1}^{\dim\left(R_j(a_k)\right)-1} (x_{i-1} - \frac{\sum_{i=1}^{\dim\left(R_j(a_k)\right)-1} \boldsymbol{x_{-i}}}{\dim\left(R_j(a_k)\right)-1})^2}$$

The correction in these experiments is a positive or negative factor of the standard deviation of the other nodes in the System at the current iteration, towards the mean. There is no restriction on what the correction function could be but it needs to be tested to determine its effect on convergence.

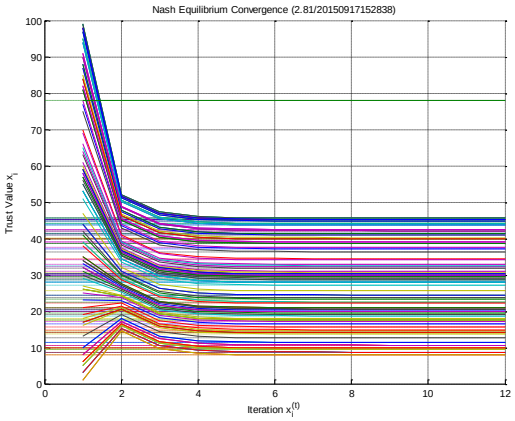


Figure 70 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction zero

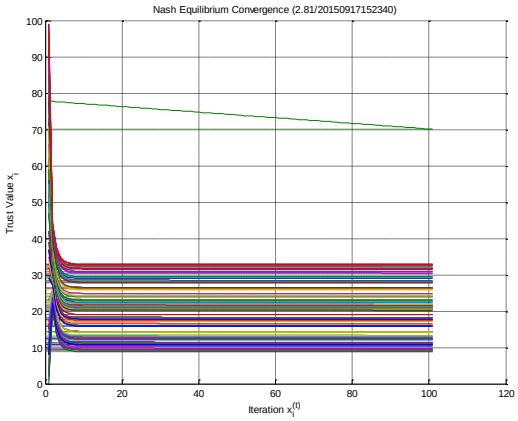


Figure 71 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction 0.01

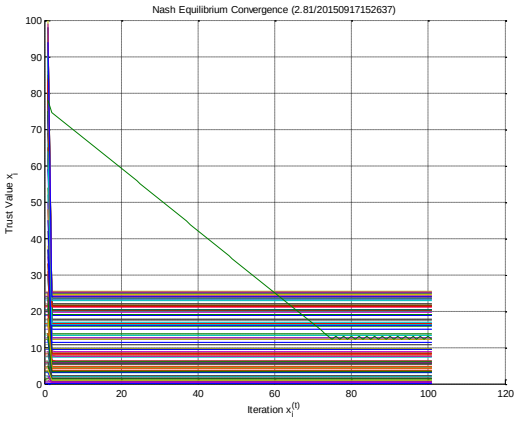


Figure 72 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction 0.10

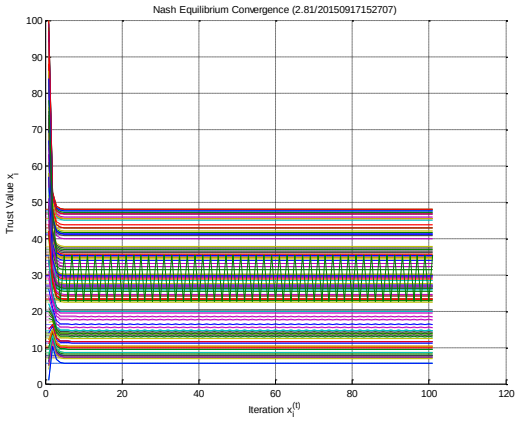


Figure 73 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction 1.00

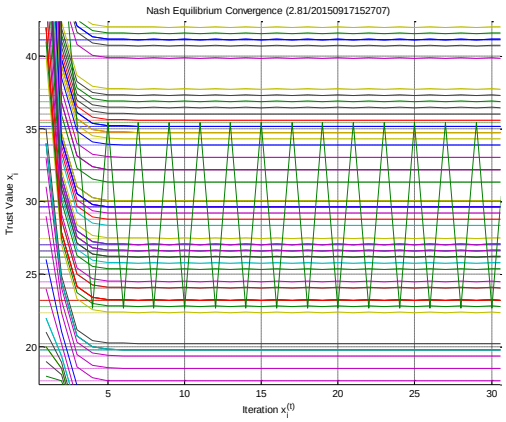


Figure 74 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction 1.00

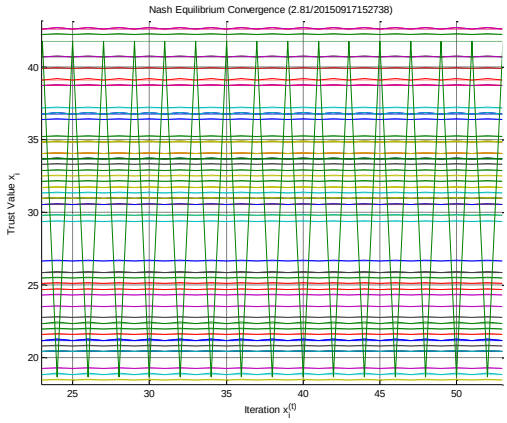


Figure 75 Nash Equilibrium convergence plot for experiment 2.81 with Trust value correction 2.00

For experiment 2.81, Node a_{100} is non-responsive (would remain at a Trust Value of 78 without adjustment (Figure 70)) and undergoes correction at each iteration. The introduction of the correction causes divergence. As the Trust Value correction factor (X_7) increases, the upper iteration bound ($X_{10} = 100$) is breached and the simulation terminates (Figure 71). If the non-responsive node's Trust Value has been corrected to the mean, the Trust Value starts to oscillate by a factor of the standard deviation either side of the mean (Figure 72) until the iteration upper bound is breached. As the factor gets higher, the mean is reached more quickly and the oscillations have a higher amplitude (but similar frequency) (Figure 73 and Figure 74). For $X_7 = 2$ (Figure 75), the Trust Value range is between 18.66 and 41.74 (to 2 decimal places) which is approximately two standard deviations from the mean (23.07 to 2 decimal places).

The divergence is caused by the non-responsive node never meeting the Convergence Condition. There are two potential approaches to addressing this:

1. Ensure that the amplitude of the Trust Value oscillations are statically or dynamically smaller than the Convergence Condition. Then if the Convergence Condition is met by the other nodes in the System within the iteration upper bound, then the condition is also met by the corrected nodes by configuration, or;
2. Remove non-responsive nodes from the Convergence Condition and not entirely from the System. Once the Condition is met by the other nodes in the System, convergence is achieved regardless of the non-responsive nodes. This would have to be considered in the case where nodes change state from being non-responsive to responsive.

These serve both the needs of reducing the influence of a non-responsive node in the System, while not excluding it completely from the trust consensus. In principle, these stability approaches attempt to reduce the impact of the volatility of the System's topology. These means trying to reduce the influence of non-responsive nodes without compromising the final Reputation Profile. To apply a Stability Strategy to a specific Emerging System case, requires that it be tuned to meet the Systems of the System. An optimal Stability Strategy should be derived from the factors that conjure the Emerging System. Potentially, there is a function that best suits an Emerging System, for example for ε, there is some function g dynamic over time t:

$$\varepsilon_t = X_9 = g(X_3, X_5, X_7, X_8R_3\{0.0, 1-q, q\}$$

of initial Trust Value range, initial Reputation Profile dimension, Stability Strategy / Readjustment Scheme / Reinstatement Criteria / Correction and node stability. The Emerging System cannot calculate X_8R_3 and it would have anticipate it by extrapolating the value from historical data or some such. This value could be dynamic and change between the determination of Reputation Profiles.

Final Trust values are greatly influenced by non-responsive nodes. Each iteration reduces each responsive node's Trust Value response until the resulting Reputation Profile is significantly different from the Initial Reputation Profile, as convergence is approached. Consideration should be given to the requirements of the Emerging System to which the choice of NPOST Framework's Stability Strategy is applied.

For this batch of experiments and the following, phases can be considered as discrete experiments therefore inductively, we can conclude that convergence will be reached from the results of the stress experiments. This is how the analysis was carried out with each phases inheriting the previous experiment's output as new input. Predictions can be made as to the behaviour of the simulation on this basis. Phases can be interchangeably introduced to the same original data set to simulate alternative final results against differing topological variances.

4.6.2.2.2.5 Conclusion

The experimental results are adduced to conclude the following:

> *For Experiment Batch 1.4, we must accept H_0 and refute H_1. It is apparent that convergence within the Emerging System to stability can be influenced by changing Stability Strategies, comprising Readjustment Schemes, Response Correction, Reinstatement Criteria, Convergence Conditions and Trust Value Correction, configured for topologically volatility. The experiments do not support a universal optimal strategy that suits all Emerging Systems. The Stability Strategy needs to be adapted to support the configuration and required outcomes, such as the final Reputation Profile, of the System to which it is applied. However, the flexibility of the Stability Strategy as clearly evidenced by the experimental results, supports the hypothesis that a specific, optimal strategy can be configured.*

4.6.2.2.3 Experiment Batch 1.5

4.6.2.2.3.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : O_1 and O_3 will remain consistent with a node volume Expansion phase and convergence (O_5) will be achieved

H_1 : O_1 and O_3 will significantly vary with Expansion phase volume and convergence (O_5) may not be achieved

for Stability Strategy (X_7), for randomly distributed, non-responsive and randomly non-responsive nodes:

- exclude node after $s = 1$, where s and $n \in \mathbb{N}$, failed responses, with an Accumulative reinstatement ($s_i, \dots, s_n$ for any i and $n = 1$, where i and n are iteration counts);

with Convergence Conditions (X_9), where the initial condition triggers the Expansion phase and the second is the true convergence measure, and $\varepsilon_1 > \varepsilon_2$:

- $\varepsilon = \{0.0001, 0.0100\}$ for any Δx;

and (X_5) after Expansion phase:

- $\max(\dim(R_j(a_k)) = \{1{,}000, 1{,}200, 1{,}300, 1{,}400, 1{,}500\}$;

and for experimental variable (X_8R_3):

- X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $=$ $\{100.0,0.0,0.0\}$,and;
- for the Expansion phase, X_8R_3[static non-responsive %, static responsive %, uniformly pseudorandom %] $= \{[100.0, 0.0, 0.0], [0.0, 100.0, 0.0], [0.0, 0.0, 100.0], [0.0, 90.0, 10.0]\}$.

It is expected that when the topology undergoes an Expansion phase, where additional nodes are introduced to the System, there is no significant change to convergence efficiency, considered in terms of iteration count and computational real execution time, and convergence is always attained. The Stability Strategy has been maintained from the previous experimental batches, as has the randomly non-responsive percentage.

4.6.2.2.3.2 Simulation Configuration

There are three possible ways to trigger the introduction of additional nodes into the System. They can either be introduced after a set number of iterations have taken place, when a convergence threshold is reached or Stability Threshold is breached. For the first approach, it cannot be guaranteed where the iteration that triggers the introduction of the additional nodes will occur in the whole convergence progression. It could occur right at the start or at the end. It is not possible to know this in advance because we cannot be sure how many iterations the simulation will take until convergence. It is possible that it could even occur after convergence has been reached and has no effect on the simulation at all. In the second approach, it is reasonable to assume that if the convergence threshold that triggers the introduction of the additional nodes is less than the overall convergence threshold, then it will be breached first and so, the introduction of new nodes is guaranteed before convergence. As with the first approach, it cannot be determined in advance, when this will occur in the overall convergence but it is assured to occur at some iteration before final convergence and randomly. As with the first case, the third case where a Stability Threshold is breached, could potential not occur before overall convergence is reached and is heavily dependent on the stability strategy adopted for the NPOST framework simulation.

The addition of more nodes to the System can be arbitrarily triggered by:

1. Iteration count;
2. Convergence Threshold reach, or;
3. Stability Strategy breach

and is dictated by the configuration of the simulation only.

For the purposes of the experiments that follow, we adopt the configuration that additional nodes are introduced when the Convergence Threshold is reached at:

$$\Delta x = \left\| x_i^{(k)} - x_i^{(k+1)} \right\| \leq 100\varepsilon \text{ for all } \Delta x \text{ and } \varepsilon = 0.0001$$

Initial Reputation Profile (X_4R_1) and environmental factors (X_6R_2) pseudo-randomly generated.

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	[-100, 100] ∈ $\mathbb{Z}$
X_4R_1	=	multFibonacci
X_5	=	1,000 and experimental variable (Expansion phase)
X_6R_2	=	Horizontally and vertically symmetric, and uniformly pseudorandom.

Table 31 Simulation framework configuration for Experiment Batch 1.5

The algorithm was configured as follows:

X_7	=	Accumulative, no correction
X_8R_3	=	Experimental variable (Expansion phase)
X_9	=	0.0001 and 0.0100 (Uninhibited phase)
X_{10}	=	100 per phase
X_{11}	=	1

Table 32 Simulation JOR algorithm configuration for Experiment Batch 1.5

4.6.2.2.3.3 Results

300 independent experiments were conducted. With $X_8R_3 = \{0.0,100.0,0.0\}$ (100% responsive nodes) for Expansion and Uninhibited phases, with 1,000 initial nodes in the System:

X_5 Expansion phase percentage nodes introduced (total nodes)	O_1 Number of algorithm iterations Uninhibited (to 2 decimal places)	O_3 Computation real execution Uninhibited (seconds to 9 decimal places)	O_1 Number of algorithm iterations Expansion (to 2 decimal places)	O_3 Computation real execution Expansion (seconds to 9 decimal places)	O_1 Number of algorithm iterations total (to 2 decimal places)	O_3 Computation real execution total (seconds to 9 decimal places)
10% (1,100)	4.80	0.016622226	10.00	0.044720763	14.80	0.061342989
20% (1,200)	5.40	0.018599096	9.60	0.045839401	15.00	0.064438496
30% (1,300)	5.60	0.019231697	11.40	0.061746092	17.00	0.080977789
40% (1,400)	4.80	0.016496503	10.40	0.077360364	15.20	0.093856867
50% (1,500)	4.40	0.015051065	8.80	0.054347901	13.20	0.069398966

Table 33 Results for Expansion phase percentage volumes for Experiment Batch 1.5

A further 240 independent experiments were conducted. With $X_5 = 10\%$ for the Expansion phase (maximum 1,100 nodes in the System), accumulative Convergence Strategy (no correction) and a Stability Threshold of 1:

X_8R_3 for Expansion phase (to 0 decimal places)			O_1 Phases iterations (to 2 decimal places)			O_3 Phases computation real execution (seconds to 9 decimal places)		
0	1	2	Uninhibited	Expansion	Readjustment	Uninhibited	Expansion	Readjustment
100	0	0	4.60	2.00	1.00	0.016245689	0.008837787	0.000030500
0	*100*	*0*	*4.80*	*10.00*	*0.00*	*0.016622226*	*0.044720763*	*0.000000000*
0	0	100	5.00	1.00	16.80	0.017656825	0.006140390	0.033842415
0	90	10	4.20	1.00	2.50	0.014490542	0.006125370	0.000403537

Table 34 Results for Expansion non-responsive nodes for Experiment Batch 1.5

X_8R_3 for Expansion phase (to 0 decimal places)			Total			O_6 Final Reputation Profile dimension (to 2 decimal places)
0	1	2	Phases	Iterations	Time	
100	0	0	3.00	7.60	0.025113950	1,099.00
0	*100*	*0*	*2.00*	*14.80*	*0.061342989*	*1,100.00*
0	0	100	10.70	22.80	0.057639630	1,091.30
0	90	10	4.50	7.70	0.021019449	1,097.50

Table 35 Total results for Expansion non-responsive nodes for Experiment Batch 1.5

4.6.2.2.3.4 Discussion

During topological change, the simulation undergoes a combination of some or all of three phases of convergence:

1. *Uninhibited* – where the Emerging System retains a stable number of nodes;
2. *Readjustment* – where a node is sufficiently unresponsive, that it is no longer considered a part of the Emerging System and is excluded and;
3. *Expansion* – where additional node or cluster of nodes join the Emerging System, explored here.

To examine a nature of an Expansion phase and its effect on the stability of the System, we consider a specific example.

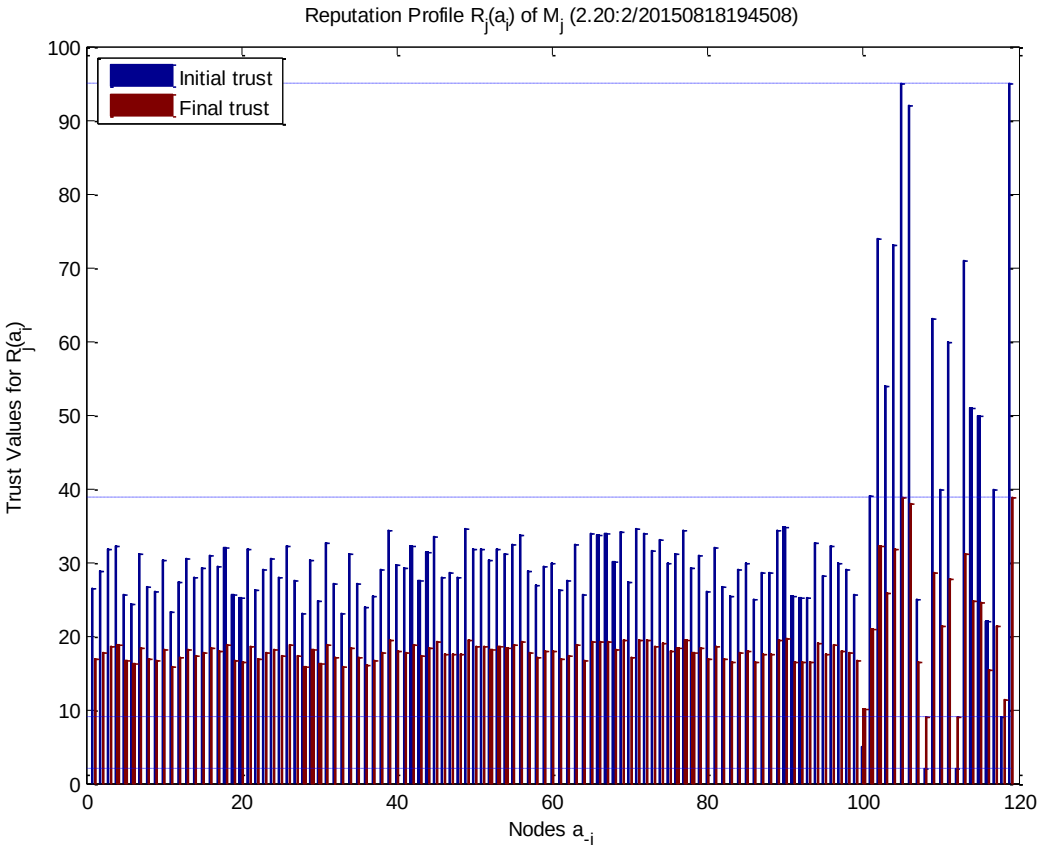


Figure 76 Initial and final Trust Value plot for an Expansion phase of experiment 20150818194338

In the final, Expansion phase of the simulation, a new 10 node cluster is introduced to the System, clearly visible to the right-hand side of Figure 76. Now $\dim(R_j(a_k)) = 109$.

Variable	Minimum	Maximum	Norm	Mean	Median	Standard Deviation
X_4R_1 Initial Reputation Profile Trust Values	2.00	95.00	93.00	43.55	29.80	14.37
O_4 Final Reputation Profile Trust Values	9.09	38.92	29.84	18.89	18.00	4.61

Table 36 Initial and final Reputation Profile Trust Values for experiment Expansion phase of experiment 20150818194338

With the introduction of 10 new nodes to the System, observational variables were significantly different. The norm, minimum and maximum Trust Values reverted to values similar to those at the beginning of the Uninhibited phase, since the new cluster all assumed values from the initial Trust Value range, $X_3 = 0 \leq x_i \leq 100$. The range was reduced by the end of the phase when convergence was attained with the standard deviation indicating the close proximity of the final Reputation Profile Trust Values to the mean.

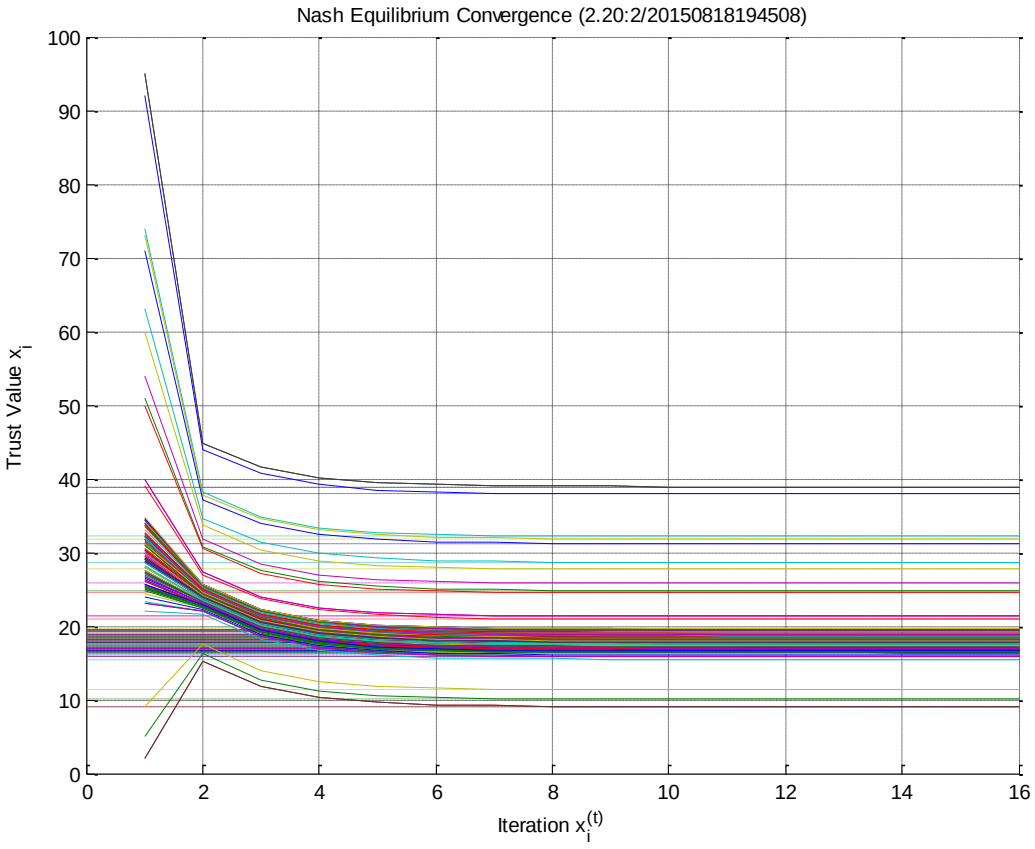


Figure 77 Nash Equilibrium convergence plot for an Expansion phase of experiment 20150818194338

The Expansion phase completed in 15 iterations (O_1) after 0.004633787 seconds (O_3) and reaches the convergence threshold (X_9). It is clearly observable in Figure 76 where convergence takes place for the newly introduced cluster of nodes and the end of the Readjustment phase. The newly introduced nodes are far more broadly distributed as they were introduced within the original range of initial Trust Values, before any consensus convergence has progressed.

Complete convergence occurs over the three phases after 19 iterations and in 0.006388776 seconds.

X_8R_3			O_1 Number of algorithm iterations (to 2 decimal places)	O_3 Computation real execution time (seconds to 9 decimal places)
0	**1**	**2**		
0.0	99.9	0.1	19.00	0.006388776
0.1	99.9	0.0	9.45	0.036313577

Table 37 Initial and final Reputation Profile Trust Values for experiment Expansion phase of experiment 20150818194338

From the experimental results, it appears that as the Expansion percentage of X_5 increases, the initial Uninhibited phase iteration count (O_1) and execution time (O_3) do not vary significantly. The mean iteration count for the Uninhibited initial phase is 5.00 (to 2 decimal places) with a standard deviation of 0.490 (to 3 decimal places). Similar can be observed for the Expansion phase with a mean iteration count of 10.04 (to 2 decimal places) with a standard deviation of 0.963 (to 3 decimal places). Naturally, the total iteration count and execution time for the total phases reflects a similar observation; mean total iteration count of 15.04 (to 2 decimal places) with a standard deviation of 1.352 (to 3 decimal places), and mean total execution time of 0.074003021 (to 9 decimal places) with a standard deviation of 0.013376999 (to 9 decimal places), respectively.

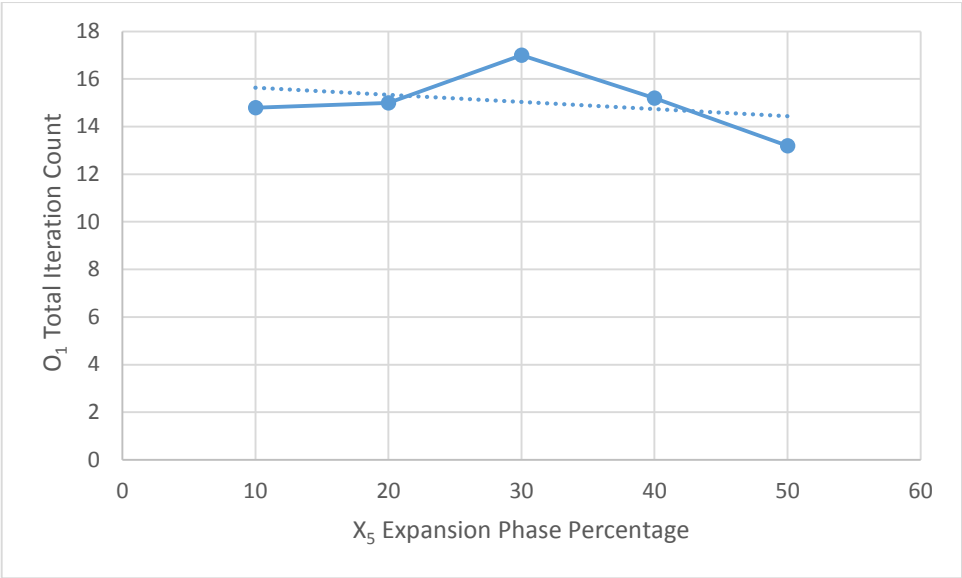


Figure 78 Total iteration count against Expansion phase percentage

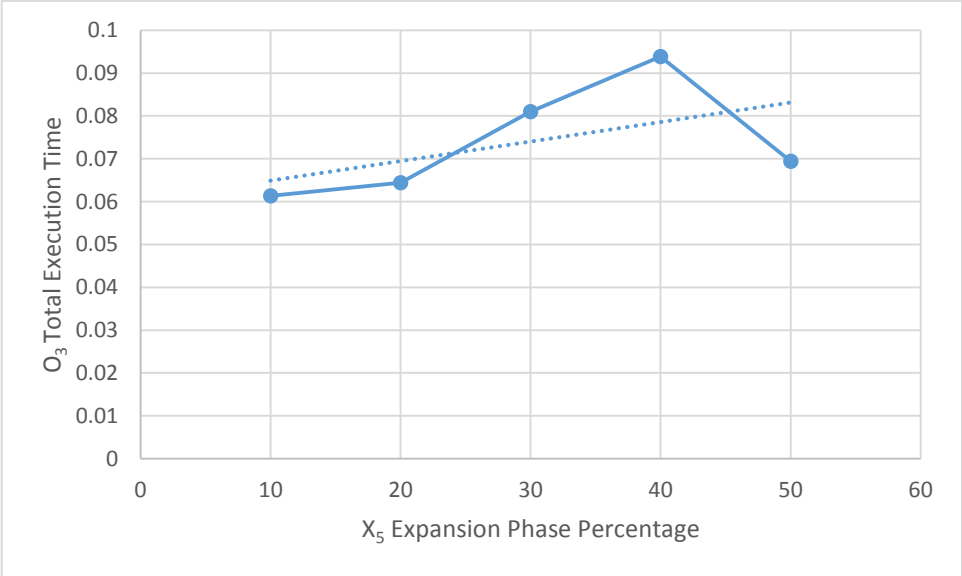


Figure 79 Total execution time against Expansion phase percentage

The balance of evidence suggests that as the volatility of the System increases, that more nodes have the potential to be non-responsive, the number of phases, iterations and total execution time increases, consistent with previous experimental results. It is not apparent that the increase is due to the introduction of new nodes.

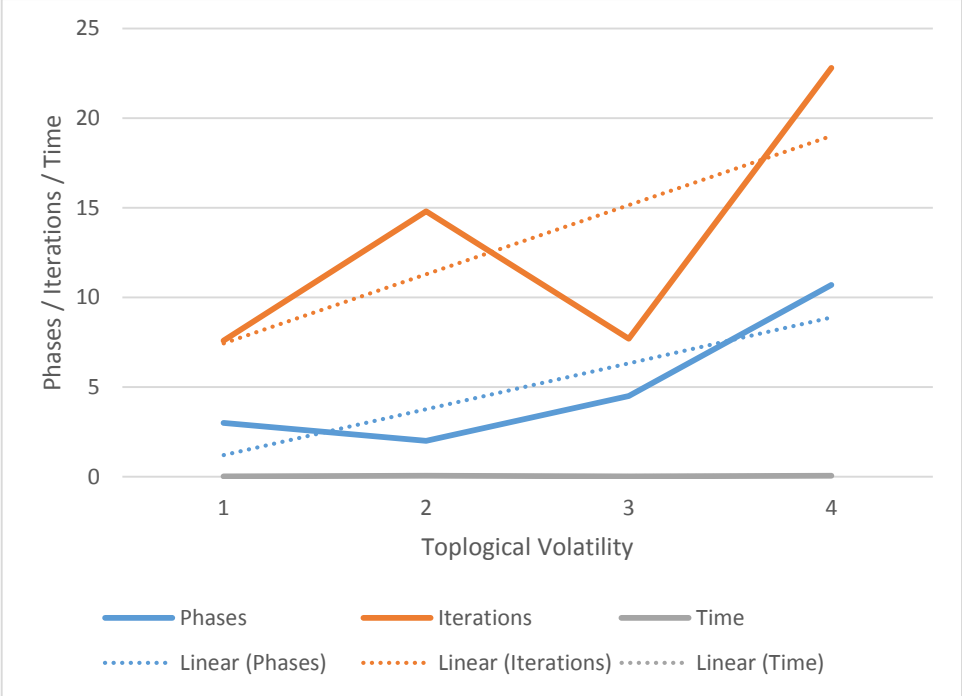


Figure 80 Volatility Expansion phase nodes against total phases, iteration count and execution time

The mean final $\dim\left(R_j(a_k)\right) = 1{,}096.95$ (to 2 decimal places), (O_6) with a standard deviation of 3.904 (to 3 decimal places), indicating very similar outcomes. In the fully responsive case, $\dim\left(R_j(a_k)\right)$ remained at 1,100 as would be expected since all nodes responded for all iterations, the System did not undergo a Readjustment phase to remove any of them, that is:

$$X_5 + Readjustment\ volume = O_6$$

4.6.2.2.3.5 Conclusion

The experimental evidence supports the conclusion:

> *For Experiment Batch 1.5, we must accept H_0 and refute H_1. Execution time and iteration count are not significantly altered by Expansion phases. The volatility of the Expansion phase nodes effects all aspects of convergence, however. This is consistent with previous experimental findings. Convergence was attained in all experiments.*

4.6.2.3 Conclusion

These experiments contribute to:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**
3. **Proof of the robustness of the NPOST framework when partitioned.**

Having already established a scale foundation for the NPOST simulation, these experiment batches more truly represent the topological conditions expected in an Emerging System. There is high topological volatility with randomly persisting nodes.

The premise of the topology and stability experiments is to test the response of the NPOST simulation when an Emerging System is partitioned. Topological volatility occurs when nodes in the system fail to respond to requests for Trust Values and do not contribute to convergence persistently. The experiments test the convergence of the system under topological volatility – expansion and readjustment - with alternative Stability Strategies. It is possible that one or many nodes may join the System and then depart multiple times between iterations partitioning the initial System either temporarily or permanently, requiring the simulation to compensate for the changes. Stability strategies define how the simulation responds to these changes.

We introduced the concept that during topological change, the simulation undergoes multiple combinations of one, some or all of three phases of convergence:

1. *Uninhibited* – where the Emerging System retains a stable number of nodes;
2. *Readjustment* – where a node is sufficiently unresponsive, that it is no longer considered a part of the Emerging System and is excluded, and;
3. *Expansion* - where an additional node or cluster of nodes join the Emerging System.

The balance of evidence suggests a discernible influence on stability from the volatility of the System topology. Almost all configurations of the Stability Strategy contributed to a change in the convergence behaviour of the simulation while maintaining convergence. In only one experiment where Trust Values were directly corrected, was there divergence and there are possibilities to mature this approach for a better result. While expansion of the System influenced convergence, its effect was more apparent on the final Reputation Profile than the efficiency of convergence, though there was influence on both.

These experiments support the claim that a Stability Strategy is possible to optimise for a specific Emerging System by calibration of it component parts:

- Readjustment Schemes;
- Response Correction;
- Reinstatement Criteria;
- Convergence Conditions, and;
- Trust Value Correction.

The success of the Stability Strategy can only be measured against the requirements of the Emerging System to which it is applied. These experiments demonstrate a consistent relation between Stability Strategy and stability in the System supporting the hypothesis that the NPOST framework is a suitable approach to supporting consensus trust in Emerging Systems.

These conclusions support the conjecture that the NPOST framework is suitably stable and robust to support different and changing topological volatile Emerging System nodes under Expansion and Readjustment, regulated by a suitably optimised Stability Strategy.

4.6.3 Environment

4.6.3.1 Summary

Within any System, there are factors that affect the determination of Trust Values that are defined for the complete environment.

These factors can be universal such that all nodes in the System are affected similarly by their influence or they can be similar in nature but vary in weight between nodes. This symmetry alludes to a conservation of trust within the System. Environmental Factors can be normalised so that the total trust available in the System is always numerically one.

A System administrator might control *Environmental Factors* to alter the influence of the various facets of the final Trust Function. This could be in response to events outside the System such as a security breach or inside such as a change to the nature of an application. This calibration could be human or machine.

The significance then, of Environmental Factors is that they are not derived from the experiences of the nodes in the System or influenced by a consensus of nodes as we have seen already, rather, they are dictated at a System level.

These experiments were designed to determine the behaviour of the NPOST simulation under the influence of changing *Environmental Factors*. Environmental Factors can have two dimensional symmetry:

1. *Horizontal* – where a node's Environmental Factors always sum to exactly one, and;
2. *Vertical* – where every node shares the same Environmental factors.

A Trust Function can have dominant Environmental Factors, either strictly such that one Environmental Factor is greater than the others or just that there is an upper-bound. By varying these dimensions, we are able to establish the tolerance and capacity of the simulation and its suitability as a framework for use in Emerging Systems.

These experiments contribute to:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**
3. **Proof of the robustness of the NPOST framework with volatile Environmental Factors**.

The principal experimental variable is:

$X_6 R_2$	=	Environmental Factors ($\boldsymbol{E}$, R_2): • horizontally symmetric ($\sum_{j=1}^{n} e_{kj} = 1$) and horizontally non-symmetric ($\sum_{j=1}^{n} e_{kj} \neq 1$), • dominant ($e_i^* \geq e_{-i}$) and strictly dominant ($e_i^* > e_{-i}$), • vertically symmetric ($e_{kj} = e_{lj}$) or vertically non-symmetric (e_{kj} does not necessarily equal e_{lj}), and; • uniformly, pseudorandom

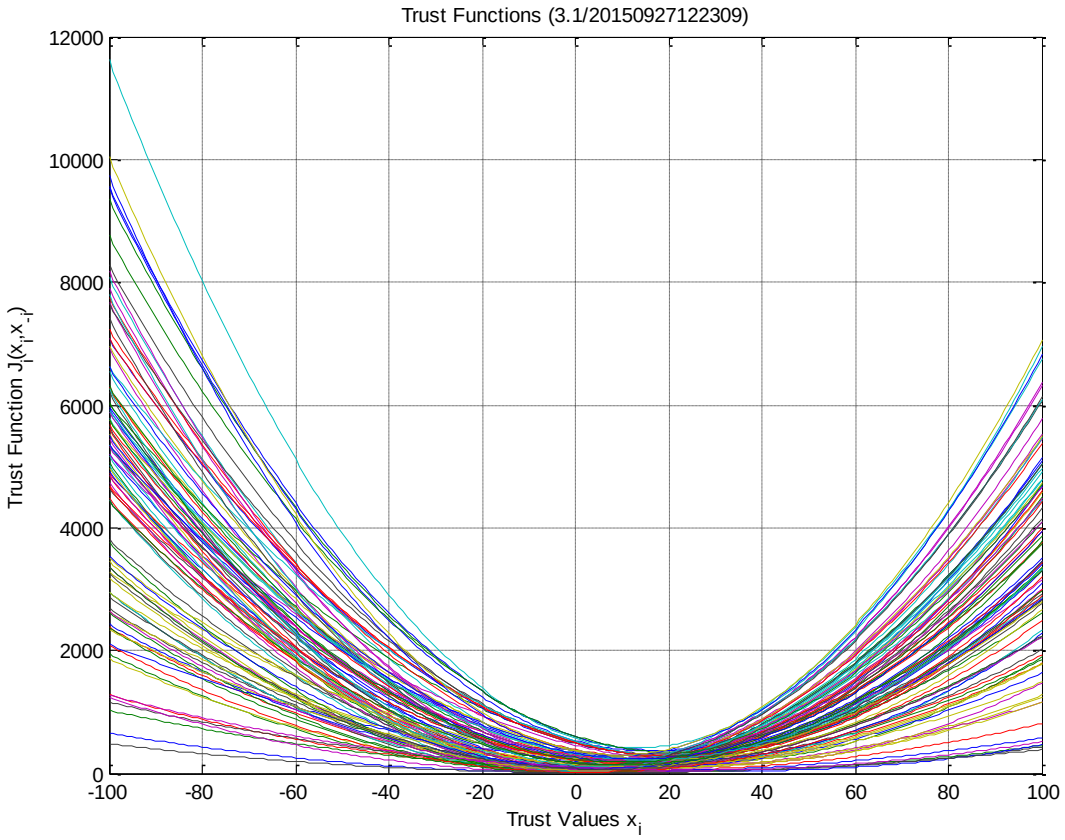


Figure 81 Typical Trust Function family of quadratic equations with environmental factor volatility plot

The Trust Function plot shows the family of quadratics, all varying by their Environmental Factors in contrast to the single line plot when all quadratics are similar (Figure 9) for Experiment Batch 1.1 and Experiment Batch 1.1.

The system of Trust Functions describes a multiple quadratics since the environmental factors are not symmetric:

$$X_6 R_2 \text{ with } e_{kj} \neq e_{lj}, \forall\, k, l, j$$

and:

$$\sum_{i=1}^{n} e_i \neq 1$$

4.6.3.2 Experiments

4.6.3.2.1 Experiment Batch 1.6

4.6.3.2.1.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : O_5 will be divergent as X_6R_2 symmetry is relaxed

H_1 : O_5 will be convergent as X_6R_2 symmetry is relaxed

for Stability Strategy (X_7), for randomly distributed, non-responsive and randomly non-responsive Environmental Factors ($\boldsymbol{E}$, R_2) (X_6R_2):

- horizontally symmetric ($\sum_{j=1}^{n} e_{kj} = 1$) and horizontally non-symmetric ($\sum_{j=1}^{n} e_{kj} \neq 1$),
- dominant ($e_i^* \geq e_{-i}$) and strictly dominant ($e_i^* > e_{-i}$),
- vertically symmetric ($e_{kj} = e_{lj}$) or vertically non-symmetric (e_{kj} does not necessarily equal e_{lj}), and;
- uniformly, pseudorandom,

with $0 \leq e_{kj} \leq 1$.

This experiment batch explores the four permutations of Environmental Factor symmetry configurations:

1. Horizontally and vertically symmetric (results from previous experiments);
2. Horizontally non-symmetric and vertically symmetric
3. Horizontally symmetric and vertically non-symmetric;, and;
4. Horizontally non-symmetric and vertically non-symmetric.

The Environmental Factors remain the same between iterations, retaining all their symmetric properties (an infinite iteration interval). It is expected that as the symmetry properties of the simulation are relaxed, the simulation will converge more slowly, and diverge. Stability will often not be attained before the iteration count upper-bound is breached.

4.6.3.2.1.2 Simulation Configuration

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	[-100, 100] ∈ $\mathbb{Z}$
X_4R_1	=	multFibonacci
X_5	=	1,000
X_6R_2	=	Experimental variable

Table 38 Simulation framework configuration for Experiment Batch 1.6

The algorithm was configured as follows:

X_7	=	None
X_8R_3	=	Static responsive (1)
X_9	=	0.0001
X_{10}	=	100
X_{11}	=	1

Table 39 Simulation JOR algorithm configuration for Experiment Batch 1.6

4.6.3.2.1.3 Results

3,000 independent experiments were conducted.

With an infinite iteration interval (∞), that is that the Environmental factors remain the same for all iterations, and variable Environmental Factor symmetry:

X_6R_2 Environmental Factor symmetry		O_5 Stability Convergence percentage (to 0 decimal places)	O_1 Iteration count (to 2 decimal places)	O_3 Computation real execution (seconds to 8 decimal places)	X_3 Initial Trust Value range (to 2 decimal places)	O_4 Final Trust Value range (to 2 decimal places)
Horizontal	Vertical					
x	*x*	*100*	*8.65*	*0.03239101*	*200.00*	*82.12*
	x	91	26.78	0.09240351	199.99	104.87
x		100	9.83	0.03408362	199.98	152.72
		100	14.24	0.04941439	199.98	190.17

Table 40 Results for infinite iteration interval with variable Environmental Factor symmetry

4.6.3.2.1.4 Discussion

To examine the nature of changing Environmental Factor symmetry and its influence on the stability of a System, we consider some specific examples.

For all the experiments conducted in this batch, the initial Trust Value range (X_3) remains consistent at 200 (to 0 decimal places). Most all experiments achieved stability (O_5) though there is some anomalistic behaviour in the only horizontally symmetric case, achieving only 91% convergence.

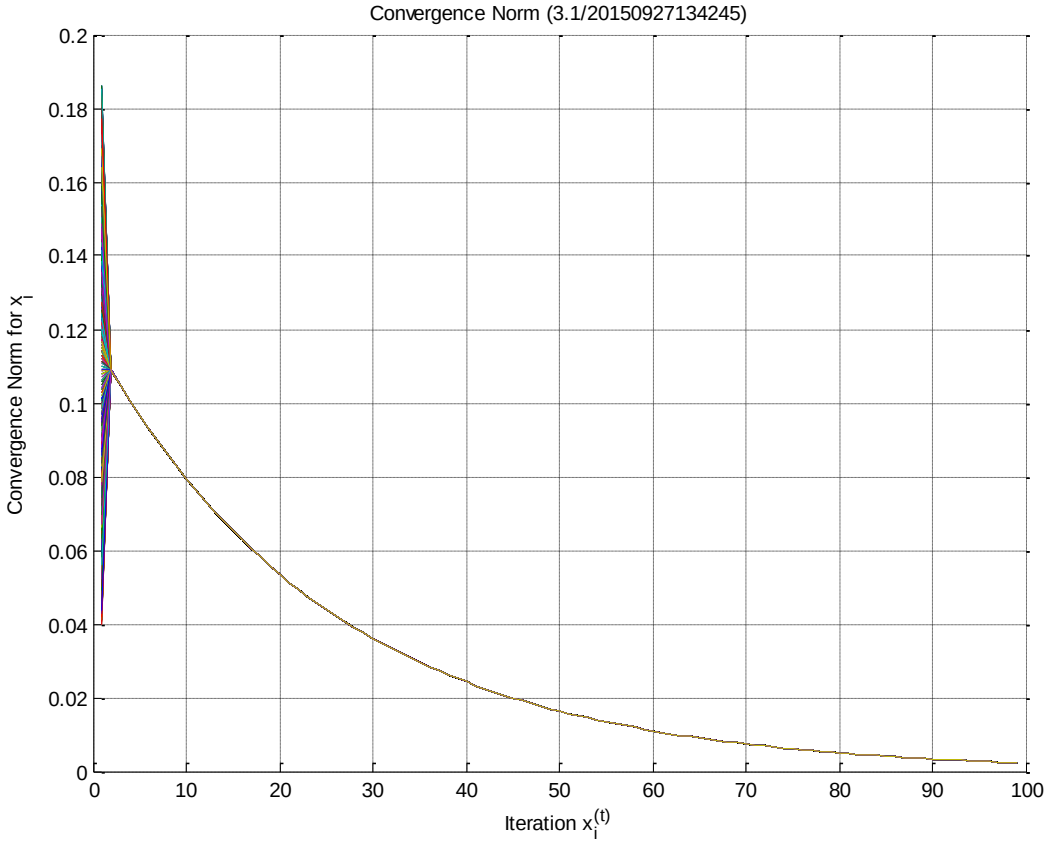


Figure 82 Convergence norm in the horizontally symmetric only Environmental Factors, divergent plot

In every one of these cases, convergence was very close to being achieved. It is reasonable to suggest that stability would have been achieved at a higher upper iteration bound (X_{10}) greater than 100. The mean final convergence norm for the divergent cases was:

$$\left\| x_i^{(99)} - x_i^{(100)} \right\| 0.057439055 > 0.0001 = \varepsilon$$

where i is a divergent case. In only 30 experiments of 1,000 (0.03%) was the norm above 0.002812393 after 100 iterations.

This consideration is further attested to by the matrix analysis which indicated that the System matrix was symmetric, full rank, strictly diagonally dominant therefore, positive definite with all non-zero diagonal elements (0

Convergence).

In the vertically and horizontally symmetric case, the mean range difference between the initial Reputation Profile (X_4R_1) and final Reputation Profile (O_4) is 58.94% (to 2 decimal places) and the standard deviation for Trust Values is 23.59 (to 2 decimal places).

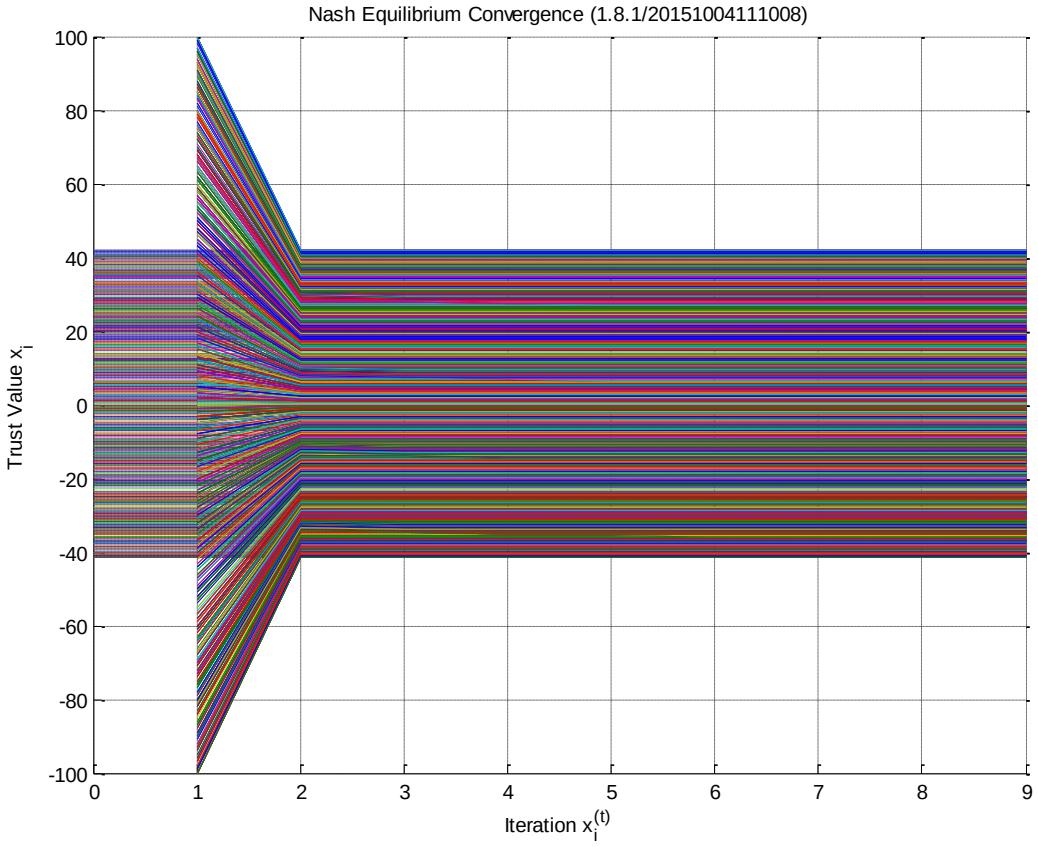


Figure 83 Typical Nash Equilibrium vertically and horizontally symmetric Environmental Factors plot

In the only vertically symmetric case, the mean range difference between the initial Reputation Profile (X_4R_1) and final Reputation Profile (O_4) is 47.56% (to 2 decimal places) and the standard deviation for Trust Values is 30.43 (to 2 decimal places).

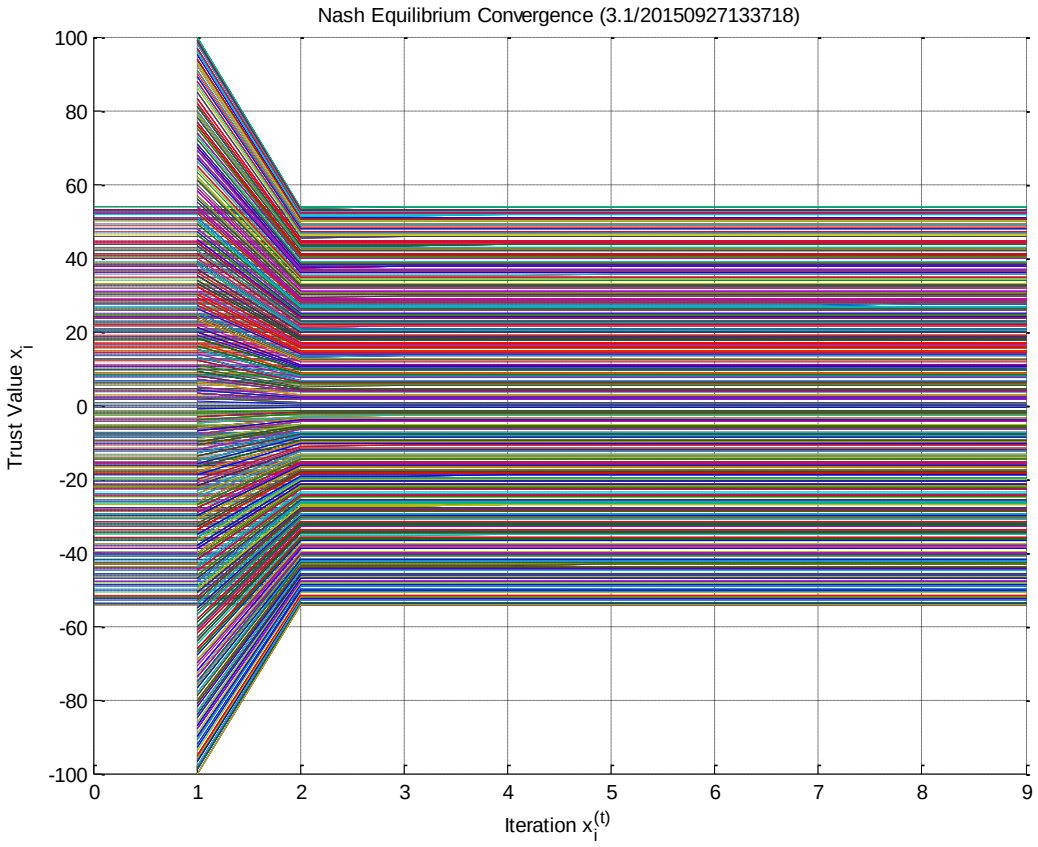


Figure 84 Typical Nash Equilibrium vertically non-symmetric and horizontally symmetric Environmental Factors plot

In the only horizontal symmetric case, the mean range difference between the initial Reputation Profile (X_4R_1) and final Reputation Profile (O_4) is 23.63% (to 2 decimal places) and the standard deviation for Trust Values is 22.04 (to 2 decimal places).

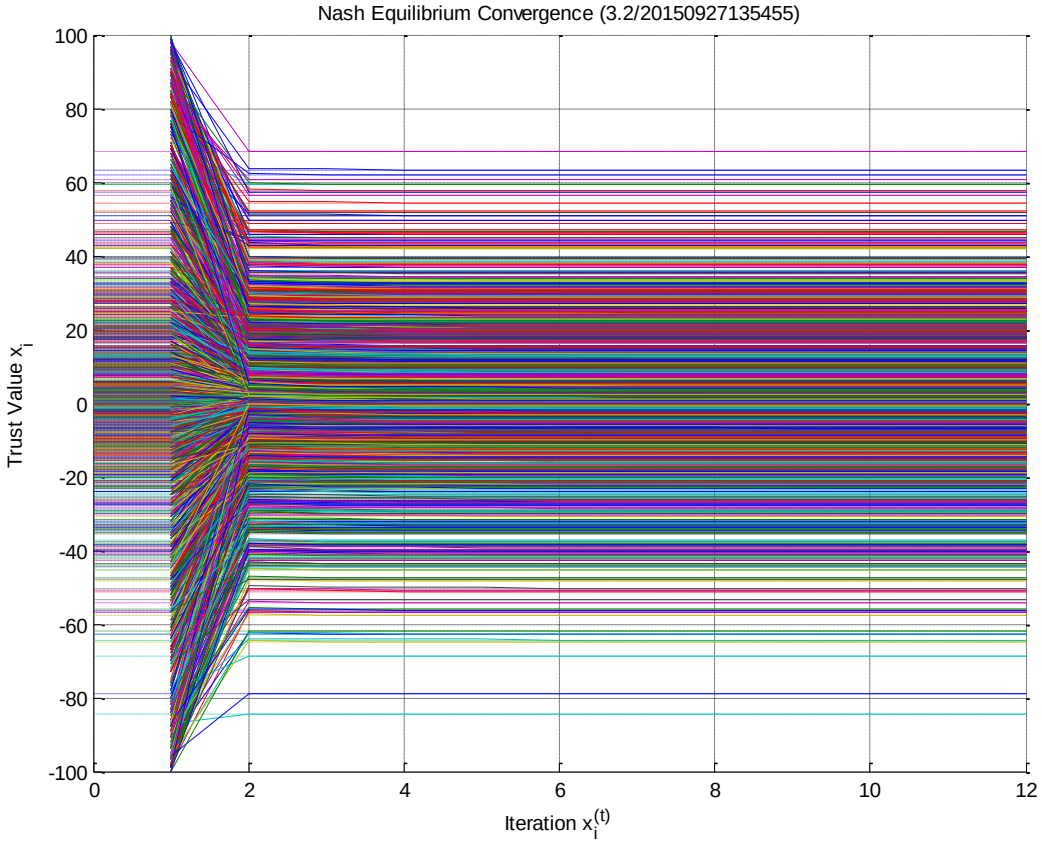


Figure 85 Typical Nash Equilibrium vertically symmetric and horizontally non-symmetric Environmental Factors plot

In the vertically and horizontally non-symmetric case, the mean range difference between the initial Reputation Profile (X_4R_1) and final Reputation Profile (O_4) is 4.90% (to 2 decimal places) and the standard deviation for Trust Values is 33.48 (to 2 decimal places).

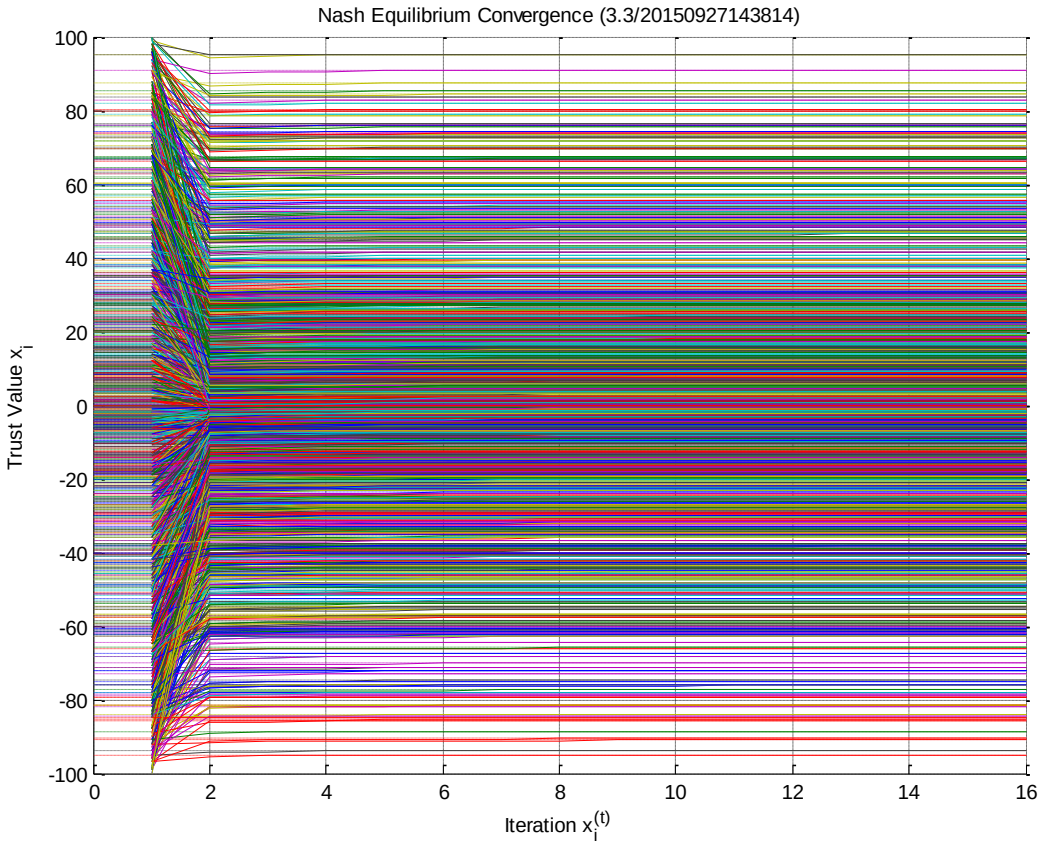


Figure 86 Typical Nash Equilibrium vertically and horizontally non-symmetric Environmental Factors plot

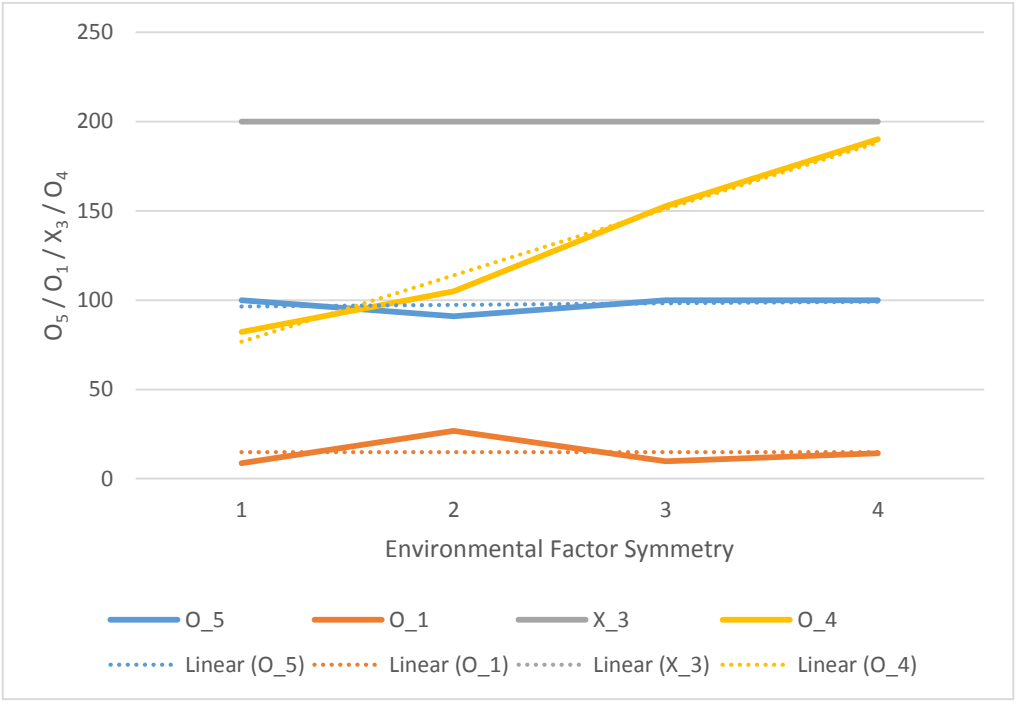


Figure 87 Typical Trust Function family of quadratic equations with environmental factor volatility plot

There appears no discernible pattern to either Iteration Count (O_1) or computation real time execution time (O_3), remaining similar throughout all experiments. Both are linearly consistent within experimental bounds, potentially only influenced by the random nature of the initial Reputation Profile (X_4R_1) and the Environmental Factors (X_6R_2). This is also the case for the standard deviation of final Trust Values.

Iteration count is a valid measure in these experiments because there are no Readjustment phases that artificially increase it when multiple readjustments occur during a single iteration. In the Topology and Scale experiments, the removal of a non-responsive node from the System during a readjustment phase, prompted the increment of the iteration count (O_1) as well as a true, full iteration of the algorithm (0

Discussion).

The most significant variation is in the final Reputation Profile Trust Values (O_4). As the symmetry of the Environmental Factors is relaxed, stability is still achieved but with significantly less correction to the initial Reputation Profile Trust Values in the final values. Convergence becomes less compact as the Environmental Factors influence the correction between iterations by different amounts.

X_6R_2 Environmental Factor symmetry		**O_4 Final Trust Value range standard deviation (to 2 decimal places)**	**O_4 Final Trust Value range difference (percentage to 2 decimal places)**
Horizontal	**Vertical**		
x	*x*	*23.59*	*58.94*
	x	30.43	47.56
x		22.04	23.63
		33.47	4.90

Table 41 Results for infinite iteration interval with variable Environmental Factor symmetry

This can be seen clearly from a typical Convergence Norm plot of a vertically and horizontally non-symmetric System.

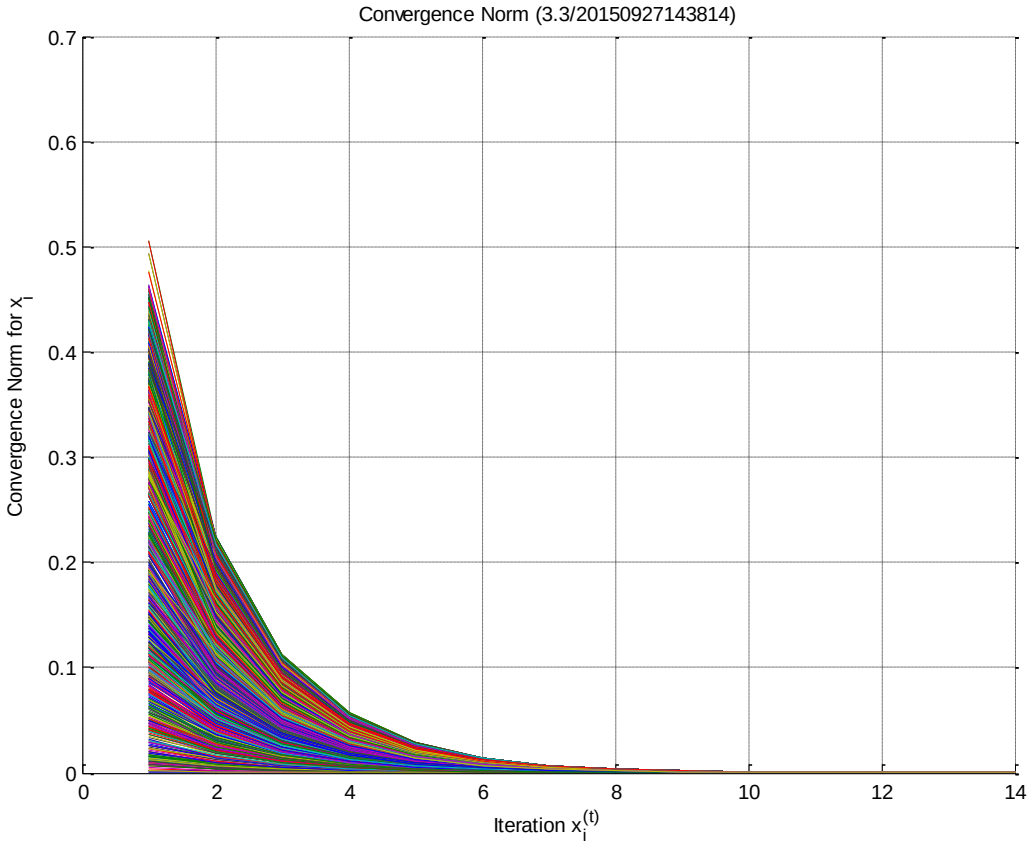


Figure 88 Typical Convergence Norm for non-symmetric Environmental Factors plot

The initial Reputation Profile Trust Values are densely similar and converge quickly to final Trust Values increasingly similar the original values as symmetry is relaxed.

For these experiments, no specific exploration was made for the range of the values of the Environmental Factors, they were consistently constrained: $0 \leq e_{kj} \leq 1$. It is reasonable to assert that the ratio of this range between Environmental Factors contributes to the stability of the System and not the values themselves. This is consistent with "conservation of trust" principle (3.5.2 Horizontally and Vertically Symmetric Environmental Factors). Similarly, we do not consider multi-component Trust Spaces (X_2), $\dim(\boldsymbol{M}) > 1$ as they are implicitly either isolated Systems that contribute a constant or variable to another System that can be considered a specialisation of an Environmental Factor. The implementation of the algorithm in the NPOST framework simulation makes these assertions possible (0

Numerical Example).

4.6.3.2.1.5 Conclusion

The experimental results are adduced to conclude the following:

> *For Experiment Batch 1.6, we must refute H_0 and accept H_1. There is no evidence to support the hypothesis that increased relaxation of the symmetry of Environmental Factors, causes the System to diverge. The experiments show that relaxed symmetry, while not increasing execution time or iteration count, contributes to a reduction in the range of Trust Values in the final Reputation Profile. Relaxed vertical Environmental Factors exhibit this stability behaviour more than horizontal.*

4.6.3.2.2 Experiment Batch 1.7

4.6.3.2.2.1 Operational Hypothesis

Experimental operational hypothesis:

H_0 : O_5 will be divergent as X_6R_2 symmetry is relaxed

H_1 : O_5 will be convergent as X_6R_2 symmetry is relaxed

for Stability Strategy (X_7), for randomly distributed, non-responsive and randomly non-responsive Environmental Factors ($\boldsymbol{E}$, R_2) (X_6R_2):

- horizontally symmetric ($\sum_{j=1}^{n} e_{kj} = 1$) and horizontally non-symmetric ($\sum_{j=1}^{n} e_{kj} \neq 1$),
- dominant ($e_i^* \geq e_{-i}$) and strictly dominant ($e_i^* > e_{-i}$),
- vertically symmetric ($e_{kj} = e_{lj}$) or vertically non-symmetric (e_{kj} does not necessarily equal e_{lj}), and;
- uniformly, pseudorandom,

with $0 \leq e_{kj} \leq 1$ and unfixed Environmental Factors.

Unlike Experiment Batch 1.6, the Environmental Factors in these experiments are not fixed between iterations. The experiments were conducted as before but with the Environmental Factors being recalculated at intervals of iterations. It is expected that stability within the System is only reached if it does so quickly enough between changes of Environmental Factors, during the periods where Environmental Factors remain consistent. As convergence is approached, the effect of the Environmental Factors on the Trust Values will diminish, increasing the chance of stability. This has to take place however, before the upper iteration bound is breached. As previously conjectured, it is expected that as the symmetry properties of the simulation are relaxed, the simulation will

converge more slowly, and diverge. Stability will often not be attained before the iteration count upper-bound is breached.

4.6.3.2.2.2 Simulation Configuration

The framework was configured as follows:

X_1	=	TF1
X_2	=	1
X_3	=	[-100, 100] ∈ $\mathbb{Z}$
X_4R_1	=	multFibonacci
X_5	=	1,000
X_6R_2	=	Experimental variable

Table 42 Simulation framework configuration for Experiment Batch 1.7

The algorithm was configured as follows:

X_7	=	None
X_8R_3	=	Static responsive (1)
X_9	=	0.0001
X_{10}	=	100
X_{11}	=	1

Table 43 Simulation JOR algorithm configuration for Experiment Batch 1.7

4.6.3.2.2.3 Results

4,800 independent experiments were conducted.

With vertically and horizontally symmetric Environmental Factors:

Iteration interval	O_5 Stability Convergence percentage (to 0 decimal places)	O_1 Iteration count (to 2 decimal places)	O_3 Computation real execution (seconds to 8 decimal places)	X_3 Initial Trust Value range (to 2 decimal places)	O_4 Final Trust Value range (to 2 decimal places)
∞	*100*	*8.65*	*0.03239101*	*200.00*	*82.12*
1	0	100	0.36690969	200.*00*	64.39
2	0	100	0.37525532	200.*00*	82.66
3	60	65.65	0.23354915	200.*00*	80.75
4	95	23.35	0.08411916	200.*00*	83.01
5	100	29.1	0.10410822	200.*00*	86.02
10	100	10.6	0.03793313	199.95	69.10

Table 44 Vertically and horizontally symmetric Environmental Factors with variable iteration intervals

With vertically symmetric and non-horizontally symmetric Environmental Factors:

Iteration interval	O_5 Stability Convergence percentage (to 0 decimal places)	O_1 Iteration count (to 2 decimal places)	O_3 Computation real execution (seconds to 8 decimal places)	X_3 Initial Trust Value range (to 2 decimal places)	O_4 Final Trust Value range (to 2 decimal places)
∞	*91*	*26.78*	*0.09240351*	*199.99*	*104.87*
1	0	100	0.35708907	199.95	83.85
2	5	95.1	0.34160293	200.*00*	91.23
3	50	68.45	0.25013755	200.*00*	97.51
4	90	37.35	0.13901531	200.*00*	111.10
5	100	32.8	0.11926312	200.*00*	95.83
10	100	17.65	0.06477773	200.*00*	96.55

Table 45 Vertically non-symmetric and horizontally symmetric Environmental Factors with variable iteration intervals

With vertically non-symmetric and horizontally symmetric Environmental Factors:

Iteration interval	O_5 Stability Convergence percentage (to 0 decimal places)	O_1 Iteration count (to 2 decimal places)	O_3 Computation real execution (seconds to 8 decimal places)	X_3 Initial Trust Value range (to 2 decimal places)	O_4 Final Trust Value range (to 2 decimal places)
∞	*100*	*9.83*	*0.03408362*	*199.98*	*152.72*
1	0	100	0.37472707	200.*00*	149.80
2	0	100	0.36704714	200.*00*	152.66
3	0	100	0.35867070	200.*00*	151.37
4	0	100	0.36690610	200.*00*	151.19
5	20	89	0.33017549	200.*00*	150.09
10	100	14.6	0.05352857	199.95	151.66

Table 46 Vertically symmetric and horizontally non-symmetric Environmental Factors with variable iteration intervals

With vertically and horizontally non-symmetric Environmental Factors:

Iteration interval	O_5 Stability Convergence percentage (to 0 decimal places)	O_1 Iteration count (to 2 decimal places)	O_3 Computation real execution (seconds to 8 decimal places)	X_3 Initial Trust Value range (to 2 decimal places)	O_4 Final Trust Value range (to 2 decimal places)
∞	*100*	*14.24*	*0.04941439*	*199.98*	*190.17*
1	0	100	0.38264844	199.95	190.19
2	0	100	0.36454654	199.95	190.93
3	0	100	0.35681124	200.00	192.24
4	0	100	0.35711752	200.00	189.45
5	0	100	0.37027084	200.00	188.57
10	15	92.4	0.34200602	199.95	191.52

Table 47 Vertically and horizontally non-symmetric Environmental Factors with variable iteration intervals

4.6.3.2.2.4 Discussion

As the symmetry of the Environmental Factors is relaxed, the percentage of cases where stability is attained decreases. In all cases, as the iteration interval increases, the likelihood of reaching stability increases. The more relaxed the symmetry, the smaller the influence of the iteration interval becomes. Aside from the anomalistic results for horizontal asymmetry, stability is always attained for an infinite iteration interval.

Iteration interval	O_5 Stability Convergence percentage (to 0 decimal places) (1)	O_5 Stability Convergence percentage (to 0 decimal places) (2)	O_5 Stability Convergence percentage (to 0 decimal places) (3)	O_5 Stability Convergence percentage (to 0 decimal places) (4)
1	0	0	0	0
2	0	5	0	0
3	60	50	0	0
4	95	90	0	0
5	100	100	20	0
10	100	100	100	15
∞ / 100	100	91	100	100

Table 48 Iteration interval against stability percentage

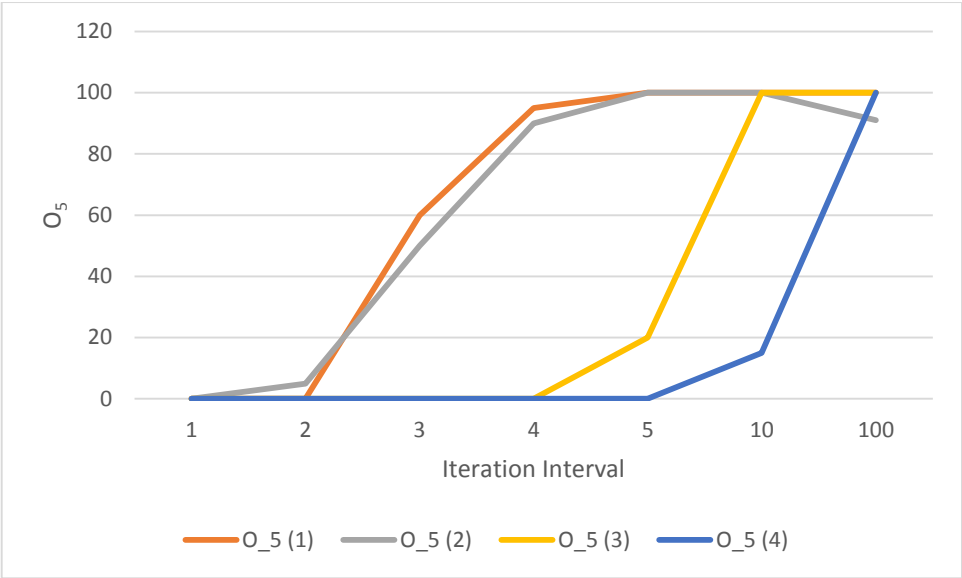


Figure 89 Iteration interval against stability percentage plot

The iteration interval can never be greater than the upper iteration bound, $\sup(t) \leq N\ (X_{10})$. With this value fixed for these experiments, $X_{10} = 100$, the infinite (∞) iteration interval is equivalent to the upper iteration bound. This is because the System will either achieve convergence within the iteration upper bound, or breach a convergence or non-responsive threshold but is never permitted to complete a greater number of iterations than the upper iteration bound, by definition. Any subset of the iteration upper bound, such as an iteration interval cannot be greater than its super set.

Wherever there is 0% convergence, there is an associated 100 iteration count. The higher the convergence percentage, the lower the iteration count – range of iterations between 48.19 and 86.66 (to 2 decimal places) with execution time increasing proportionally, between 0.17632367 and 0.31754500 seconds (to 8 decimal places).

Considering specific experiments demonstrates by observation, the changing volatility of the System under different Environmental Factor symmetry and iteration interval:

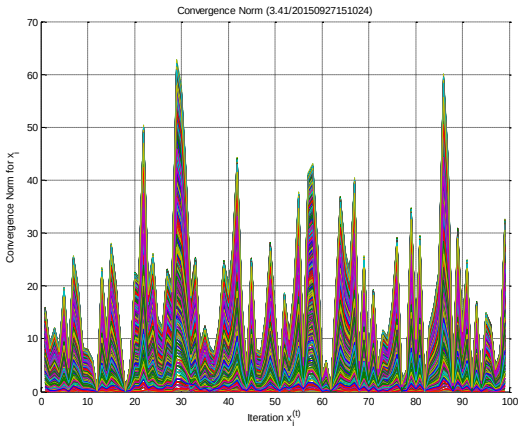


Figure 90 Convergence Norm with horizontal and vertical Environmental Factor symmetry, and iteration interval of 1

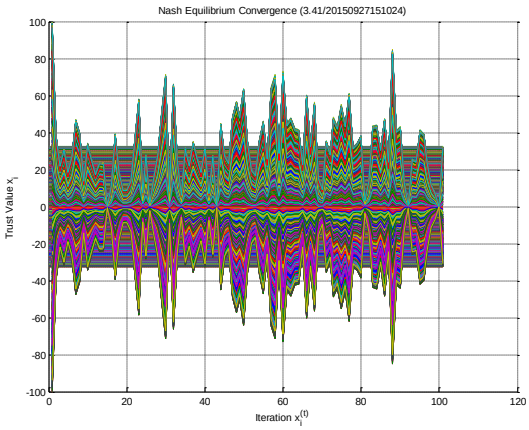


Figure 91 NE Convergence with horizontal and vertical Environmental Factor symmetry, and iteration interval of 1

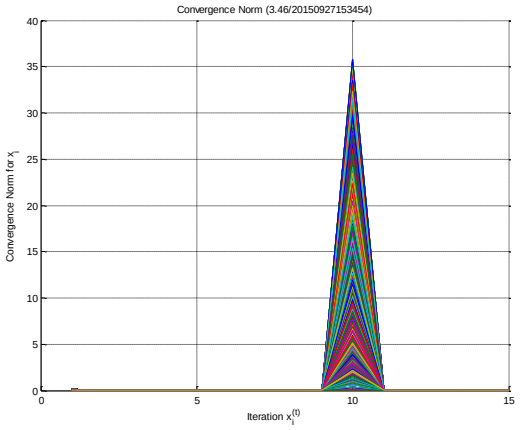


Figure 92 Convergence Norm with horizontal and vertical Environmental Factor symmetry, and iteration interval of 10

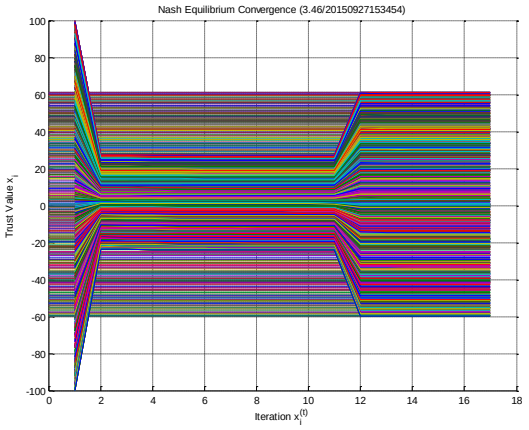


Figure 93 NE Convergence with horizontal and vertical Environmental Factor symmetry, and iteration interval of 10

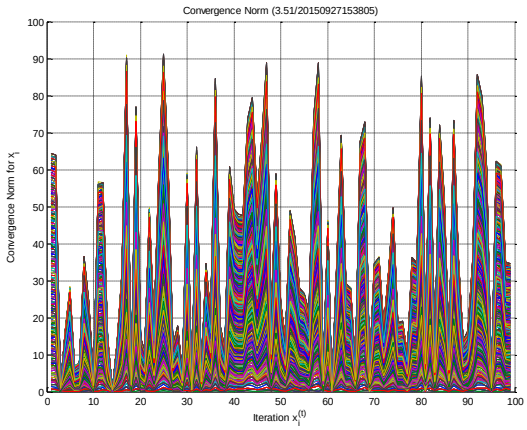


Figure 94 Convergence Norm with non-symmetric horizontal and symmetric vertical Environmental Factors, and iteration interval of 1

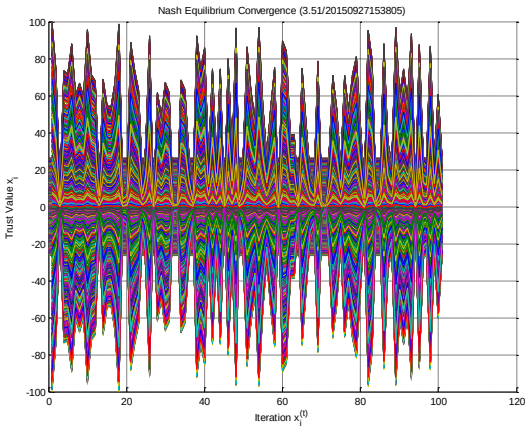


Figure 95 NE Convergence with non-symmetric horizontal and symmetric vertical Environmental Factors, and iteration interval of 1

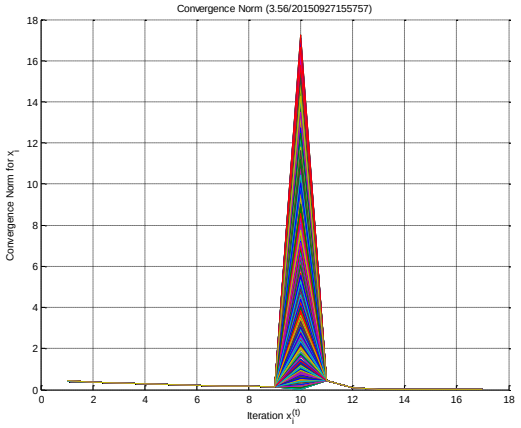


Figure 96 Convergence Norm with non-symmetric horizontal and symmetric vertical Environmental Factors, and iteration interval of 10

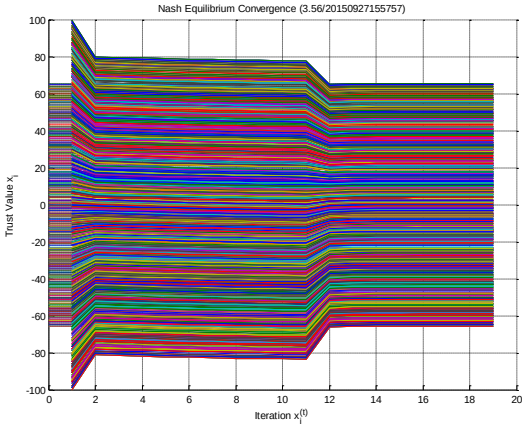


Figure 97 NE Convergence with non-symmetric horizontal and symmetric vertical Environmental Factors, and iteration interval of 10

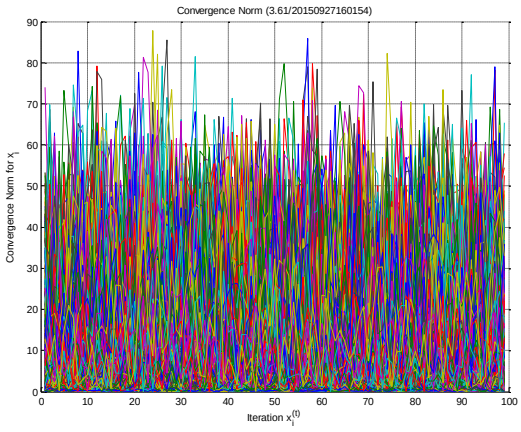


Figure 98 Convergence Norm with symmetric horizontal and non-symmetric vertical Environmental Factors, and iteration interval of 1

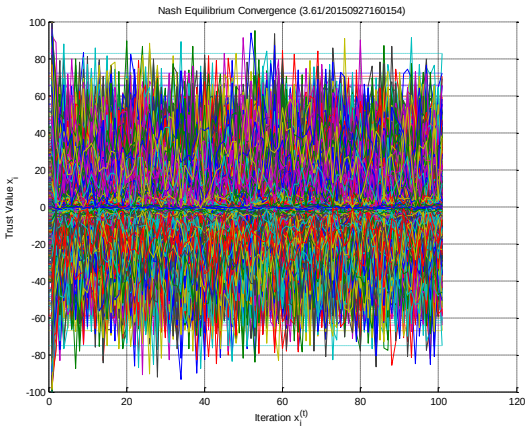


Figure 99 NE Convergence with symmetric horizontal and non-symmetric vertical Environmental Factors, and iteration interval of 1

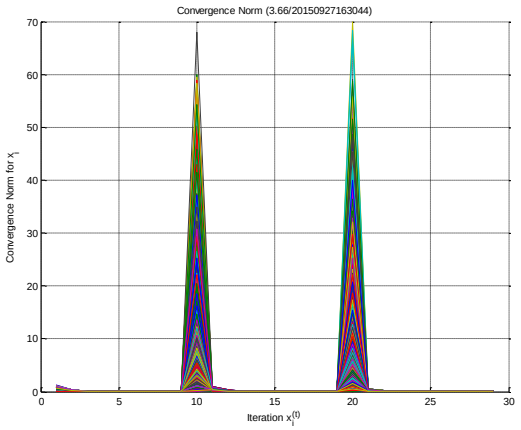


Figure 100 Convergence Norm with symmetric horizontal and non-symmetric vertical Environmental Factors, and iteration interval of 10

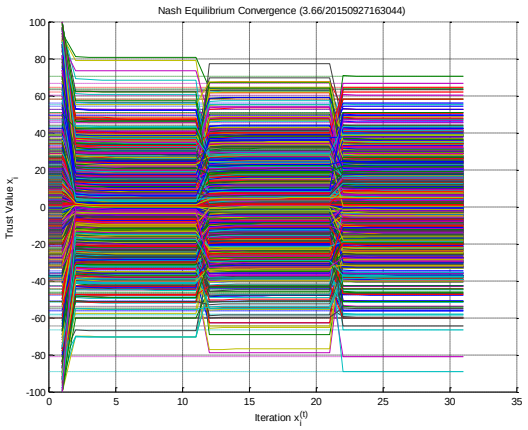


Figure 101 NE Convergence with symmetric horizontal and non-symmetric vertical Environmental Factors, and iteration interval of 1

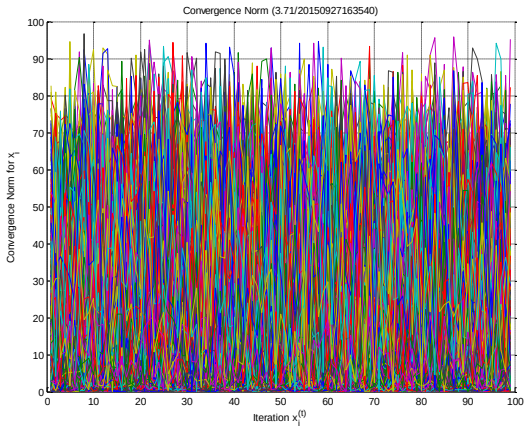


Figure 102 Convergence Norm with non-symmetric horizontal and vertical Environmental Factors, and iteration interval of 1

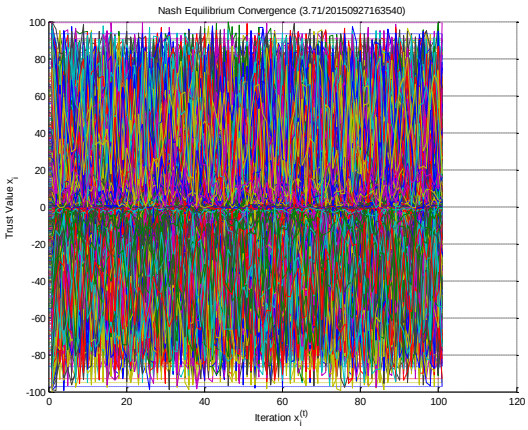


Figure 103 NE Convergence with non-symmetric horizontal and vertical Environmental Factors, and iteration interval of 1

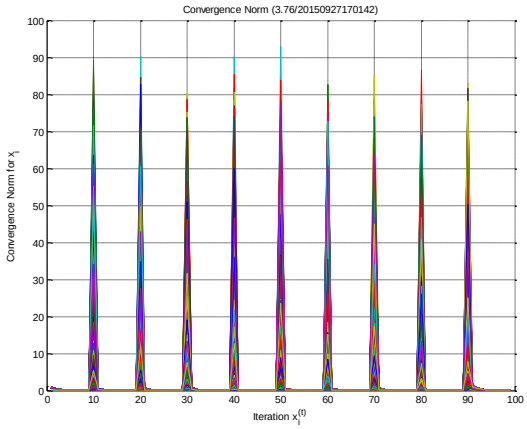


Figure 104 Convergence Norm with non-symmetric horizontal and vertical Environmental Factors, and iteration interval of 10

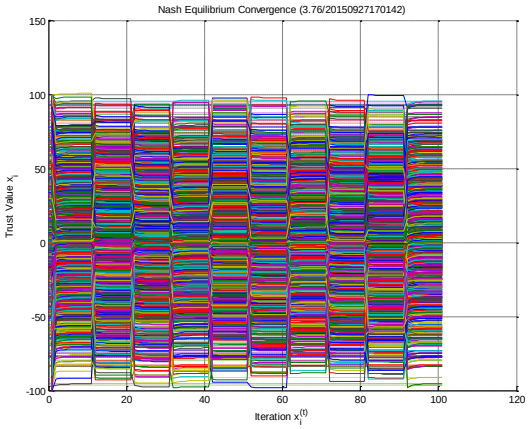


Figure 105 NE Convergence with non-symmetric horizontal and vertical Environmental Factors, and iteration interval of 10

From the plots, the volatility of Convergence Norm and Nash Equilibriums is apparent. As the iteration interval is increased, the plots become less erratic. However, when there is no symmetry to the Environmental Factors and the iteration interval is high as 10, the likelihood of the System attaining stability is extremely low (15%). Convergence is much more likely with symmetry and large iteration intervals.

The matrix analysis, referred to in Experiment Batch 1.6 is only applied to the initial set of Environmental Factors. It is only possible to know that the criteria for JOR algorithm convergence was met from the initial Environmental Factor matrix analysis. The analysis is not carried out for every iteration of the simulation so it is not possible to determine empirically, for every iteration if

convergence is possible or guaranteed. However, mathematically this is the case from the range of possible values of the Environmental Factors and the Trust Functions themselves. The simulation is often far too volatile for stability to be attained.

The result from Experiment Batch 1.6 that the range of Trust Values in the final Reputation Profile increases as Environmental Factor symmetry is relaxed, reoccurs in this experiment batch even with the inclusion of the iteration interval. The means and standard deviations for different symmetry permutations are consistent with the initial findings with infinite iteration interval, regardless of the interval tested (1, 2, 3, 4, 5 or 10):

X_6R_2 Environmental Factor symmetry		O_4 Final Trust Value range standard deviation (to 2 decimal places)	O_4 Final Trust Value range (percentage to 2 decimal places)
Horizontal	Vertical		
x	x	8.16	72.29
	x	8.83	97.28
x		1.14	151.36
		1.24	190.44

Table 49 Final Reputation Profile Trust Value range for Environmental Factor symmetry with variable iteration interval

In cases where stability is not achieved, it is misleading to cite the final Reputation Profile Trust Values as definitive as these values increase and decrease greatly as the simulation progresses. The final Reputation Profile achieved is reflective of Trust Values when the upper iteration bound was breached. No final Reputation Profile was actually obtained as a consensus of trust in the System.

4.6.3.2.2.5 Conclusion

The experimental evidence supports the conclusion:

> *For Experiment Batch 1.7, we must accept H_0 and refute H_1. The evidence supports the hypothesis that increased relaxation of the symmetry of Environmental Factors with variation between iterations, causes the System to diverge. The experiments show that the decreasing symmetry, while not increasing execution time or iteration count, contributes to a reduction in the range of Trust Values in the final Reputation Profile, consistent with previous experiments.*

4.6.3.3 Conclusion

These experiments contribute to this thesis by supporting:

1. **Proof of the suitability of the NPOST framework for Emerging Systems;**
2. **Proof of the practical implementation potential of the NPOST framework, and;**
3. **Proof of the robustness of the NPOST framework with volatile Environmental Factors.**

These experiments were designed to determine the behaviour of the NPOST simulation under the influence of changing *Environmental Factors*. Environmental Factors can have two dimensional symmetry:

1. *Horizontal* – where a node's Environmental Factors always sum to exactly one, and;
2. *Vertical* – where every node shares the same Environmental factors.

The significance of Environmental Factors is that they are not derived from the experiences of the nodes in the System or influenced by a consensus of nodes as we have seen already for Trust Values, rather, they are dictated at a System level.

Environment Factors in the simulation reflect the changing environment experienced by nodes in an Emerging System. The precise nature of the Environmental Factors is determined by the application of the framework to a specific Emerging System, and can be reflected in the structure of Trust Functions and change in the execution of the simulation algorithm. These experiments identify many configurations that can be applied to Emerging Systems that have low volatility (symmetric Environmental Factors and an infinite iteration interval) to high volatility (non-symmetric Environmental Factors with low iterative intervals). The practical suitability of the NPOST framework to Emerging Systems is dependent on the configuration best suited to support the Emerging System to which they are applied. The range of volatility has been explored in these experiments with divergent results indicating that not all configurations are universally applicable to all Emerging Systems.

These conclusions support the conjecture that the NPOST framework is suitably stable and robust to support different and changing Environmental Factors, with measurable constraints.

4.7 Conclusion

Consistent with the findings of the literature, this section set out to examine experimentally the behaviour of the Non-cooperative Programmable Open System Trust (NPOST) framework under simulated conditions, to determine the suitability of the framework to Emerging Systems.

To establish suitability, the contribution and significance of this section is to support:

1. **proof of the suitability of the NPOST framework for Emerging Systems;**
2. **proof of the practical implementation potential of the NPOST framework;**
3. **proof of the robustness of the NPOST framework:**
    a. **when scaled;**
    b. **when partitioned, and;**
    c. **under changing environmental influencing factors.**

The aim was to devise and carry out experiments that demonstrate the behaviour of the NPOST framework in the following categories and batches:

1. Scale:
    a. Node volume;
    b. Trust Value range.
2. Topology and Stability:
    a. Non-responsive nodes;
    b. Additional nodes.
3. Environmental Factors:
    a. Symmetry;
    b. Volatility.

The results of the experiments were interpreted to determine whether the NPOST framework was suitably reflective of Emerging Systems' general characteristics, and to which types of Emerging Systems could certain configurations be applied.

The evidence supports the conclusions that:

4.7.1 Scale

While the volume of nodes in the Emerging System does not significantly affect the number of computational cycles the framework has to carry out, it increases the total real time computation length significantly until the framework reaches its stress limit. At this point, the nodes are no longer able to fulfil the conditions for Nash Equilibrium (3.7.4 Sufficient Conditions). Type or range of Trust Values does not influence the execution of the simulation.

What is acceptable for these parameters is dependent on the application of the framework and the criteria that define the underlying Emerging System. As the volume of nodes in the System increases, performance degrades. Whether or not this is a reasonable degradation depends on the expectations of the Emerging System. An Emerging System could require high-performance, low volume, integrity dependence such as a financial transactional exchange or, it could only be necessary to establish consensus trust weekly, at very high volume with more relaxed data integrity in the case of a digital social network. The evidence attests that the framework has the ability to scale suitably for different configurations of Emerging System with relatively modest instruments and materials.

4.7.2 Topology and Stability

The evidence supports the conclusion that as the randomness of node responses increases, stability is harder to attain – higher iteration counts and longer computation time. The circumstances under which stability is determined to have been reached, greatly influences this conclusion and can also lead to trivial results.

Whether or not the final Reputation Profile is improved for certain configurations can only be assessed against the requirements of the Emerging System. When volumes of nodes are introduced into the System, there is a significant division between the convergent state of the nodes in the System before the nodes were introduced and the nodes introduced. The original nodes' Trust Values become very small, consistently and relatively much smaller than the new nodes'. Only the applications of the framework can decide if this is acceptable. Under threat of compromising the definition component of Emerging Systems that requires decentralised authority and control, there are instances where an Emerging System could remain static for a long time within the bounds of the System's environment. Physical network elements in a corporate infrastructure can remain consistent for long periods, particularly in organisations with a consistently operational posture, for instance. Determination of a suitable Stability Strategy is vital and the NPOST framework is suitably configurable to support topological variation.

4.7.3 Environment

Environmental factors greatly influenced final the Reputation Profile Trust Values while having little preponderance to disrupt stability. Horizontal asymmetry more than vertical, contributed to an increase in the range of the final Reputation Profile Trust Values, with a complete absence of symmetry contributing the most. Consistent with previous experimental findings, volatility, in this case of Environmental Factors between iterations, caused the most disruption to stability, and often resulted in divergence.

In all previous experiment batches, the results did not divulge much insight into the state of the final Reputation Trust Values. The Environment experiments produced the most surprising result in this observational variable.

More so than previous experiment's variables, Environmental Factors are specifically dictated by the underlying Emerging System. The significance of Environmental Factors is that they are not derived from the experiences of the nodes in the System or influenced by a consensus of nodes, they are dictated at a System level. Unlike previous experiments, the volatility of Environmental Factors caused the simulation to diverge. Measuring an Emerging System's nature and range of Environmental Factors is significant in assuring a high degree of certainty that the framework is applicable.

5 Conclusion

5.1 Statement of Claim

This work postulates that that there is no comprehensive definition of contemporary enterprise systems. Current comparable definitions do not consider all of the characteristics of an open system with highly programmable nodes that are not beholden to central governance that consequently, cannot be assumed cooperative. Without an established definition, the Community is unable to distinctly identify systems of this type and develop them accordingly.

This work provides the definition of *Emerging Systems* to address the research gap identified, and to demonstrate theoretically and experimentally, the suitability of a supporting trust (Non-cooperative Programmable Open System Trust (NPOST)) framework for the definition.

5.2 Findings

The over-arching findings of this work and contributions to knowledge, are that it:

1. **establishes the need for a definition of a sui generis class of computational system designed to support the nature of the contemporary enterprise and provides it – *Emerging Systems*;**
2. **supports the definition of Emerging Systems with a mathematical underpinning and nomenclature, so that they can be described and explored universally in a well-defined manner;**
3. **validates the need and suitability of a non-cooperative game theoretical trust framework to support the reliable formation of an Emerging System, and;**
4. **develops and experimentally examines the trust framework to establish its suitability to specifically support the characteristics of Emerging Systems.**

5.3 Contributions and Originality

From the Literature the contribution and originality of this work supports:

1. the need for and the definition of *Emerging Systems,* and;
2. a Trust Framework that:
    a. is suitable for Emerging Systems
    b. can be implementable to support the application layer, and;
    c. is specifically, non-cooperative.

Emerging Systems are characterised by:

- Decentralisation;
- High distribution;
- Self-configuration;
- Self-regulation;
- Non-cooperation;
- Pervasiveness;
- Dynamic open ad-hoc ("for this" purpose) topology – non-generalisable;
- No fixed infrastructure;
- Wireless connectivity;
- High scalability, and;
- Consisting of highly reprogrammable nodes.

The Mathematical Framework, supports the contribution to:

3. demonstrate the formulation of mathematical constructs can define a trust nomenclature as a foundation for a trust framework;
4. assist proof of the suitability of rigorous applications of non-cooperative game theoretical techniques to establish stability and equilibrium applied to the constructs;
5. assist proof of the suitability of iterative methods and algorithms as the computational mechanics of these techniques for a trust framework, and;
6. derive a well-constructed cost function as a candidate for the experimental analysis of the trust framework.

The Experimental Analysis, contributes support for:

7. proof of the suitability of the NPOST framework for Emerging Systems;
8. proof of the practical implementation potential of the NPOST framework;

9. proof of the robustness of the NPOST framework:
    d. when scaled;
    e. when partitioned, and;
    f. under changing environmental influencing factors.

5.4 Limitations

5.4.1 Emerging Systems

The definition of Emerging Systems assumes a notion of refutation or transcension of previous definitions, post hoc ergo ultra hoc. The reason this is possible is because of the fluid and transitional nature of the subject domain the definition aims to clarify and categorise. Once a definition is established, it is fixed in time and therefore susceptible to the same transience that lead to its formulation in the first place. The Emerging Systems definition aims to encompass current and future states, as they are currently predicted, for enterprise systems, but there can be no assurance that it will be enduring. Developments in technology will continue to question the applicability of the definition; Low-power, long-range Bluetooth "beacons" (Beacons 2015) for instance, facilitate location-awareness or indoor proximity for systems, but they do not meet the highly-programmable requirement of the definition to constitute a "node" in an Emerging System, as such. It could be argued that "highly-programmable" is a subjective criteria that requires further clarification. Recent innovations in "wearable" technology (Page 2015) – "smart" watches (Google 2015) and glasses (Google 2015) - pose similar threats to the validity of the definition in ways as yet, unrealised.

The burgeoning presence of the Internet of Things (Tatnall 2015) – hopefully not to the detriment to mankind as Tatnall (2015) whimsically and ominously allude – inevitably introduces new communication methods and types of nodes to the enterprise. To sustain the relevance of the definition or expand upon it, these developments should be incorporated.

5.4.2 Mathematical Framework

The Mathematical Framework provides a sufficient foundation for the purposes of this work but is by no means, complete. The analysis was conducted on a single Trust Function, with the strong emphasis on being able to describe the function in terms of the Mathematical Framework nomenclature, prove that it was suitable for experimental analysis and demonstrating the behaviour of the framework under conditions characteristic of Emerging Systems. This work naturally lends itself to the analysis of further Trust Functions, with specific application to permit classification, with:

- representation of more complex relationships between nodes in a system;
- richer and higher component Trust Spaces and sub-spaces that represent differences of consensus opinion within the same system;
- more granular representation of opinion and reputation;
- more environmental factors with variable influences and alternative symmetries, and;
- composition of more complex functions beyond convexity, within ranges that continue to assure a unique solution.

5.4.3 Experimental Analysis

The analysis here is limited by and could be extended to include:

5.4.3.1 Scale

Most foundational limitations to the experiments are demonstrable through the Scale experiments as they were primarily designed to not only test the limits of the framework but of the simulation too. Consequently, much of the underlying modifications and enhancements considered here, would also effect all other experiment categories. It is more felicitous to ensure that the experiments are consistent between categories so that they may be reasonably compared.

- Larger Systems – greater node volume – refine the increments of volume to determine the Stress threshold more precisely;
- Larger ranges of Trust Values – positive and negative;
- Further Trust Value types – imaginary, natural, discrete natural numbers;
- Comparison of random generators as alternatives to Multiplicative Lagged Fibonacci:
    - Mersenne Twister;
    - SIMD-oriented Fast Mersenne Twister;
    - Combined Multiple Recursive;
    - Legacy MATLAB® 5.0 uniform generator;
    - Legacy MATLAB 5.0 normal generator, or;
    - Legacy MATLAB 4.0 generator.
- Experiment instruments:
    - Higher performing simulation hardware;
    - Higher performing simulation software – MATLAB R2015b;
    - Reduce competing resources:
        - Remove background processes;
        - Allocate memory;
        - Allocate CPU.

    - Physical implementation;
    - Wide Area Network (WAN) simulation;
    - Cloud compute implementation.
- Algorithm modification:
    - Alternative algorithm types, suited to specific Emerging Systems (3.8.2 Iterative Methods):
        - Arnoldi;
        - BiCGSTAB (BiConjugate Gradient STABilized);
        - Conjugate gradient;
        - Gauss-Seidel;
        - GMRES (Generalized Minimum RESidual);
        - Lanczos;
        - MINRES (MINimal RESidual);
        - QMR (Quasi Minimal Residual);
        - QMR;
        - SOR, and;
        - TFQMR (Transpose-Free QMR).
    - Variable relaxation factor – for these experiments, fixed at 1;
    - Implementation modification (Australia 2015):
        - Parallel processing (Australia 2015);
        - Preallocate matrix sizes. For large arrays, MATLAB must allocate a new block of memory and copy the older array contents to the new array as it makes each assignment;
        - Use functions instead of scripts;
        - Prefer local functions over nested functions;
        - Modular programming;
        - Vectorise — Reduce loop-based code;
        - Place independent operations outside loops — If code does not evaluate differently with each “for” or “while” loop iteration, move it outside of the loop to avoid redundant computations;
        - Create new variables if data type changes —Changing the class or array shape of an existing variable takes extra time to process;
        - Use short-circuit operators — Use short-circuiting logical operators, && and || when possible. Short-circuiting is more efficient because MATLAB

evaluates the second operand only when the result is not fully determined by the first operand;

        - Avoid global variables — Global variables can decrease performance;
        - Avoid overloading built-ins — Avoid overloading built-in functions on any standard MATLAB data classes;
        - Avoid using "data as code" — Load variables instead of executing code to generate them.
    - Consider algorithms beyond one “hop” – comparative trust between clusters to improve efficiency at high volume. Adopt a less parochial assumption of trust between nodes.

5.4.3.2 Topology and Stability

- Parallel removal of non-responsive nodes – Readjustment phases;
- Context aware dynamic configuration (Dynamic Stability Strategy):
    - Dynamic Readjustment Schemes;
    - Dynamic Response Correction;
    - Dynamic Reinstatement Criteria;
    - Dynamic Convergence Condition;
    - Dynamic Trust Value Correction.
- Extend exploration of Stability Strategy functions;
- Alternative node responsiveness allocation:
    - Higher non-responsive node percentage;
    - Alternative random generation;
    - Removal and reintroduction of the same node or cluster of nodes.
- Introduction of new nodes based on variable criteria:
    - Multiple Expansion phases;
    - Random;
    - Repetition.
- Extend Trust Value correction techniques:
    - Decay;
    - Regret;
    - Malice;
    - Gullibility;
    - Preference;
    - Historical bias.

- Comparative iteration count – currently every Readjustment phase is considered a complete single iteration when the phase could have in fact removed multiple nodes in the same iteration, and;
- Variable Scale and Environment Factors.

5.4.3.3 Environment

- Improve Environmental Factor reporting:
    - Matrix analysis after each iterative interval;
    - Record changing variable Environmental Factors.
- Dynamic iterative range;
- Dynamic Convergence Condition;
- Variation of the Environmental Factor value range;
- Higher dimension Trust Spaces, and;
- Variable Scale and Topology and Stability.

In general, there is a mathematically sublime (Smith 2015) (Carson and Shabel 2015) (Carson and Shabel 2015) number and type of possible Trust Functions that could have been tested through these experiments. The genesis of the framework, however, was that while this work describes and defines Trust Functions in principle, it should be flexible enough to support any Trust Function with convergent properties. For this reason, the experiments were design to test the framework and not specifically Trust Functions en masse. Further types of Game Theory could also be applied for instance, co-operative or Gaussian.

The experiments were carried out in discrete categories (Scale, Topology and Stability, and Environment) but could be combined to understand the influential effects of one type on another. While it certainly admits of closer analysis, it is reasonable to suggest that many of the results of these experiments can be extrapolated from the phases of the experiments that were carried out. By breaking the experiments into named ephemeral phases, it is reasonable to assert that a blended experiment is in fact, a series of concatenated phases the results for which, we already have. We cannot be certain though how one category might subvert another and what incongruity there might be in practice.

The conclusions of the Experimental Analysis are consistently tempered by the Emerging System characteristics to which the NPOST framework is to be applied and how it is configured. To extend this work, it would be appropriate to study specific cases to calibrate their circumstances to the configuration of the framework to determine what is most appropriate. In general terms, they do corroborate the contribution and significance of the chapter.

5.5 Future Research

Beyond exploring the limitations of this work, further research predominantly lies in the application of the NPOST framework to enterprises that exhibit the characteristics of Emerging Systems and evaluating how suitable the framework is in practice. This research can be extended to consider case studies of specific applications of the framework, in areas such as:

1. Social networks Ziegler and Golbeck (2015) posit algorithms to determine interpersonal trust in social networks;
2. Online ranking - Zervas, Proserpio et al. (2015) examine the role of trust for people identifying and selecting accommodation in the Airbnb (Airbnb 2015) online service;
3. Games - Clark, Leavitt et al. (2015) explore the concept of "social capital" derived from trust in online games;
4. Semantic Web - Wang, Huang et al. (2015) propose a "trust-aware" composite sematic web service selection approach that permits consumers to select and use services from unfamiliar sources without specific configuration;
5. Internet of Things (IoT) - Alshehri and Hussain (2015) conduct a comparative analysis of the role of trust management for the IoT. Tatnall (2015) adopt a socio-technical view of Actor-Network Theory and its application to the IoT;
6. Search Engines – Burguet, Caminal et al. (2015) examine the potential bias in search engines towards paid advertisements and how much their results can be trusted, and;
7. Cloud computing Noor, Sheng et al. (2015) propose "CloudArmor" as a reputation-based trust management system for cloud services (see 3.9.2.6 CloudArmor).

Worthy of specific consideration is the role of trust in Payments. Developed to address the perceived weaknesses in Internet commerce, Nakamoto (2008) proposed the "Bitcoin" (Bitcoin Foundation 2015) (Martins and Yang 2011) peer-to-peer electronic cash system. Financial exchanges are almost exclusively transacted through a financial institution acting as a trusted third-party, and charging transaction and validation fees. The Bitcoin system allows online payments to be sent directly from one party to another without going through a financial institution, generally without charge. The approach uses digital signatures and an ongoing chain of hash-based "proof-of-work" that forms a record that cannot be changed without carrying out the machine work again. Bitcoin transactions are recorded in a public ledger called the "block chain" (Barber, Boyen et al. 2012).

The system is based on a cryptographic proof as trust. Nakamoto (2008) specifically explores a solution to the "double-spending problem" (Karame, Androulaki et al. 2012) using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of

transactions, to ensure that transactions are unique so that currency can only be spent synchronously. The security of the system is assured as long as the legitimate nodes in the system collectively command more CPU power (machine work) than any distributed malicious entity.

The payment system is structurally under-pinned by an Emerging System where nodes are highly computationally capable (some nodes generate or contextually, "mine" the Bitcoin currency to contribute to the economy requiring substantial compute power to determine a SHA-256 (Gilbert and Handschuh 2004) hash from a "nonce" (Rogaway 2004) with a difficultly target), with their communication network resilient under volatile ad-hoc topological change and without any central intermediary authority. As with the approach proposed in this work, rules and incentives are enforced with a consensus mechanism (Eyal and Sirer 2013). Trust is established by the volume of work carried out in a form of "race" where it is assumed that the "leading" opinion is assured.

Without a central authority, significant legal (Dougherty 2015), and regulatory (United States District Court 2013) (U.S. Government Accountability Office 2013) questions have arisen with the U.S. Treasury classifying Bitcoin as a, "…convertible decentralized virtual currency" (FinCen 2013). The Bitcoin system has been trading live since 2009 with a real-value exchange rate (@coindesk 2015) and as such, is a significant application of Emerging Systems as it is a radical departure from the traditional approaches to financial exchange. Whether Bitcoin fully establishes itself as a viable and widely adopted currency, is yet to be determined (Barber, Boyen et al. 2012) but it is an indication of the possible applications of Emerging Systems to the fundamental global infrastructure.

6 Bibliography

@coindesk (2015). "Bitcoin Price Index - Real-time Bitcoin Price Charts."

Aberer, K. and Z. Despotovic (2001). Managing trust in a peer-2-peer information system. Proceedings of the tenth international conference on Information and knowledge management, ACM.

Acma (2014). "ACMA." **2014**.

Adams, W. J. and N. J. Davis IV (2005). Toward a decentralized trust-based access control system for dynamic collaboration. Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC, IEEE.

Ahn, T.-K., E. Ostrom and J. Walker (2003). "Incorporating motivational heterogeneity into game theoretic models of collective action." Public Choice **117**(3-4).

Airbnb (2015). "Holiday Rentals, Homes, Apartments & Accommodation - Airbnb Australia." **2015**.

Akcigit, N. B. a. U. (2004). "Math for Economists - Handout on Second Order Conditions." Retrieved 7/5/2013, 2013, from http://web.mit.edu/14.102/www/notes/soc.pdf.

Akyildiz, I. F., X. Wang and W. Wang (2005). "Wireless mesh networks: a survey." Computer networks **47**(4): 445-487.

Allais, M. (1979). The so-called Allais paradox and rational decisions under uncertainty, Springer.

Alpcan, T. and T. Başar (2005). "A globally stable adaptive congestion control scheme for internet-style networks with delay." Networking, IEEE/ACM Transactions on **13**(6): 1261-1274.

Alpcan, T., T. Başar, R. Srikant and E. Altman (2001). CDMA uplink power control as a noncooperative game. Decision and Control, 2001. Proceedings of the 40th IEEE Conference on.

Alpcan, T., C. Rencik, A. Levi and E. Sava (2010). A game theoretic model for digital identity and trust in online communities. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security. Beijing, China, ACM**:** 341-344.

Alshehri, M. D. and F. K. Hussain (2015). A Comparative Analysis of Scalable and Context-Aware Trust Management Approaches for Internet of Things. Neural Information Processing, Springer.

Altman, E., T. Boulogne, R. El-Azouzi, T. Jiménez and L. Wynter (2006). "A survey on networking games in telecommunications." Computers & Operations Research **33**(2): 286-311.

Apple (2014). "Apple Developer." **2014**.

Arrow, K. J. and G. Debreu (1954). "Existence of an equilibrium for a competitive economy." Econometrica: Journal of the Econometric Society: 265-290.

Arrow, K. J. and L. Hurwicz (1958). "On the stability of the competitive equilibrium, I." Econometrica: Journal of the Econometric Society: 522-552.

Ashton, K. (2014). "That 'Internet of Things' Thing." RFID Journal **2014**.

Aumann, R. and A. Brandenburger (1995). "Epistemic conditions for Nash equilibrium." Econometrica: Journal of the Econometric Society: 1161-1180.

Australia, M. (2015). "Parallel Computing Toolbox - MATLAB." **2015**.

Australia, M. (2015). "Techniques to Improve Performance - MATLAB & Simulink." **2015**.

Ausubel, L. M. and R. J. Deneckere (1993). "A generalized theorem of the maximum." Economic Theory **3**(1): 99-107.

Awad, S. (2010). "Introduction to Matlab: Random Numbers." from http://ieeeumd.com/member-downloads/fall-2010/matlab-october-2010/11-matlab-random-numbers/download.html.

Axelrod, R. (1987). "The evolution of strategies in the iterated prisoner's dilemma." The dynamics of norms: 1-16.

Balagurusamy (1999). Numerical Methods, McGraw-Hill Education (India) Pvt Limited.

Bamberger, W. (2010). "Interpersonal Trust – Attempt of a Definition." **2014**.

Baras, J. S. and T. Jiang (2004). Cooperative games, phase transitions on graphs and distributed trust in manet. Decision and Control, 2004. CDC. 43rd IEEE Conference on, IEEE.

Baras, J. S. and T. Jiang (2005). Cooperation, trust and games in wireless networks. Advances in Control, Communication Networks, and Transportation Systems, Springer**:** 183-202.

Barber, S., X. Boyen, E. Shi and E. Uzun (2012). Bitter to better—how to make bitcoin a better currency. Financial Cryptography and Data Security, Springer**:** 399-414.
Barnes, S. J. (2003). "Enterprise mobility: concept and examples." International Journal of Mobile Communications **1**(4): 341-359.
Barr, N. (2012). Economics of the Welfare State, OUP Oxford.
Basagni, S., M. Conti, S. Giordano and I. Stojmenovic (2004). Mobile ad hoc networking, John Wiley & Sons.
Başar, T. and P. Bernhard (2008). H-infinity optimal control and related minimax design problems: a dynamic game approach, Springer.
Başar, T. and G. J. Olsder (1999). Dynamic Noncooperative Game Theory, Society for Industrial and Applied Mathematics.
Başar, T., G. J. Olsder, G. Clsder and G. J. Olsder (1995). Dynamic noncooperative game theory, SIAM.
Basole, R. C. (2008). "Enterprise mobility: Researching a new paradigm." Information-Knowledge-Systems Management **7**(1, 2): 1-7.
Bazaraa, M. S., H. D. Sherali and C. M. Shetty (2006). Nonlinear Programming: Theory and Algorithms, Wiley.
Beacons, B. (2015). "Bluetooth Beacons | Buy iBeacon & Eddystone Beacons." **2015**.
Beck, M. (2015). "Australia's infrastructure." Ecodate **29**(2): 2.
Begelfor, E. and M. Werman (2006). Affine invariance revisited. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, IEEE.
Berners-Lee, T. (1989). "Information Management: A Proposal." **2014**.
Bertsekas, D. P., A. Nedić and A. E. Ozdaglar (2003). Convex Analysis and Optimization, Athena Scientific.
Bertsekas, D. P. and J. N. Tsitsiklis (1997). Parallel and Distributed Computation: Numerical Methods, Athena Scientific.
Bhattacharjee, B. (2015). "Project NICE at University of Maryland." **2015**.
Bianchi, M. and R. Pini (2003). "A note on stability for parametric equilibrium problems." Operations Research Letters **31**(6): 445-450.
Binmore, K. and J. Davies (2001). Calculus: Concepts and Methods, Cambridge University Press.
Bitcoin Foundation (2015). "Bitcoin - Open source P2P money."
Bjørner, N. and L. d. Moura (2014). "Z3." **2014**.
Blaze, M., J. Feigenbaum and J. Lacy (1996). Decentralized trust management. Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on, IEEE.
Bluetooth.org (2015). "Core Specification | Bluetooth Development Portal." **2015**.
Bonanno, G. (1990). "GENERAL EQUILIBRIUM THEORY WITH IMPERFECT COMPETITION1." Journal of economic surveys **4**(4): 297-328.
Borg, W. R. and M. D. Gall (1989). Educational Research: An Introduction, Longman.
Börgers, T. (1991). "Upper hemicontinuity of the correspondence of subgame-perfect equilibrium outcomes." Journal of Mathematical Economics **20**(1): 89-106.
Borwein, J. M. and A. S. Lewis (2000). Convex Analysis and Nonlinear Optimization: Theory and Examples, Springer Verlag.
Boyd, S., L. E. Ghaoul, E. Feron and V. Balakrishnan (1994). Linear Matrix Inequalities in System and Control Theory, Society for Industrial and Applied Mathematics.
Boyd, S. P. and L. Vandenberghe (2004). Convex Optimization, Cambridge University Press.
Braden, R. (1989). "Requirements for Internet Hosts - Communication Layers."
Braess, D. (1965). "Einfluß der Wechselwirkung im Endzustand auf die Elektrospaltung des Deuterons in der Nähe der Schwelle." Zeitschrift für Physik **184**(3): 241-270.
Brewer, E. A. (2000). Towards robust distributed systems. PODC.
Brewer, E. A. (2000). Towards robust distributed systems. Principles of Distributed Computing.

Buchegger, S. and J.-Y. Le Boudec (2002). Performance analysis of the CONFIDANT protocol. Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, ACM.
Burguet, R., R. Caminal and M. Ellman (2015). "In Google we trust?" International Journal of Industrial Organization **39**: 44-55.
Burroughs, W. S. (1991). "The War Universe." Grand Street **1**(37).
Capra, L. (2004). "Towards a human trust model for mobile ad-hoc networks."
Carathéodory, C., N. Hadjisavvas and P. M. Pardalos (2001). Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873-1950), Springer.
Carson, E. and L. Shabel (2015). "Kant: Studies on Mathematics in the Critical Philosophy."
Carson, E. and L. Shabel (2015). "Mathematics in Kant's Critical Philosophy."
Chadwick, J. N. and D. S. Bindel (2015). "An Efficient Solver for Sparse Linear Systems Based on Rank-Structured Cholesky Factorization." arXiv preprint arXiv:1507.05593.
Chae, J. S. U. and J. Hedman (2015). "Business Models for NFC based mobile payments." Journal of Business Models **3**(1).
Chakraborti, S., D. Acharjya and S. Sanyal (2015). "Application Security framework for Mobile App Development in Enterprise setup." arXiv preprint arXiv:1503.05992.
Chapman, C. (2012). Communications Report 2011-12, Australian Communications and Media Authority (ACMA).
Chen, L. (2015). A Model of Mobile Work Continuance of Knowledge Workers: Evidences from China. The Third International Conference on E-Technologies and Business on the Web (EBW2015).
Chen, L., S. H. Low and J. C. Doyle (2010). "Random access game and medium access control design." IEEE/ACM Transactions on Networking (TON) **18**(4): 1303-1316.
Chen, X. and X. Deng (2006). Settling the Complexity of Two-Player Nash Equilibrium. FOCS.
Cho, J.-H. and A. Swami (2009). Towards trust-based cognitive networks: A survey of trust management for mobile ad hoc networks, DTIC Document.
Cho, J.-H., A. Swami and I.-R. Chen (2012). "Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks." Journal of Network and Computer Applications **35**(3): 1001-1012.
Chou, C.-H., P. Liu, T. Wu, Y. Chien and Y. Zhao (2014). Implementation of Parallel Computing FAST Algorithm on Mobile GPU. Unifying Electrical Engineering and Electronics Engineering, Springer**:** 1275-1281.
Cindy McArthur : Hq, D. M. M. (2009). "NASA - Do-It-Yourself Podcast: Rocket Evolution." **2015**.
Clark, J., A. Leavitt and D. Williams (2015). "Online Games, Community Aspects of." The International Encyclopedia of Digital Communication and Society.
Cocosila, M. and H. Trabelsi (2015). A user perspective on contrasting factors of contactless mobile payments adoption. Proceedings of International Academic Conferences, International Institute of Social and Economic Sciences.
Comer, D. (2006). Internetworking with TCP/IP: Principles, protocols, and architecture, Pearson Prentice Hall.
Constantin P. Niculescu, L.-E. P. (2004). "CONVEX FUNCTIONS AND THEIR APPLICATIONS A contemporary approach."
Conti, M. and S. Giordano (2014). "Mobile ad hoc networking: milestones, challenges, and new research directions." Communications Magazine, IEEE **52**(1): 85-96.
Costello, S. (2015). "Huawei predicts 100B connected terminals by 2025." **2015**.
Costello, S. (2015). "ZTE touts 100B connections in 2020; talks 'Smart 2.0'." **2015**.
Cournot, A. A. (1838). Recherches sur les principes mathématiques de la théorie des richesses, L. Hachette.
Cox, D. A. (2004). The arithmetic-geometric mean of Gauss. Pi: A Source Book, Springer**:** 481-536.
Creswell, J. W. (2003). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, SAGE Publications.

Cui, P., J. Zhou and J. Wang (2014). "Stackelberg games-based distributed algorithm of transmission rate match in end-to-end communication." Future Information Engineering (2 Volume Set) **49**: 323.
Cui, S. (2013). "Lecture 2: Convex functions." ECEN 629 Retrieved 27/4/2013, 2013, from http://ece.tamu.edu/~cui/ECEN629/lecture2.pdf.
Dantzig, G. B. (2014). "The Nature of Mathematical Programming." Retrieved 4/5/2013, 2013, from http://glossary.computing.society.informs.org/index.php?page=nature.html.
Daskalakis, C., P. W. Goldberg and C. H. Papadimitriou (2009). "The complexity of computing a Nash equilibrium." SIAM Journal on Computing **39**(1): 195-259.
Day, J. D. and H. Zimmermann (1983). "The OSI reference model." Proceedings of the IEEE **71**(12): 1334-1340.
Dean, J. and S. Ghemawat (2004). "MapReduce: Simplied Data Processing on Large Clusters." Google, Inc.
Debreu, G. (1959). Theory of Value: An Axiomatic Analysis of Economic Equilibrium, Yale University Press.
Demirkol, I., C. Ersoy and F. Alagoz (2006). "MAC protocols for wireless sensor networks: a survey." Communications Magazine, IEEE **44**(4): 115-121.
Derrida, J. (1986). "There is No "One" Narcissism." Le bon plaisir de Jacques Derrida: 200.
Dickhaut, J. and T. Kaplan (1991). "A program for finding Nash equilibria." The Mathematica Journal **1**(4): 87-93.
Diwekar, U. (2003). Introduction to applied optimization, Distributors for North, Central and South America, Kluwer Academic Publishers.
Dixit, A. and S. Skeath (1999). Games of Strategy, WW Norton & Company, New York.
Donaldson, S. E., S. G. Siegel, C. K. Williams and A. Aslam (2015). Enterprise Cybersecurity for Mobile and BYOD. Enterprise Cybersecurity, Springer**:** 119-129.
Donoghue, W. F. (1969). Distributions and Fourier transforms, Elsevier Science.
Dougherty, C. (2015). "Gambling Website's Bitcoin-Denominated Stock Draws SEC Inquiry."
Doup, T. (1988). Simplicial algorithms on the simplotope, Springer-Verlag.
Dye, M., R. McDonald and A. Rufi (2007). Network Fundamentals, CCNA Exploration Companion Guide, Cisco press.
E. Kohlberg, J. M. (1986). "On the strategic stability of equilibria." Econometrica: 1003–1037.
Eatwell, J., M. Milgate and P. K. Newman (1989). Game Theory, W W Norton & Company Incorporated.
Eha, B. P. (2013). "An Accelerated History of Internet Speed (Infographic)." **2015**.
Eissa, T., S. A. Razak, R. H. Khokhar and N. Samian (2013). "Trust-Based Routing Mechanism in MANET: Design and Implementation." Mob. Netw. Appl. **18**(5): 666.
Eissa, T., S. A. Razak, R. H. Khokhar and N. Samian (2013). "Trust-Based Routing Mechanism in MANET: Design and Implementation." Mob. Netw. Appl. **18**(5): 666-677.
ElDelgawy, R. and R. J. La (2015). A case study of internet fast lane. Information Sciences and Systems (CISS), 2015 49th Annual Conference on, IEEE.
Ephraim, Y. and B. L. Mark (2015). "Causal recursive parameter estimation for discrete-time hidden bivariate Markov chains." Signal Processing, IEEE Transactions on **63**(8): 2108-2117.
Ericsson (2015). Ericsson Mobility report.
Eriksen, E. (2010). "Principal Minors and the Hessian." Retrieved 7/5/2013, 2013, from http://home.bi.no/a0710194/Teaching/BI-Mathematics/GRA-6035/2010/lecture5-hand.pdf.
Eschenauer, L., V. D. Gligor and J. Baras (2004). On trust establishment in mobile ad-hoc networks. Security Protocols, Springer.
Etesami, S. R. and T. Basar (2014). "Complexity of Equilibrium in Diffusion Games on Social Networks." arXiv preprint arXiv:1403.3881.
Eyal, I. and E. G. Sirer (2013). "Majority is not enough: Bitcoin mining is vulnerable." arXiv preprint arXiv:1311.0243.

Fehr, E. and U. Fischbacher (2003). "The nature of human altruism." Nature **425**(6960): 785-791.
Fenchel, W. (1949). "On conjugate convex functions." Canad. J. Math **1**: 73-77.
FinCen, U. S. D. o. t. T. (2013). "Statement of Jennifer Shasky Calvery, Director Financial Crimes Enforcement Network United States Department of the Treasury."
Firdhous, M., O. Ghazali and S. Hassan (2014). "Robust Multi-Dimensional Trust Computing Mechanism for Cloud Computing." Jurnal Teknologi **69**(2).
Fletcher, R. (1987). Practical methods of optimization, Wiley.
Floyd, S. and K. Fall (1999). "Promoting the use of end-to-end congestion control in the Internet." Networking, IEEE/ACM Transactions on **7**(4): 458-472.
Forouzan, B. A. and S. C. Fegan (2003). Data Communications and Networking, McGraw-Hill Higher Education.
Forum, N. (2015). "NFC Forum: Technical Specifications." **2015**.
Foucault, M. (2013). "What is an Author?".
Freedman, A. (2015). "MANAGING PERSONAL DEVICE USE IN THE WORKPLACE: HOW TO AVOID DATA SECURITY ISSUES AND TO DIG YOURSELF OUT OF YOUR FAILED BYOD POLICY." Suffolk J. Trial & App. Adv. **20**: 284-361.
French, A., C. Guo, M. Schmidt and J. Shim (2015). "An Exploratory Study on BYOD in Class: Opportunities and Concerns."
French, S. (1986). Decision theory: an introduction to the mathematics of rationality, Halsted Press.
Gairing, M., T. Lücking, M. Mavronicolas, B. Monien and M. Rode (2004). Nash equilibria in discrete routing games with convex latency functions. Automata, Languages and Programming, Springer**:** 645-657.
Gigerenzer, G. and R. Selten (2002). Bounded Rationality: The Adaptive Toolbox, MIT Press.
Gilbert, H. and H. Handschuh (2004). Security analysis of SHA-256 and sisters. Selected areas in cryptography, Springer.
Golbeck, J. (2006). Computing with trust: Definition, properties, and algorithms. Securecomm and Workshops, 2006, IEEE.
Golbeck, J. and J. Hendler (2004). Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. Engineering knowledge in the age of the semantic web, Springer**:** 116-131.
Gong, W., Z. You, D. Chen, X. Zhao, M. Gu and K.-Y. Lam (2010). "Trust Based Routing for Misbehavior Detection in Ad Hoc Networks." Journal of Networks **5**(5).
Goode, E. (2015, 20150524). "John F. Nash Jr., Math Genius Defined by a 'Beautiful Mind,' Dies at 86." Retrieved 5/6/2015, 2015, from http://www.nytimes.com/2015/05/25/science/john-nash-a-beautiful-mind-subject-and-nobel-winner-dies-at-86.html.
Google (2014). "Google Developers." **2014**.
Google (2015). "Android Wear - Android Powered Wearables - Google Store." **2015**.
Google (2015). "Google Glass." **2015**.
Goralski, W. (2009). The illustrated network: how TCP/IP works in a modern network, Morgan Kaufmann.
Govindan, S. and R. Wilson (2003). "A global Newton method to compute Nash equilibria." Journal of Economic Theory **110**(1): 65-86.
Gowthami, V. and R. Buvaneswari (2013). "An Efficient Attribute Based Schema for Trust and Cluster Based Authentication Mechanism in MANET." International Journal **2**(6): 77.
Gowthami, V. and R. Buvaneswari (2013). "An Efficient Attribute Based Schema for Trust and Cluster Based Authentication Mechanism in MANET." International Journal **2**(6).
Gunasekaran, M. and K. Premalatha (2013). "TEAP: trust-enhanced anonymous on-demand routing protocol for mobile ad hoc networks." IET Information Security **7**(3): 203-211.
Gupta, S. K. (1995). Numerical Methods For Engineers, New Age International (P) Limited.
Haas, P. J. (2002). Stochastic petri nets, Springer.

Han, Z. and K. R. Liu (2008). Resource allocation for wireless networks: basics, techniques, and applications, Cambridge university press.
Han, Z., D. Niyato, W. Saad, T. Başar and A. Hjorungnes (2012). Game theory in wireless and communication networks, Cambridge University Press.
Harsanyi, J. C. (1963). "A simplified bargaining model for the n-person cooperative game." International Economic Review **4**(2): 194-220.
Harsanyi, J. C. (1966). "A general theory of rational behavior in game situations." Econometrica: Journal of the Econometric Society: 613-634.
Harsanyi, J. C. (1967). "Games with Incomplete Information Played by "Bayesian" Players, I-III Part I. The Basic Model." Management science **14**(3): 159-182.
Harsanyi, J. C. (2004). "Games with Incomplete Information Played by "Bayesian" Players, I–III: Part I. The Basic Model&." Management science **50**(12 supplement): 1804-1817.
Harsanyi, J. C. and R. Selten (1988). "A general theory of equilibrium selection in games." MIT Press Books **1**.
Harter, G., R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner and J. Vandaele (2015). Demo: FIT IoT-LABA: Large Scale Open Experimental IoT Testbed. Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, ACM.
He, J., S. Ma and B. Zhao (2013). "Analysis of Trust-based Access Control Using Game Theory." International Journal of Multimedia & Ubiquitous Engineering **8**(4).
He, T., C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher (2003). Range-free localization schemes for large scale sensor networks. Proceedings of the 9th annual international conference on Mobile computing and networking, ACM.
Hitchens, C. (2003). "Mommie Dearest." The Skeptic's Skeptic: 86.
Holman Rector, L. (2008). "Comparison of Wikipedia and other encyclopedias for accuracy, breadth, and depth in historical articles." Reference services review **36**(1): 7-22.
Horn, R. A. and C. R. Johnson (1990). Matrix Analysis, Cambridge University Press.
Huang, K. (1963). Statistical mechanics, Wiley.
Huang, K. L., S. S. Kanhere and W. Hu (2014). "On the need for a reputation system in mobile phone based sensing." Ad Hoc Networks **12**: 130-149.
Huth, M. and J. H.-P. Kuo (2014). PEALT: An Automated Reasoning Tool for Numerical Aggregation of Trust Evidence. Proc. of 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2014), Lecture Notes in Computer Science (ARCoSS). Springer.
Imbachi, J. L. C., D. L. N. Jacome and G. Ramirez (2015). "Mobile payments system employing NFC technology under the Android operating system." Sistemas y Telemática **13**(33): 77-87.
Intel. (2013, 2013). "Notebooks, Desktops, Ultrabook, Tablets, Servers - Intel Australia." from http://www.intel.com.
Itpro (2015). "Surge in BYOD sees 7/10 employees using their own devices." **2015**.
Ivos, A. (2013). "Matlab: JOR and SOR method." Retrieved 9/6/2013, 2013, from http://ivosabroad.wordpress.com/2013/01/03/matlab-jor-and-sor-method/.
J. A. Bondy and U. S. R. Murty (2008). Graph Theory, New York : Springer.
Jiang, T. and J. S. Baras (2004). Ant-based adaptive trust evidence distribution in MANET. Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on, IEEE.
John R. Gilbert, C. M., Robert Schreiber. (1992). "Sparse Matrices in Matlab: Design and Implementation." from http://www.nada.kth.se/kurser/kth/2D1252/gilbert92sparse.pdf.
John von Neumann, O. M. (2013). "Theory of Games and Economic - Books - Debate.org."
Kakutani, S. (1941). "A generalization of Brouwer's fixed point theorem." Duke mathematical journal **8**(3): 457-459.
Kalai, E. and M. Smorodinsky (1975). "Other solutions to Nash's bargaining problem." Econometrica: Journal of the Econometric Society: 513-518.

Kamvar, S., M. Schlosser and H. Garcia-Molina (2003). EigenRep: Reputation management in peer-to-peer networks. Proceedings of 12th International World Wide Web Conference, Budapest, Hungary.
Kanoc, T. (1999). "Mobile MIddleware: The Next Frontier in Enterprise Application Integration."
Karame, G., E. Androulaki and S. Capkun (2012). "Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin." IACR Cryptology ePrint Archive **2012**: 248.
Kelly, F. (2003). "Fairness and Stability of End-to-End Congestion Control." European Journal of Control **9**(2): 159-176.
Keppel, G. (1991). Design and Analysis: A Researcher's Handbook, Prentice Hall.
Kingman, J. F. C. (1961). "A Convexity Property of Positive Matrices." The Quarterly Journal of Mathematics **12**(1): 283-284.
Klir, G. J. and B. Yuan (1995). Fuzzy sets and fuzzy logic, Prentice Hall New Jersey.
Knackmuß, J. and R. Creutzburg (2015). Enterprise Mobility Management (EMM)-a way to increase the security of mobile devices. IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics.
Koller, D. and B. Milch (2003). "Multi-agent influence diagrams for representing and solving games." Games and Economic Behavior **45**(1): 181-221.
Korilis, Y. A., A. A. Lazar and A. Orda (1999). "Avoiding the Braess paradox in non-cooperative networks." Journal of Applied Probability **36**(1): 211-222.
Kovacs, E., K. Robrie and M. Reich (2006). "Integrating Mobile Agents into Mobile Middleware." Mobile Agents **1477/1998**: 124-135.
Kozierok, C. M. (2005). The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference, No Starch Press.
Krasnosel'skiĭ, M. A. and I. A. B. Rutit͡skiĭ (1961). Convex functions and Orlicz spaces, P. Noordhoff.
Kreps, D. M., P. Milgrom, J. Roberts and R. Wilson (1982). "Rational cooperation in the finitely repeated prisoners' dilemma." Journal of Economic theory **27**(2): 245-252.
Kuhn, H. W. (1997). Classics in Game Theory, Princeton University Press.
Kurose, J. F. and K. W. Ross (2007). Computer networking: a top-down approach, Addison-Wesley.
Lai, C.-C. and C.-F. Wu (2014). "Display and device size effects on the usability of mini-notebooks (netbooks)/ultraportables as small form-factor Mobile PCs." Applied ergonomics.
Lee, S., R. Sherwood and B. Bhattacharjee (2003). Cooperative peer groups in NICE. INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, IEEE.
Lemke, C. E. (1965). "Bimatrix equilibrium points and mathematical programming." Management science **11**(7): 681-689.
Lemke, C. E. and J. Howson, Joseph T (1964). "Equilibrium points of bimatrix games." Journal of the Society for Industrial & Applied Mathematics **12**(2): 413-423.
Lenth, R. V. (2001). "Some Practical Guidelines for Effective Sample Size Determination." The American Statistician **55**(3).
Lewis, D. (2008). Convention: A philosophical study, Wiley. com.
Lincoln, Y. S., Guba, E., G. (2000). Handbook of Qualitative Research, Thousand Oaks, CA: Sage Publications, Inc.
Lindfield, G. and J. Penny (2012). Numerical Methods: Using MATLAB, Elsevier Science.
logicalis (2012). "BYOD Research - Ovum Research Published. CXO Unplugged."
Loke, S. W. (2015). "The Internet of Flying-Things: Opportunities and Challenges with Airborne Fog Computing and Mobile Cloud in the Clouds." arXiv preprint arXiv:1507.04492.
Luce, R. D. and H. Raiffa (1957). Games and Decisions: Introduction and Critical Survey, Dover Publications.
Lucy, R. K. Q. P. B. (2014). "Do you trust the "right" amount?" **2014**: Do you trust the "right" amount?: Trust does matter.
Luenberger, D. G. (1997). Optimization by Vector Space Methods, Wiley.

Luenberger, D. G. (2003). Linear and Nonlinear Programming: Second Edition, Kluwer Academic.
Ma, R. T., D. M. Chiu, J. C. Lui, V. Misra and D. Rubenstein (2011). "On cooperative settlement between content, transit, and eyeball internet service providers." Networking, IEEE/ACM Transactions on **19**(3): 802-815.
MacKenzie, A. B. and L. A. DaSilva (2006). "Game theory for wireless engineers." Synthesis Lectures on Communications **1**(1): 1-86.
Malm, E.-J., M. Jani and J. Kela (2003). "Managing context information in mobile devices." IEEE pervasive computing **2**(3): 42-51.
Mangasarian, O. L. and H. Stone (1964). "Two-person nonzero-sum games and quadratic programming." Journal of Mathematical Analysis and Applications **9**(3): 348-355.
Marsan, M. A. (1990). Stochastic Petri nets: an elementary introduction. Advances in Petri Nets 1989, Springer**:** 1-29.
Marti, S., T. J. Giuli, K. Lai and M. Baker (2000). Mitigating routing misbehavior in mobile ad hoc networks. Proceedings of the 6th annual international conference on Mobile computing and networking, ACM.
Martins, S. and Y. Yang (2011). Introduction to bitcoins: a pseudo-anonymous electronic currency system. Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp.
Mathews, J. H. and K. D. Fink (2006). Numerical Methods Using MATLAB, Pearson Education, Limited.
MathWorks. (2012). "System Requirements - Release 2012a." Retrieved 18/6/2013, 2013, from http://www.mathworks.com.au/support/sysreq/release2012a/index.html.
Mathworks. (2013). "Functions - MATLAB & Simulink - MathWorks Australia." Retrieved 11/6/2013, 2013, from http://www.mathworks.com.au/help/matlab/functions.html.
MathWorks (2015). "Eigenvalues and eigenvectors - MATLAB eig - MathWorks Australia." **2015**.
MathWorks. (2015). "Matlab: Random Number Generation." Retrieved 15/6/2013, 2013, from http://www.mathworks.com.au/help/matlab/random-number-generation.html.
MathWorks. (2015). "One-way analysis of variance - MATLAB anova1 - MathWorks Australia." Retrieved 19/6/2013, 2013, from http://www.mathworks.com.au/help/stats/anova1.html.
Mayer, R. C., J. H. Davis and F. D. Schoorman (1995). "An integrative model of organizational trust." Academy of management review **20**(3): 709-734.
McCabe, K., M. Rigdon and V. Smith (2002). Cooperation in single play, two-person extensive form games between anonymously matched players. Experimental Business Research, Springer**:** 49-67.
McClure, C. R. and P. Hernon (1991). Library and Information Science Research: Perspectives and Strategies for Improvement, Ablex Publishing Corporation.
McDonald, I. R. (1985). Statistical Mechanics, Allied.
McKelvey, R. D. and A. McLennan (1996). "Computation of equilibria in finite games." Handbook of computational economics **1**: 87-142.
McKelvey, R. D., A. M. McLennan and T. L. Turocy (1995). "Gambit: Software Tools for Game Theory, Version 0.97. 0.7 (2004)." URL http://econweb. tamu. edu/gambit.
McKenzie, L. W. (1959). "On the existence of general equilibrium for a competitive market." Econometrica: journal of the Econometric Society: 54-71.
McQuarrie, D. A. (2000). Statistical Mechanics, University Science Books.
Mejia, M., N. Peña, J. L. Muñoz, O. Esparza and M. A. Alzate (2011). "A game theoretic trust model for on-line distributed evolution of cooperation inMANETs." Journal of Network and Computer Applications **34**(1): 39-51.
Menaka, R. and V. Ranganathan (2013). "A Survey of Trust related Routing Protocols for Mobile Ad Hoc Networks."

Michalopoulou, M. and P. Mahonen (2012). Game theory for wireless networking: Is a Nash equilibrium always a desirable solution? Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on.
Microsoft. (2013). "Windows 7 system requirements - Microsoft Windows." Retrieved 18/6/2013, 2013, from http://windows.microsoft.com/en-AU/windows7/products/system-requirements.
Miller, H. R. (1999). Optimization: Foundations and Applications, Wiley.
Mitchell, I. M., A. M. Bayen and C. J. Tomlin (2005). "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games." Automatic Control, IEEE Transactions on **50**(7): 947-957.
Montgomery, D. C. (2012). Design and Analysis of Experiments, 8th Edition, John Wiley & Sons, Incorporated.
Morrison, C. C. (1998). "Cournot, bertrand, and modern game theory." Atlantic Economic Journal **26**(2): 172-174.
Murthy, C. S. R. and B. Manoj (2004). Ad hoc wireless networks: Architectures and protocols, Pearson education.
Myerson, R. B. (1978). "Refinements of the Nash equilibrium concept." International journal of game theory **7**(2): 73-80.
Myerson, R. B. (1997). Game Theory: Analysis of Conflict, Harvard University Press.
Myerson, R. B. (2013). Game theory: analysis of conflict, Harvard university press.
Nakamoto, S. (2008). "Bitcoin: A peer-to-peer electronic cash system." Consulted **1**(2012): 28.
Nash, J. (1951). "Non-cooperative games." Annals of mathematics: 286-295.
Nash, J. (1951). "Non-cooperative games." The Annals of Mathematics **54**(2): 286-295.
Nash, J. (1953). "Two-person cooperative games." Econometrica: Journal of the Econometric Society: 128-140.
Nash, J. F. (1950). "Equilibrium points in n-person games." Proceedings of the national academy of sciences **36**(1): 48-49.
Nash, J. F. (1996). Essays on game theory, Edward Elgar Publishing.
Nash Jr, J. F. (1950). "The bargaining problem." Econometrica: Journal of the Econometric Society: 155-162.
Nesterov, Y. and I. U. E. Nesterov (2004). Introductory Lectures on Convex Optimization: A Basic Course, Springer.
Neumann, J. v. (1928). "Zur theorie der gesellschaftsspiele." Mathematische Annalen **100**(1): 295-320.
Nguyen, V. H., J. J. Strodiot and P. Tossings (2000). Optimization: Proceedings of the 9th Belgian-French-German Conference on Optimization, Namur, September 7-11, 1998, Springer Verlag.
Nicholson, W. and C. M. Snyder (2011). Microeconomic Theory: Basic Principles and Extensions [With Access Code], CengageBrain. com.
Nie, N. and C. Comaniciu (2005). Adaptive channel allocation spectrum etiquette for cognitive radio networks. New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on.
Niyato, D. and E. Hossain (2008). "Competitive Pricing for Spectrum Sharing in Cognitive Radio Networks: Dynamic Game, Inefficiency of Nash Equilibrium, and Collusion." Selected Areas in Communications, IEEE Journal on **26**(1): 192-202.
Nocedal, J. A. and S. J. Wright (1999). Numerical optimization, Springer-Verlag New York.
Noor, T., Q. Sheng, L. Yao, S. Dustdar and A. Ngu (2015). "CloudArmor: Supporting Reputation-based Trust Management for Cloud Services."
Nudelman, E., J. Wortman, Y. Shoham and K. Leyton-Brown (2004). Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2, IEEE Computer Society.

O'Sullivan, T. C. (1971, April, 1971). "User/Server Site Protocol: Network Host Questionnaire." from https://tools.ietf.org/html/rfc112.
Olver, P. J. (2008). "Numerical Analysis Lecture Notes." Retrieved 3/5/2013, 2013, from http://www.math.umn.edu/~olver/num_/lnv.pdf.
Oosterbeek, H., R. Sloof and G. Van De Kuilen (2004). "Cultural differences in ultimatum game experiments: Evidence from a meta-analysis." Experimental Economics **7**(2): 171-188.
OpenGroup (2015). "TOGAF Core Concepts." **2015**.
Ortega, J. M. (1990). Numerical Analysis: A Second Course, Society for Industrial and Applied Mathematics.
Osborne, M. J. and A. Rubinstein (1994). A Course in Game Theory, MIT Press.
Page, T. (2015). "A Forecast of the Adoption of Wearable Technology." International Journal of Technology Diffusion (IJTD) **6**(2): 12-29.
Pakazad, S. K., A. Hansson, M. S. Andersen and A. Rantzer (2015). "Distributed semidefinite programming with application to large-scale system analysis." arXiv preprint arXiv:1504.07755.
Pakes, A., M. Ostrovsky and S. Berry (2007). "Simple estimators for the parameters of discrete dynamic games (with entry/exit examples)." The RAND Journal of Economics **38**(2): 373-399.
Palumbo, F., P. Barsocchi, S. Chessa and J. C. Augusto (2015). A stigmergic approach to indoor localization using Bluetooth Low Energy beacons. Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, IEEE.
Pande, S. and N. Gomes (2015). "Leveraging mobile devices for human resource information systems." International Journal of Business Information Systems **20**(1): 23-40.
Papadimitriou, C. H. (2003). Computational complexity, John Wiley and Sons Ltd.
Parnell, C. (2013). "Chapter 2: Iterative Methods." Retrieved 31/5/2013, 2013, from http://www-solar.mcs.st-andrews.ac.uk/~clare/Lectures/num-analysis/Numan_chap2.pdf.
Perkins, C. E. and P. Bhagwat (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review, ACM.
Perkins, C. E. and P. Bhagwat (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review, ACM.
Peters, B. G. and J. Pierre (2004). "Multi-level governance and democracy: a Faustian bargain?" Multi-level governance: 75-89.
Pham, T.-T. T. and J. C. Ho (2015). "The effects of product-related, personal-related factors and attractiveness of alternatives on consumer adoption of NFC-based mobile payments." Technology in Society.
Pigou, A. C. (1905). Principles & methods of industrial peace, Macmillan.
Pigou, A. C. (1924). The economics of welfare, Transaction Publishers.
Polak, E. (1997). Optimization: Algorithms and Consistent Approximation, Springer-Verlag.
POPPER, K. R. A. (2002). Karl Popper: The Logic of Scientific Discovery, Routledge Classics.
Porter, R., E. Nudelman and Y. Shoham (2008). "Simple search methods for finding a Nash equilibrium." Games and Economic Behavior **63**(2): 642-662.
Poundstone, W. (1992). Prisoner's dilemma, Doubleday.
Poundstone, W. and N. Metropolis (1992). "Prisoner's dilemma: John von Neumann, game theory, and the puzzle of the bomb." Physics Today **45**: 73.
Qian, H. and D. Andresen (2015). Extending mobile device's battery life by offloading computation to cloud. Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems, IEEE Press.
Rabin, M. (2000). "Risk aversion and expected-utility theory: A calibration theorem." Econometrica **68**(5): 1281-1292.
Rahaman, M. and M. M. Islam (2015). "A Review on Progress and Problems of Quantum Computing as a Service (QCaaS) in the Perspective of Cloud Computing." Global Journal of Computer Science and Technology **15**(4).

Rahman, M. M., Maksud-Ul-Alam and S. Monzurur Rahman (2015). "An open multi-tier architecture for high-performance data mining using SOA." International Journal of Data Mining, Modelling and Management **7**(1): 60-82.
Raiffa, H. (1982). The art and science of negotiation, Harvard University Press.
Raja, S. (2015). "THE NOT SO BROAD-BAND: PUBLIC POLICY ARGUMENT ABOUT BROADBAND LEGISLATION IN NORTH CAROLINA AND TENNESSEE AND THE POTENTIAL NATIONAL IMPACT." NCJL & Tech. On. **16**: 106-301.
Rajesh, A. and N. M. Kumar (2014). "Multi-Level Trust Architecture for Mobile Adhoc Networks Based on Context Aware." Journal of Theoretical and Applied Information Technology **59**(2).
Rapoport, A. (1965). Prisoner's dilemma: A study in conflict and cooperation, University of Michigan Press.
Rath, K. P. (1996). "Existence and upper hemicontinuity of equilibrium distributions of anonymous games with discontinuous payoffs." Journal of Mathematical Economics **26**(3): 305-324.
Ray, I. and S. Chakraborty (2004). A vector model of trust for developing trustworthy systems. Computer Security–ESORICS 2004, Springer**:** 260-275.
Reich, E. (1949). The Convergence of the Gauss-Seidel Iterative Method. Massachusetts Institute of Technology. Servomechanisms Laboratory: Project Whirlwind, Servomechanisms Laboratory, Massachusetts Institute of Technology.
Resnick, P., R. Zeckhauser, J. Swanson and K. Lockwood (2006). "The value of reputation on eBay: A controlled experiment." Experimental Economics **9**(2): 79-101.
Richardson, M., R. Agrawal and P. Domingos (2003). Trust management for the semantic web. The Semantic Web-ISWC 2003, Springer**:** 351-368.
Rockafellar, R. T. (1997). Convex analysis, PRINCETON University Press.
Rodriguez, J. M., C. Mateos and A. Zunino (2014). "Energy-efficient job stealing for CPU-intensive processing in mobile devices." Computing **96**(2): 87-117.
Rogaway, P. (2004). Nonce-based symmetric encryption. Fast Software Encryption, Springer.
Rosen, J. B. (1965). "Existence and uniqueness of equilibrium points for concave n-person games." Econometrica: Journal of the Econometric Society: 520-534.
Saad, Y. (2003). Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics.
Saaty, T. L. (1980). "The analytic hierarchy process: planning, priority setting, resources allocation." M cGraw-Hill.
Saaty, T. L. (1988). What is the analytic hierarchy process?, Springer.
Saha, D. and A. Mukherjee (2003). "Pervasive computing: a paradigm for the 21st century." Computer **36**(3): 25-31.
Sahinidis, N. V. (2002). Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Springer.
Salus, P. H. and G. Vinton (1995). Casting the Net: From ARPANET to Internet and Beyond, Addison-Wesley Longman Publishing Co., Inc.
Sanchez, C. A. (2013). A Risk and Trust Security Framework for the Pervasive Mobile Environment, Citeseer.
Schneider, H. (1977). "Olga Taussky-Todd's influence on matrix theory and matrix theorists." Linear and Multilinear Algebra **5**(3): 197-224.
Schuett, C., J. Butts and S. Dunlap (2014). "An evaluation of modification attacks on programmable logic controllers." International Journal of Critical Infrastructure Protection **7**(1): 61-68.
Schumpeter, J. A. and A. Nichol (1934). "Robinson's economics of imperfect competition." The Journal of Political Economy **42**(2): 249-259.
Schwalbe, U. and P. Walker (2001). "Zermelo and the early history of game theory." Games and Economic Behavior **34**(1): 123-137.

Selten, R. (1965). "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit: Teil I: Bestimmung des dynamischen Preisgleichgewichts." Zeitschrift für die gesamte Staatswissenschaft/Journal of Institutional and Theoretical Economics **121**(2): 301-324.

Sen, J. (2010). A distributed trust and reputation framework for mobile ad hoc networks. Recent Trends in Network Security and Applications, Springer**:** 538-547.

Sen, J., P. R. Chowdhury and I. Sengupta (2007). A distributed trust establishment scheme for mobile ad hoc networks. Computing: Theory and Applications, 2007. ICCTA'07. International Conference on, IEEE.

Sethumadhavan, S., A. Waksman, M. Suozzo, Y. Huang and J. Eum (2015). "Trustworthy hardware from untrusted components." Communications of the ACM **58**(9): 60-71.

Shadish, W. R., T. D. Cook and D. T. Campbell (2002). Experimental and Quasi-experimental Designs for Generalized Causal Inference, Houghton Mifflin.

Shah, I., S. Jan and K.-K. Loo (2010). Selfish Flow Games in Non-Cooperative Multi-Radio Multi-Channel Wireless Mesh Networks With Imperfect Information. Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on, IEEE.

Shah, I., S. Jan, K.-K. Loo and C. E.-A. Campbell (2011). "Selfish Flow Games in Non-Cooperative Multi-Radio Multi-Channel Wireless Mesh Networks With Interference Constraint Topology." International Journal On Advances in Telecommunications **4**(1 and 2): 172-182.

Shah, I. A., S. Jan, I. Khan and S. Qamar (2012). "An Overview of Game Theory and its Applications in Communication Networks." Interational Journal of Multidisciplinary Sciences and Engineering **3**(4).

Shah, I. A., S. Jan, I. Khan and S. Qamar (2012). "An Overview of Game Theory and its Applications in Communication Networks." Interational Journal of Multidisciplinary Sciences and Engineering **3**(4): 10.

Shastri, M. V., G. Patil and M. U. Karande (2013). "Cloud Computing Used In Mobile Network." IJRCCT **2**(4): 175-179.

Sher, A. (2015). Simulation of Attacks in a Wireless Sensor Network using NS2, Texas A&M University-Corpus Christi.

Shiraz, M., E. Ahmed, A. Gani and Q. Han (2014). "Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing." The Journal of Supercomputing **67**(1): 84-103.

Smith, J. M. (1993). Evolution and the Theory of Games, Springer.

Smith, K. L. (2015). "PEAF - Framework." **2015**.

Smith, S. D. (2015). "Kant's Mathematical Sublime and the Role of the Infinite: Reply to Crowther." Kantian Review **20**(01): 99-120.

Snowden, D. J. and M. E. Boone (2015). "A Leader's Framework for Decision Making." Harvard Business Review.

Sorensen, C. (2011). Enterprise mobility: tiny technology with global impact on work, Palgrave Macmillan.

Srivastava, R. K. and J. Ashok (2005). STATISTICAL MECHANICS, PHI Learning.

Stallings, W. (2007). Data and Computer Communications, Pearson/Prentice Hall.

Stevens, W. R. and G. R. Wright (1995). TCP/IP Illustrated: Vol. 2: The Implementation, Addison-Wesley Professional.

Stojmenovic, I., S. Wen, X. Huang and H. Luan (2015). "An overview of Fog computing and its security issues." Concurrency and Computation: Practice and Experience.

Strang, G. (2003). Introduction to Linear Algebra, Wellesley-Cambridge Press.

Strum, R. D. and D. E. Kirk (1999). Contemporary linear systems using MATLAB, Brooks/Cole Publishing Co.

Subramanian, S. and B. Ramachandran (2012). "Trust Based Scheme for QoS Assurance in Mobile Ad-Hoc Networks." International Journal of Network Security & Its Applications **4**(1).

Sun, R., E. Ding, H. Jiang, R. Geng and W. Chen (2014). "Game Theoretic Approach in Adapting QoS Routing Protocol for Wireless Multimedia Sensor Networks." International Journal of Distributed Sensor Networks **2014**.
Sundaram, S. and J. S. C. Babu (2015). "Performance evaluation and validation of 5MW p grid connected solar photovoltaic plant in South India." Energy Conversion and Management **100**: 429-439.
Surhone, L. M., M. T. Timpledon and S. F. Marseken (2010). Prisoner's Dilemma: Game Theory, Merrill M. Flood, Melvin Dresher, Albert W. Tucker, Framing Device, Experimental Economics, Betascript Publishing.
Tam, D. (2013). "Facebook by the numbers: 1.06 billion monthly active users." CNET News: 8301-1023.
Tan, K.-K., J. Yu and X.-Z. Yuan (1995). "Existence theorems of Nash equilibria for non-cooperative n-person games." International Journal of Game Theory **24**(3): 217-222.
Tanenbaum, A. S. (2003). Computer Networks, Prentice Hall PTR.
Tatnall, A. D., B. (2015). "The Internet of Things and Beyond: Rise of the Non-Human Actors." International Journal of Actor-Network Theory and Technological Innovation **7(4)**(5/12/2015): 58-69.
Taussky, O. (1949). A recurring theorem on determinants.
Tavakolifard, M. and K. C. Almeroth (2012). A Taxonomy to Express Open Challenges in Trust and Reputation Systems.
TechNet. (2015). "Set Processor Affinity." Retrieved 09/08/2015, 2015, from https://technet.microsoft.com/en-us/library/cc759098(v=ws.10).aspx.
TechNet, M. (2015). "Typeperf." **2015**.
Thomson, B. S. (1994). Symmetric Properties of Real Functions, TBC, Marcel Dekker Incorporated.
Thooyamani, K., R. Udayakumar and V. Khanaa (2014). "Cooperative Trust Management Scheme for Wireless Sensor Networks." 254.
Thooyamani, K., R. Udayakumar and V. Khanaa (2014). "Cooperative Trust Management Scheme for Wireless Sensor Networks."
Thrall, R. M. and W. F. Lucas (1963). "N-person games in partition function form." Naval Research Logistics Quarterly **10**(1): 281-298.
Tim, B.-L., H. James and L. Ora. (2001). "Scientific American: The Semantic Web." Scientific America Retrieved 30/3/2014, 2014, from http://www.cs.umd.edu/~golbeck/LBSC690/SemanticWeb.html.
Trifunovic, S., M. Kurant, K. A. Hummel and F. Legendre (2014). "Preventing Spam in Opportunistic Networks." Computer Communications.
Tuckman, B. W. (1988). Conducting Educational Research, Harcourt Brace Jovanovich.
Tversky, A. (2004). Preference, belief, and similarity: selected writings, MIT Press.
U.S. Government Accountability Office. (2013). "VIRTUAL ECONOMIES AND CURRENCIES: Additional IRS Guidance Could Reduce Tax Compliance Risks." Retrieved 27/1/2015, from http://www.gao.gov/products/gao-13-516.
United States District Court (2013). Securities and Exchange Commission v Trenton T. Shavers and Bitcoin Savings and Trust, United States District Court.
Urpi, A., M. Bonuccelli and S. Giordano (2003). Modelling cooperation in mobile ad hoc networks: a formal description of selfishness. WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks.
Urruty, J. B. H. and C. Lemaréchal (2001). Fundamentals of convex analysis, Springer Verlag.
Van Benthem, J. (2001). "Logic in games." Lecture Notes, ILLC Amsterdam.
Van den Berg, A., I. Bos, P. Herings and H. Peters (2012). "Dynamic Cournot duopoly with intertemporal capacity constraints." International Journal of Industrial Organization **30**(2): 174-192.

Van der Laan, G., A. Talman and L. Van der Heyden (1987). "Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling." Mathematics of operations research **12**(3): 377-397.
Vandenberghe, B. a. (2013). "Convex Optimization." Retrieved 27/4/2013, 2013, from http://www.ee.ucla.edu/ee236b/lectures/functions.pdf.
Varga, R. S. (2000). Matrix Iterative Analysis, Springer.
Varian, H. R. (2010). Intermediate Microeconomics: A Modern Approach, W W Norton & Company Incorporated.
Venkataraman, R., P. M and T. Rama Rao (2012). Implementation of a Regression-based Trust Model in a Wireless Ad hoc Testbed.
Venkataraman, R., M. Pushpalatha and T. Rama Rao (2012). "Regression-based trust model for mobile ad hoc networks." Information Security, IET **6**(3): 131-140.
Von Stackelberg, H. (1932). "Grundlagen einer reinen Kostentheorie." Journal of Economics **3**(4): 552-590.
Von Stackelberg, H. (1934). Marktform und gleichgewicht, J. Springer.
Von Stengel, B. (2002). "Computing equilibria for two-person games." Handbook of game theory with economic applications **3**: 1723-1759.
Wald, A. (1942). "On the principles of statistical inference. Notre Dame Mathematical Lectures I." On the principles of statistical inference. Notre Dame Mathematical Lectures I.
Wald, A. (1950). "Statistical decision functions."
Walker, P. (2013). "Chronology of Game Theory."
Wang, A., T.-Y. Lin, S. Ouyang, W.-H. Huang, J. Wang, S.-H. Chang, S.-P. Chen, C.-H. Hu, J. C. Tai and K.-S. Tan (2014). 10.3 heterogeneous multi-processing quad-core CPU and dual-GPU design for optimal performance, power, and thermal tradeoffs in a 28nm mobile application processor. Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International, IEEE.
Wang, B., Y. Wu and K. J. R. Liu (2010). "Game theory for cognitive radio networks: An overview." Computer Networks **54**(14): 2537-2561.
Wang, D., H. Huang and C. Xie (2015). "A Novel Trust-Aware Composite Semantic Web Service Selection Approach." Mathematical Problems in Engineering **2015**.
Wang, K., M. Wu and S. Shen (2008). A trust evaluation method for node cooperation in mobile ad hoc networks. Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on, IEEE.
Weber, R. J. (1994). "Games in coalitional form." Handbook of Game Theory with Economic Applications **2**: 1285-1303.
Weeger, A., X. A. Wang, H. Gewald, O. P. Sanchez, M. Raisinghani, G. Grant and S. Pittayachawan (2015). Determinants of Intention to Participate in Corporate BYOD-Programs–The Case of Digital Natives–. Academy of Management Proceedings, Academy of Management.
Weimerskirch, A. and G. Thonet (2002). A distributed light-weight authentication model for ad-hoc networks. Information Security and Cryptology—ICISC 2001, Springer**:** 341-354.
Weintraub, E. R. (1992). Toward a history of game theory, Duke University Press.
William, H. T., P. T. Desmond, E. Z. Rodger, F. M. Nicholas and W. M. Jon (2007). OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection. The Best of the Best:Fifty Years of Communications and Networking Research, Wiley-IEEE Press**:** 599-606.
Williamson, K., A. Bow and C. S. U. C. f. I. Studies (2002). Research Methods for Students, Academics and Professionals: Information Management and Systems, Centre for Information Studies, Charles Sturt University.
Wright, S. J. (1997). Primal-Dual Interior-Point Methods, Society for Industrial and Applied Mathematics.
Xia, H., Z. Jia, L. Ju, X. Li and E. H.-M. Sha (2013). "Impact of trust model on on-demand multi-path routing in mobile ad hoc networks." Computer Communications **36**(9): 1078-1093.

Xia, H., Z. Jia, L. Ju, X. Li and Y. Zhu (2011). A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules. Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on, IEEE.
Xia, H., Z. Jia, L. Ju, X. Li and Y. Zhu (2011). A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules. Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on, IEEE.
Xia, H., Z. Jia, X. Li, L. Ju and E. H.-M. Sha (2013). "Trust prediction and trust-based source routing in mobile ad hoc networks." Ad Hoc Networks **11**(7): 2096-2114.
Xiong, L. and L. Liu (2002). Building trust in decentralized peer-to-peer electronic communities. Fifth International Conference on Electronic Commerce Research (ICECR-5).
Xiong, L. and L. Liu (2003). A reputation-based trust model for peer-to-peer e-commerce communities. E-Commerce, 2003. CEC 2003. IEEE International Conference on, IEEE.
Yan, Z., P. Zhang and A. V. Vasilakos "A Survey on Trust Management for Internet of Things." Journal of Network and Computer Applications(0).
Yang, K. (2014). Wireless Sensor Networks, Springer.
Yang, S., C. K. Yeo and B. S. Lee (2012). "Toward reliable data delivery for highly dynamic mobile ad hoc networks." Mobile Computing, IEEE Transactions on **11**(1): 111-124.
Yang, X. I. and R. Mittal (2014). "Acceleration of the Jacobi iterative method by factors exceeding 100 using scheduled relaxation." Journal of Computational Physics **274**: 695-708.
Yi, S., C. Li and Q. Li (2015). "A Survey of Fog Computing: Concepts, Applications and Issues."
Young, A. E. (2008). "Every Need to be Alarmed." International Journal of Web Portals **1**: 34-49.
Young, A. E. (2009). "Mobilising the Enterprise." International Journal of Web Portals **6**.
Young, A. E. (2009). "Service Oriented Architecture Conceptual Landscape PART I." International Journal of Web Portals **3**.
Young, A. E. (2009). "Service Oriented Architecture Conceptual Landscape PART II." International Journal of Web Portals **3**.
Young, A. E. (2009). "Service Oriented Architecture Conceptual Landscape PART III." International Journal of Web Portals **4**.
Young, A. E. (2009). "Service Oriented Architecture Conceptual Landscape PART IV." International Journal of Web Portals **5**.
Young, D. M. (2003). Iterative Solution of Large Linear Systems, DOVER PUBN Incorporated.
Young, E. (2012). Enhancing Enterprise and Service-oriented Architectures with Advanced Web Portal Technologies, IGI Global.
Young, E. and M. Jessopp (2010). "How Thick Is Your Client?" International Journal of Web Portals (IJWP) **2**(2): 1-11.
Young, E. and M. Jessopp (2012). "How Thick Is Your Client?" Enhancing Enterprise and Service-Oriented Architectures with Advanced Web Portal Technologies: 131.
Yu, B. and M. P. Singh (2002). An evidential model of distributed reputation management. Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, ACM.
Zermelo, E. (1913). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. Proceedings of the Fifth International Congress of Mathematicians, II, Cambridge UP, Cambridge.
Zervas, G., D. Proserpio and J. Byers (2015). "A First Look at Online Reputation on Airbnb, Where Every Stay is Above Average." Where Every Stay is Above Average (January 23, 2015).
Zeuthen, F. (1930). Problems of monopoly and economic warfare, G. Routledge & sons, ltd.
Zhang, Q., T. Yu and K. Irwin (2004). A Classification Scheme for Trust Functions in Reputation-Based Trust Management. ISWC Workshop on Trust, Security, and Reputation on the Semantic Web.
Ziegler, C.-N. and J. Golbeck (2015). Models for Trust Inference in Social Networks. Propagation Phenomena in Real World Networks, Springer**:** 53-89.

Ziegler, C.-N. and G. Lausen (2004). Spreading activation models for trust propagation. e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on, IEEE.
Zimmermann, H. (1980). "OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection." IEEE Transactions on Communications **28**(4): 425-432.

7 Appendices

7.1 Appendix: NPOST Simulation Matlab Script Listings

Simulation Environment Matlab scripts and listings are stored:

Script	Description	Location
diagdom.m	Matrix diagonal dominance test script.	https://www.dropbox.com/s/qtz9gq4r2f9tb7u/domdiag.m
NPOSTSimulation.m	Simulation JOR main script.	https://www.dropbox.com/s/n1ke0evcwwy1mha/NPOSTSimulation.m
genEnvFact.m	Environmental factors generation script.	https://www.dropbox.com/s/7tkjdrmx7qi1g8v/genEnvFact.m
NPOSTSimulationRun.m	Experiment configuration and execute simulation (NPOSTSimulation) script.	https://www.dropbox.com/s/xdmbao69k7fli66/NPOSTSimulationRun.m

All scripts were created in MathWorks Matlab R2012a (7.14.0.739) 64-bit Matlab (MathWorks 2012) and do not require any additional packages or scripts to run. The main simulation script uses the Windows utility Typeperf to monitor system resources the configuration for which, is included in the file “configtypeperf.conf” (TechNet 2015).

Permission needs to be granted for access to resources.

7.2 Appendix: NPOST Experimental Data and Figures

Simulation Environment Matlab experimental data and figures are stored:

Resource	Description	Location
Experiments.xlsx	Results and reference document for all experiments.	https://www.dropbox.com/s/rxr08dih1obeo56/Experiments.xlsx
NPOST	MySQL database repository for results.	http://npost.cugx20l8xnue.ap-southeast-2.rds.amazonaws.com:3066
/experiments	Contains experimental Matlab diary, workspace variables and figures. Single experiments are uniquely identifiable by their “stamp” reference value.	https://www.dropbox.com/sh/igx38lutifm2a8a/AABdch46wb0VsiitEP1XNGvha
configtypeperf.conf	Typeperf control configuration.	https://www.dropbox.com/s/mrwei3e85kh3mgx/configtypeperf.conf?dl=0

Permission needs to be granted for access to resources.

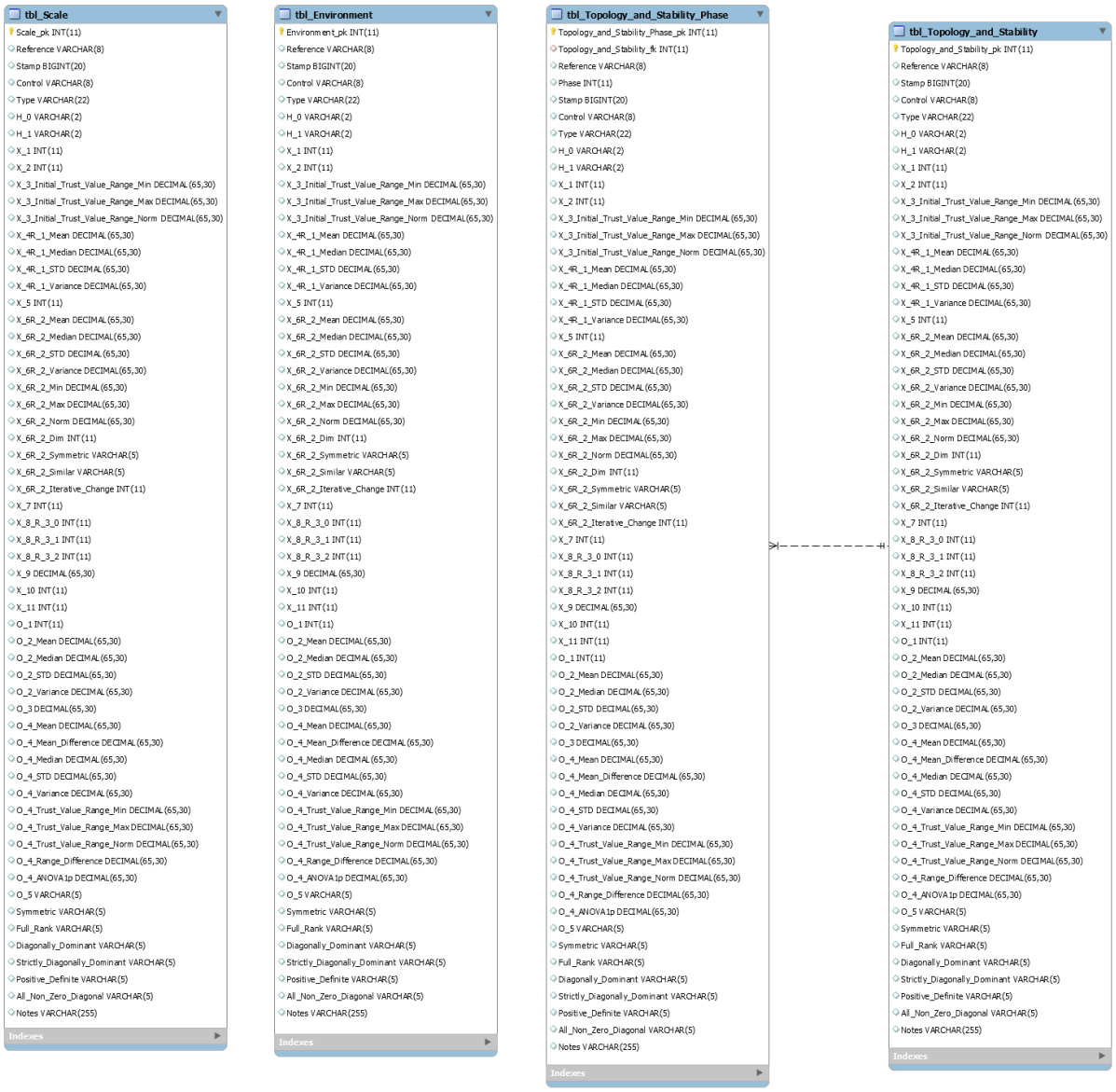
Directory / File	Description	Example
/experiments/<type>/<stamp>	Data directory.	/experiments/Scale/20150603171905
/experiments/<type>/<stamp>/<stamp>.txt	Matlab diary listing for experiment reference, “stamp”.	/experiments/Scale/20150603171905/20150603171905.txt
/experiments/<type>/<stamp>/<stamp>_initial.mat	Matlab pre-experiment workspace variables.	/experiments/Scale/20150603171905/20150603171905_initial.mat
/experiments/<type>/<stamp>/<stamp>_final.mat	Matlab most-experiment workspace variables.	/experiments/Scale/20150603171905/20150603171905_final.mat
/experiments/<type>/<stamp>/<stamp>_O_1_O_2_O_5.fig	“Convergence Norm” Matlab figure of observational variables O_1, O_2 and O_5.	/experiments/Scale/20150603171905/20150603171905_O_1_O_2_O_5.fig
/experiments/<type>/<stamp>/<stamp>_O_1_O_3_O_5.fig	“Nash Equilibrium Convergence” Matlab figure of observational variables O_1, O_3 and O_5.	/experiments/Scale/20150603171905/20150603171905_O_1_O_3_O_5.fig
/experiments/<type>/<stamp>/<stamp>_X_4R_1_O_4.fig	“Reputation Profile (Initial / Final)” Matlab figure of variables X_4R_1 and O_4.	/experiments/Scale/20150603171905/20150603171905_X_4R_1_O_4.fig
/experiments/<type>/<stamp>/<stamp>_TF.fig	“Trust Functions” Matlab figure of trust functions.	/experiments/Scale/20150603171905/20150603171905_TF.fig
/experiments/<type>/<stamp>/<stamp>_ANOVA1.fig	Matlab figure of ANOVA1 boxplot.	/experiments/Scale/20150603171905/20150603171905_ANOVA1.fig
/experiments/<type>/<stamp>/<stamp>_didnotrespond.fig	“Node Responsiveness” Matlab figure of variables X_4R_1, X_8R_3 and O_1.	/experiments/Scale/20150603171905/20150603171905_didnotrespond.fig

/experiments/<type>/<stamp>/<stamp>_presysmon.csv	Typeperf pre-processing system output.	/experiments/Scale/20150603171905/20150603171905_presysmon.csv
/experiments/<type>/<stamp>/<stamp>_prosysmon.csv	Typeperf processing system output.	/experiments/Scale/20150603171905/20150603171905_prosysmon.csv
/experiments/<type>/<stamp>/<stamp>_possysmon.csv	Typeperf post-processing system output.	/experiments/Scale/20150603171905/20150603171905_possysmon.csv

The “experiments” directory contains a sub-directory for each experiment “type” (“Scale”, “Topology and Stability” or “Environment”) for each experiment carried with experiment “reference” and “stamp” as unique identification.

Experiments do not always produce similar output artefacts.

7.3 Appendix: NPOST Database Schema



7.4 Appendix: NPOST Example Simulation Output (2.81/20150917152738)

///\\\

Experiment: <Reference>{:<Phase>/<Stamp>: 2.81/20150917152738

///\\\

****

MATRIX ANALYSIS

****

Symmetric: 1

Full Rank: 1

Diagonally Dominant: 1

Strictly Diagonally Dominant: 1

Positive Definite: 1

All Non-Zero Elements in Diagonal: 1

****

****

EXPERIMENT SUMMARY

Reference: 2.81

Stamp: 20150917152738

****

Experimental Variables

Framework

(X_1) Trust Function: TF1

(X_2) Complete Trust Space dimension: 1

(X_3) Initial Trust Value range: 0.0000 to 100.0000 (norm: 100.0000)

(X_4R_1) Initial Reputation Profile Trust Values:

Mean: 51.7000

Median: 54.0000

Standard Deviation (STD): 29.5773

Variance: 874.8182

(X_5) Initial Reputation Profile dimension: 100

(X_6R_2) Environmental Factors:

Mean: 0.3333

Median: 0.3333

Standard Deviation (STD): 0.0000

Variance: 0.0000

Range: 0.2632 to 0.3911 (norm: 0.1279)

Dimension: 3

Symmetric: 1

Similar: 1

Iterative change: 0

Iterative Method (JOR) Algorithmics

(X_7) Stability readjustment scheme: 0

(X_8) Determinant for a node’s availability in the system for an iteration (percentage):

Static non-responsive (0): 1

Static responsive (1): 99

Uniformly pseudorandom (2): 0

(X_9) Convergence condition or error tolerance: 0.00010000

(X_10) Upper iteration bound: 100

(X_11) Relaxation parameter: 1.0000

Observational Variables

(O_1) Number of algorithm iterations: 100

(O_2) Norm of the differences between subsequent iterations over time:

Mean: 22.9056

Median: 22.2885

Standard Deviation (STD): 15.3597

Variance: 235.9214

(O_3) Computation real execution time (seconds): 0.03351216

(O_4) Final Reputation Profile Trust Values:

Mean: 30.8219

Mean difference (percentage): 40.3833

Median: 31.7115

Standard Deviation (STD): 11.5336

```
   Variance: 133.0230

   Range: 10.6658 to 49.6394 (norm: 38.9736)

   Range difference (percentage): 61.0264

   ANOVA1 p: 0.0000

      Reject H_0: Initial != Final

      Iterative Reputation Profile means are significantly different (p <= 0.05).

(O_5) Stability: divergence (1)

**********************************************************************
```

7.5 Appendix: NPOST System Monitoring Counters

```
"\Memory\% Committed Bytes In Use"

"\Memory\Available Bytes"

"\Memory\Available KBytes"

"\Memory\Available MBytes"

"\Memory\Cache Bytes"

"\Memory\Cache Bytes Peak"

"\Memory\Cache Faults/sec"

"\Memory\Commit Limit"

"\Memory\Committed Bytes"

"\Memory\Demand Zero Faults/sec"

"\Memory\Free &amp;amp; Zero Page List Bytes"

"\Memory\Free System Page Table Entries"

"\Memory\Modified Page List Bytes"

"\Memory\Page Faults/sec"

"\Memory\Page Reads/sec"

"\Memory\Page Writes/sec"

"\Memory\Pages Input/sec"

"\Memory\Pages Output/sec"

"\Memory\Pages/sec"

"\Memory\Pool Nonpaged Allocs"

"\Memory\Pool Nonpaged Bytes"

"\Memory\Pool Paged Allocs"

"\Memory\Pool Paged Bytes"

"\Memory\Pool Paged Resident Bytes"

"\Memory\Standby Cache Core Bytes"

"\Memory\Standby Cache Normal Priority Bytes"

"\Memory\Standby Cache Reserve Bytes"

"\Memory\System Cache Resident Bytes"

"\Memory\System Code Resident Bytes"

"\Memory\System Code Total Bytes"
```

```
"\Memory\System Driver Resident Bytes"

"\Memory\System Driver Total Bytes"

"\Memory\Transition Faults/sec"

"\Memory\Transition Pages RePurposed/sec"

"\Memory\Write Copies/sec"

"\Process(_Total)\% Privileged Time"

"\Process(_Total)\% Processor Time"

"\Process(_Total)\% User Time"

"\Process(_Total)\Creating Process ID"

"\Process(_Total)\Elapsed Time"

"\Process(_Total)\Handle Count"

"\Process(_Total)\ID Process"

"\Process(_Total)\IO Data Bytes/sec"

"\Process(_Total)\IO Data Operations/sec"

"\Process(_Total)\IO Other Bytes/sec"

"\Process(_Total)\IO Other Operations/sec"

"\Process(_Total)\IO Read Bytes/sec"

"\Process(_Total)\IO Read Operations/sec"

"\Process(_Total)\IO Write Bytes/sec"

"\Process(_Total)\IO Write Operations/sec"

"\Process(_Total)\Page Faults/sec"

"\Process(_Total)\Page File Bytes"

"\Process(_Total)\Page File Bytes Peak"

"\Process(_Total)\Pool Nonpaged Bytes"

"\Process(_Total)\Pool Paged Bytes"

"\Process(_Total)\Priority Base"

"\Process(_Total)\Private Bytes"

"\Process(_Total)\Thread Count"

"\Process(_Total)\Virtual Bytes"

"\Process(_Total)\Virtual Bytes Peak"

"\Process(_Total)\Working Set"

"\Process(_Total)\Working Set - Private"
```

```
"\Process(_Total)\Working Set Peak"

"\Process(MATLAB)\% Privileged Time"

"\Process(MATLAB)\% Processor Time"

"\Process(MATLAB)\% User Time"

"\Process(MATLAB)\Creating Process ID"

"\Process(MATLAB)\Elapsed Time"

"\Process(MATLAB)\Handle Count"

"\Process(MATLAB)\ID Process"

"\Process(MATLAB)\IO Data Bytes/sec"

"\Process(MATLAB)\IO Data Operations/sec"

"\Process(MATLAB)\IO Other Bytes/sec"

"\Process(MATLAB)\IO Other Operations/sec"

"\Process(MATLAB)\IO Read Bytes/sec"

"\Process(MATLAB)\IO Read Operations/sec"

"\Process(MATLAB)\IO Write Bytes/sec"

"\Process(MATLAB)\IO Write Operations/sec"

"\Process(MATLAB)\Page Faults/sec"

"\Process(MATLAB)\Page File Bytes"

"\Process(MATLAB)\Page File Bytes Peak"

"\Process(MATLAB)\Pool Nonpaged Bytes"

"\Process(MATLAB)\Pool Paged Bytes"

"\Process(MATLAB)\Priority Base"

"\Process(MATLAB)\Private Bytes"

"\Process(MATLAB)\Thread Count"

"\Process(MATLAB)\Virtual Bytes"

"\Process(MATLAB)\Virtual Bytes Peak"

"\Process(MATLAB)\Working Set"

"\Process(MATLAB)\Working Set - Private"

"\Process(MATLAB)\Working Set Peak"

"\Processor(_Total)\% C1 Time"

"\Processor(_Total)\% C2 Time"

"\Processor(_Total)\% C3 Time"
```

```
"\Processor(_Total)\% DPC Time"

"\Processor(_Total)\% Idle Time"

"\Processor(_Total)\% Interrupt Time"

"\Processor(_Total)\% Privileged Time"

"\Processor(_Total)\% Processor Time"

"\Processor(_Total)\% User Time"

"\Processor(_Total)\C1 Transitions/sec"

"\Processor(_Total)\C2 Transitions/sec"

"\Processor(_Total)\C3 Transitions/sec"

"\Processor(_Total)\DPC Rate"

"\Processor(_Total)\DPCs Queued/sec"

"\Processor(_Total)\Interrupts/sec"
```