Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project Report 2016

Project Title:	**Self-Organising Error Detection and Correction in Open Multi-Agent Systems**
Student:	**Mark Zolotas**
CID:	**00738987**
Course:	**EIE4**
Project Supervisor:	**Prof Jeremy V. Pitt**
Second Marker:	**Dr Krystian Mikolajczyk**

Abstract

A common challenge in open multi-agent systems (MAS) is the requirement for sustainable distribution of computing resources amongst internal components, without relying on a centralised controller. One approach to this decentralised resource management is for the agents themselves to self-determine a resource distribution scheme. However, agent error may arise under these adaptive conditions due to misunderstanding, necessity, or malice. To compensate for these expected errors, the open system must include a rule-set that encapsulates corrective action by monitoring non-compliant behaviour, enforcing sanction policies, and employing a conflict-resolution mechanism. As a result, this project seeks to explore the impact of these mechanisms for restoring compliance on the sustainability of resources, whilst ensuring that they are congruent with the state of the local environment and its autonomous population. For this purpose, a self-organising rule-oriented MAS has been designed and implemented using an in-house Java-based simulator, Presage2. Experiments with this multi-agent testbed concluded that self-organisation of error detection and correction protocols can improve the overall utility of the common collective and significantly downgrade the propensity for non-compliant behaviour.

Acknowledgements

I would like to express my gratitude to my supervisor, Jeremy, for his expert advice throughout the project, continued support, and good humour.

Contents

List of Figures

Abbreviations

MAS - Multi-Agent System(s)
CPR - Common-Pool Resource(s)
SOEI - Self-Organising Electronic Institution(s)
DoF - Degree(s) of Freedom
LPG - Linear Public Good
DBMS - Database Management System

1 Introduction

1.1 Motivation

A common requirement in open systems is the collective action of organising and distributing computing resources amongst constituent agents of the system. Typical examples of systems performing this collectivised resource management protocol include ad hoc networks [1] and distributed systems for cloud and grid computing [2][3]. Within such systems, resources are provisioned via system components and are therefore endogenous, as opposed to exogenous, whereby an external source would control the process of supply.

The lack of a centralised controller or command structure in open systems is a fundamental feature pertaining to the endogenous nature of 'openness'. Given the overwhelming complexity, frequency, and speed necessary in the decision-making surrounding the distribution of resources, relying on external operator intervention is infeasible. Instead, open systems take a decentralised approach to resource management, for which a suitable candidate system to accomplish this feat is a *self-organising* multi-agent system (MAS). Through self-determination of the resource distribution process, internal system actors may be able to locally adapt to conventional rules through cooperation or synchronisation between one another, and thus meet transient conditions more effectively.

However, under such conditions there is the possibility of sub-ideal component behaviour due to self-interested goals that contradict and potentially impede the common goal of the system. In particular, the adaptive behaviour of agents sharing resources for a collective purpose may lead to agent error through non-compliance with the self-determined resource allocation rules, which could stem from either misunderstanding or malice. For instance, an actor might intentionally choose to 'cheat' by demanding more resources than it actually needs. The effects of this action can subsequently affect the behaviour of observing agents depending on whether the actor responsible of misconduct is both convicted and appropriately sanctioned, or goes unpunished.

To compensate for these expected errors, the open system must perform corrective action as a means for restoring a 'normative' state. This desired functionality ratifies the need to monitor agent behaviour, enforce sanction policies, and establish a dispute-resolution mechanism. Determining how to employ this set of norms within a system of autonomous agents can play a crucial role in the sustainability of the shared resource. For example, harsh sanctioning of non-compliance could diminish overall utility in the population and so poses a risk to the pool of resources. Additionally, monitoring expenses are 'paid for' via endogenous resources extracted from this resource pool, as there is no full disclosure in an open system and internal state information of components cannot forcibly be revealed. Hence, excessive or insufficient monitoring costs are also viable threats to the survival of the collective.

The field of research concerned with addressing these issues from a formal characterisation of different features of 'justice' is known as *computational justice* [4]. Computational justice is an inter-disciplinary approach to the study and observation of formal representations of justice in computer science, with overlapping concepts of philosophy, economics, psychology, and jurisprudence [4]. Within the domain of this research programme, ideas extracted from the social sciences can be applied to formalise theories about different forms of justice. Tackling the issue of how to organise error recognition and sanctioning in open systems coincides with notions of *retributive justice*, a category of justice concerned with investigating how to formulate a system of punishment for non-compliant behaviour.

Therefore, the objective of this project is to design and implement a self-organising MAS, with which to explore the various trade-offs that mechanisms of retributive justice impose on ensuring sustainability of the common collective. In this context, a set of rules must be defined to encapsulate protocols for monitoring, sanctioning, and conflict-resolution. These protocols should also take into account the nature of the open system, by maintaining congruence with the multi-agent environment and its autonomous population. Under these circumstances, the principal motivation behind this project is to seek an adaptive balance between self-organising norms of error detection and correction such that there are improvements in maximising compliance pervasion, utility, and longevity of the collective state.

1.2 Project Scope

The scope of this project revolves around the development and operation of an artificial social system that will act as a platform for conducting a study of self-organising retributive justice in sustainable resource management. As the primary aim of this study is to determine a state of harmony for recognising and dealing with expected error in open systems, the rule-set used to specify these norms is a crucial project deliverable. Another vital deliverable is the framework employed to characterise agent behaviour and decision-making, which is introduced in this report as the SpinWorld framework.

In the development of any computational social system, the formal theory of social sciences plays a significant role. For this project, the resource allocation problem in open systems is examined from an institutional point of view and is modelled as a common-pool resource (CPR) problem. From socio-economic theory, Elinor Ostrom identified eight 'institutional design principles' for self-governing CPR management [5]. These principles are the essence of this project's proposed solution for designing mechanisms of retributive justice to help assure continued survival of Self-Organising Electronic Institutions (SOEI). Furthermore, the social relations between individual entities of an SOEI are modelled as a social network.

To fulfil the core objectives of this project, the simulation of a MAS is performed to analyse different parameters and their corresponding effects on the challenge of decentralised resource distribution. An in-house simulator, Presage2[1] [6], is used to build the multi-agent environment that drives all experiments operated throughout this research project. This

[1] `https://www.presage2.info`

Java software is an agent-based animation and simulation tool, which has been selected due to its ease of implementation with respect to institutional and conventional structures prevalent in social systems [6]. Integrated into the Presage2 architecture is a rule engine that can be utilised to formulate and execute protocols of institutions.

By incorporating Ostrom's principles for monitoring, sanctioning, and dispute-resolution into a framework for SOEI, this project aspires to find an equilibrium state of sustainability that will increase collective utility, maintain costs, and downgrade non-compliant behaviour. Prior work in the field of computational justice has looked into the issue of endogenous resource allocation in open systems from the perspective of Ostrom's principles, by investigating the effects of monitoring and sanctioning non-compliant behaviour on CPR management [7]. This project can be considered an extension of their preliminary work through the development of a computer model that is capable of simulating and evaluating the automation of retributive justice in open electronic institutions.

1.3 Report Structure

The project report is structured in the following manner. Section 2 provides a background on the key features of open MAS, the socio-economic theory behind decentralised resource management, the concepts of psychological-based mechanisms for punishment, and the use of social networks for modelling dynamic multi-agent communities. An overview of a methodology for engineering social systems using the Presage2 simulator is detailed in Section 3 and related to the target requirements of this project. Section 4 introduces a formalised model of the social system to explore the study of self-organised error detection and corrective action in open systems. A structured design of this model is presented in Section 5 for the purpose of acquiring an operational model that can be simulated using Presage2. Section 6 describes the implementation of the target system; and Section 7 evaluates its performance and results under controlled experimentation. Limitations of this investigation and future work are addressed in Section 8, whilst a summary of the findings and conclusions of this project are provided in Section 9.

2 Background

This section introduces the relevant background necessary for the analysis of self-organised error detection and correction in open MAS. The main characteristics of open MAS are initially reviewed in relation to the fundamental issue of non-compliant agent behaviour in endogenous resource allocation. Situating this issue in terms of a CPR problem, Ostrom's theory of self-governance [5] and Artikis' computational framework [8] for specifying 'dynamic' norm-governed systems are discussed in order to define an SOEI. A variant of the Linear Public Good (LPG) game [9] is subsequently depicted as a testbed for analysing resource distribution in SOEI. Mechanisms for punishment and retributive justice are also presented using concepts of psychology and economics. Finally, social networks are briefly explored for their influence on modelling conceptual relations between entities of an electronic institution.

2.1 Open Multi-Agent Systems

'Open' systems are generally characterised in this report according to the work of Hewitt [10]. In Hewitt's paper, open systems are specified to contain numerous components of heterogeneous origins dealing with diverse information. There is no "centralised decision maker" included in this type of system, which means control is decentralised and distributed throughout the system. The components thus make local decisions amongst each other and can explicitly communicate these actions. However, computational agent behaviour is stated to be unpredictable due to the concealed internal activities of participants from other participants. In addition to this behaviour, constituent agents can choose to join or leave the system at any particular time.

MAS consisting of a set of intelligent agents interacting with an environment can be extended to encompass this characterisation of 'openness'. Following from Hewitt's semantics [10], the members of an open MAS are treated as *autonomous* and *heterogeneous*, indicating the lack of constraints in the behaviour and construction of agents respectively [11]. These features reflect the independence of the agents from theirs users and designers, leaving the MAS to essentially be considered as a set of specifications or protocols that its member agents must abide by [11]. Nevertheless, agents of open systems display 'untrustworthy' behaviour and do not necessarily comply with these MAS specifications. Non-compliance of this kind may be due to agent failure by acting on behalf of parties of competing interests, or due to their own individual goals [12]. A typical example application of an 'open' MAS are electronic marketplaces [13].

2.2 Self-Organising Electronic Institutions

2.2.1 Self-Governing The Commons

According to game-theory analysis, many scenarios involving a group of (human) actors pooling their resources in a self-organising institution will lead to an inevitable depletion of the common pool due to individual incentives overpowering the collective need for sustainability. Hardin stated in his article 'the tragedy of the commons' [14] that whenever multiple rational and independent individuals are required to share a scarce resource, they will actively pursue their own interests and defy the common good by draining the resource in the short-term. Similarly, Olson argued in the 'zero contribution thesis' that unless there was some form of centralised authority, or a small number of individuals, then rational and self-interested individuals will not cooperate to achieve their group interests [15, p. 2].

As a rebuttal to this predicted depletion of resources, Ostrom [5] presented numerous examples of communities that avoided the 'the tragedy of the commons' and the 'zero contribution thesis'. In these examples, the "evolution" of institutions was a successful method of self-governing CPR problems. A CPR is any natural or man-made resource system whose size is sufficiently large, such that it is costly (but not impossible) to exclude potential beneficiaries deriving benefit from its accessibility (e.g. fisheries, forests) [5, p. 30]. 'Institutions' are defined as the sets of "working rules" used to determine within some arena the eligibility for decision-making, constraints on actions, procedures to follow, information flow, and payoffs to individuals [5, p. 51]. Each of these rules contains prescriptions to forbid, permit, or necessitate an action or outcome [5, p. 51].

Ostrom distinguished between three levels of "working rules" [5, p. 52] for self-governing institutions. At the lowest level, operational-choice rules relate to processes of resource appropriation, provision, allocation, monitoring, and enforcement. Collective-choice rules are at the middle layer and are concerned with selecting and adapting the operational-choice rules for policy-making, adjudication, and CPR management. At the highest level, constitutional-choice rules specify eligibility of institution members and determine the collective-choice rules. Each of these levels of rules is mutually agreed, liable to change, and all are nested within each other.

Although Ostrom's fieldwork depicted successful occasions for sustainability in communities collectively managing their resources to avoid game-theoretical outcomes, simply forming an institution would not guarantee this outcome. Consequently, eight conventional rules were identified to regulate and coordinate agent behaviour for enduring self-management of the commons. These principles are summarised in Table 1 [5, p. 90]. A more recent meta-review only made minor clarifications over these principles [16].

The target MAS of this project intends to explore the balance between principles of retributive justice (Principles 4, 5, and 6), whilst adhering to the principle of congruence within an institution (Principles 2). In practice, these sets of rules are rarely, if ever, in harmony with one another.

Table 1: Ostrom's principles for enduring institutions [5, p. 90].

1	Clearly defined boundaries: those who have rights or entitlement to resources from the CPR must be clearly defined, as must the boundaries of the CPR.
2	Congruence between appropriation and provision rules and the state of the prevailing local environment.
3	Collective-choice arrangements: those affected by the operational rules can participate in the modification of these rules.
4	Monitoring of both CPR conditions and appropriator behaviour is by appointed authorities, who are either accountable to appropriators or are appropriators.
5	Graduated sanctions: assessed graduated sanctions for resource appropriators who violate operational rules.
6	Conflict-resolution mechanisms: appropriators have access to fast, low-cost mechanisms to resolve conflicts.
7	Appropriator rights to devise and govern institutions are not challenged by external authorities.
8	Systems of systems: layered CPR, multiple layers of nested enterprises.

2.2.2 Dynamic Norm-Governed Systems

Artikis defined a dynamic framework for agents of a norm-governed open system to modify the rules at runtime [8]. The framework has three major components: a specification of the norm-governed system, a set of protocol levels for defining changes to this specification, and a "metric space" for expressing the separation "distance" between two specification instances.

The study of social systems is often formalised into the specification of a norm-governed system. Prior research has investigated the main features of norm-governed systems and made well-established distinctions between them, such as the distinction of "institutionalised power" from "permission" and "practical possibility" [17]. Pitt *et al.* extended this standard to express five aspects of social constraint in the partial specification of a norm-governed system: physical capabilities, institutionalised powers, permissions, sanctions and enforcement policies upon non-permitted behaviour, and designated roles of empowered agents [18]. Consequently, outlining these five constraints in the system provides the first component of the framework.

With regard to the second component, the protocol stack is an infrastructure for dynamic specifications. This infrastructure allows the agents to alter the rules of an 'object' protocol during execution by starting a 'meta' protocol, which determines whether to modify the object-level protocol or not. In the same way, participants of the meta protocol could initiate a 'meta-meta' protocol to modify the specification of the lower meta-level protocol, and so forth at higher levels. Aside from the object- and meta- protocol layers, 'transition' protocols are also included. Transition protocols define the conditions for an agent to successfully initiate a meta protocol, the roles of each agent in the meta protocol, and the methods of modifying an object protocol in response to the meta protocol initiation.

In Artikis' framework, a degree of freedom (DoF) refers to any specification parameter that is modifiable at runtime [8]. An l-dimensional specification space is created from a specification with l DoF. A "specification point" or "specification instance" is denoted by an l-tuple, with each DoF *value* representing an element of this tuple. Taking the set of all possible instances and computing a 'distance' metric between specification points forms the metric space, which models dynamic specifications. Within the metric space, social constraints can be expressed via this separation distance measure.

2.2.3 Institutions as Dynamic Multi-Agent Systems

Although the framework can specify a diverse range of systems, this project is only concerned with describing a norm-governed MAS that conforms to the principles for enduring institutions. Accordingly, the target system requires this framework to encapsulate Ostrom's definition of an institution and the three-layered set of "working rules" mentioned in Section 2.2.1, as well as her principles for enduring institutions (Table 1). By casting the three-layered rule-set to the stack of protocols, the rules for self-governing the commons can be mapped to the computational framework of dynamic norm-governed systems. Furthermore, Pitt *et al.* [18] proposed a formal axiomatisation of Ostrom's institutional design principles in terms of the concepts of a norm-governed system.

In the scope of this project, the definition of an SOEI is a collection of agents plus a dynamic specification of a norm-governed open MAS. A rule-set adhering to the theory of enduring institutions is incorporated into this specification for the purpose of realising self-* features, such as self-organisation and self-regulation. Accordingly, an SOEI is essentially an electronic portrayal of Ostrom's self-governing institutions. A formal model of a MAS that uses SOEI to tackle the problem of endogenous resource allocation in open systems is provided in Section 4.

2.3 Linear Public Good Game

A useful mechanism for analysing the problem of individual resource contribution in a CPR is the Linear Public Good (LPG) game [9]. The LPG game is well-known in economics and computer science for testing the issue of voluntary contributions, which arise from the free-riding phenomenon. A free-rider is an actor that benefits from the resources provisioned by other actors to the CPR, but does not itself contribute to this public good.

In a typical LPG experiment, n people (agents) form a group or 'cluster'. All cluster members are endowed with a quantity of some divisible resource. Each subject $i, i \in 1 \ldots n$, determines independently a fraction of their resource, $r_i \in [0, 1]$, to contribute to the public good (or CPR). The contributions of the whole cluster are summed up, and the payoff u_i for each subject i is given by:

$$u_i = \frac{a}{n}\sum_{j=1}^{n} r_j + b(1 - r_i), \text{ where } a > b \text{ and } \frac{a}{n} < b$$

The first term is the payoff from the public good, which is distributed equally amongst the members (the 'public payoff'). The second term is the payoff from the resources withheld from the CPR (the 'private payoff') and is irrespective of the contribution made by the individual. Therefore, regardless of whether or not an agent chooses to devote a portion of their resources to the public good, they will benefit from the contributions of other members within the cluster. Coefficients a and b represent the relative values of the public and private payoffs, respectively. Under these conditions, a rational and selfish player has an incentive to withhold all of their resources from the public good; this is the dominant strategy.

Nonetheless, the LPG game makes three assumptions that contrast with the key features of open systems. Namely, the LPG game assumes the following: the agents do not cheat on appropriation, the utility of all resources is identical, and that there is full disclosure of information surrounding a decision that carries risk. In order to model the resource allocation problem in open systems using this game, these assumptions must be relaxed to take into account the possible erroneous behaviour of agents on appropriation, 'diminishing returns', and the lack of full disclosure.

A variant of this game, LPG$'$, has been proposed for utilisation in the context of open systems [7]. This version of the game is played in rounds $t_0, t_1, \ldots, t_\infty$. In each round t (omitting the subscript for clarity), player i:

- Determines its *available* resources, $g_i \in [0,1]$
- Determines its *need* for resources, $q_i \in [0,1]$ $(q_i > g_i)$
- Makes a *demand* for resources, $d_i \in [0,1]$
- Makes a *provision* of resources, $p_i \in [0,1]$ $(p_i \leq g_i)$
- Receives an *allocation* of resources, $r_i \in [0,1]$
- Makes an *appropriation* of resources, $r'_i \in [0,1]$

A paper further elaborating this variant enumerates the game as above [19] and outlines the following equations to describe its operation. Of considerable importance is the fact that the LPG$'$ game assumes an *economy of scarcity* by constraining the need for resources to always exceed the amount an agent generates for itself ($q_i > g_i$ inequality for agent i). This establishes a necessary dependency between the players of the game, as well as an incentive for non-compliance.

Total resources accrued by an agent i at the end of a round is denoted as R_i. This variable is equivalent to the sum of resources appropriated from the public good and available resources withheld from the common pool:

$$R_i = r'_i + (g_i - p_i)$$

Utility of agent i is given by:

$$U_i = \begin{cases} aq_i + b(R_i - q_i), & \text{if } R_i \geq q_i \\ aR_i - c(q_i - R_i), & \text{otherwise} \end{cases} \tag{1}$$

where $a, b, c \in \mathbb{R}$ are coefficients that measure, respectively, the relative utilities of getting resources that are needed, getting resources that are not needed, and not getting resources that are needed. In the game, appropriated resources r' do not accumulate from one round to the next.

An agent i in cluster C also *subjectively* evaluates its *satisfaction*, $\sigma_{i,C} \in [0, 1]$. This metric is independent of the agent's utility U_i, and is computed according to the relation between its allocated and demanded resources. If an agent is allocated at least the amount it demands in the current round, then $\sigma_{i,C}$ increases in the following round, else it will decrease:

$$\sigma_{i,C}(t+1) = \begin{cases} \sigma_{i,C}(t) + \alpha(1 - \sigma_{i,C}(t)), & \text{if } r_i \geq d_i \\ \sigma_{i,C}(t) - \beta\sigma_{i,C}(t), & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha, \beta \in [0, 1]$ are coefficients determining the rate of reinforcement of satisfaction and dissatisfaction, respectively. These parameters effectively offer a way to model player "personalities" in the game, e.g. a high α and a low β would design an overly optimistic agent [20]. A threshold or cutoff value, τ, and an interval value or dissatisfaction limit, m, are also defined. If an agent i assesses that $\sigma_{i,C}(t) < \tau$ for m consecutive rounds, then this agent will leave the cluster C. Moreover, a player will not rejoin any previously left clusters.

Previous research has explored the issues in endogenous resource allocation for CPR management using the LPG$'$ game as an experimental setting [7][19][21]. Each of these investigations was focused around a specific category of justice, as part of the computational research programme. In particular, a preliminary system of retributive justice that conformed to principles of self-governing institutions addressed the problem of non-compliant behaviour in open systems [7]. This project follows in their footsteps and tackles the problem from a similar angle, by specifying a novel model of error monitoring and corrective-action in SOEI to play the LPG$'$ game. The game presents an experimental setting of the resource distribution scenarios often found in open systems.

Experimental results obtained from using this variant of the LPG game are presented in Section 7 of the report. In spite of the fact that coefficients (a, b, c, α, β) and satisfaction threshold values (τ, m) are not necessarily the same across all players, the experimental work of this project configures these simulation parameters as global values.

2.4 Mechanisms of Punishment

From a psychological point of view, the diversity of institutions and people begs to question the motivation behind human desire to punish non-compliant action. Carlsmith and Darley addressed this matter by seeking the origin of cultural tendency for human communities to propose punishment in accordance with crime [22]. Their findings suggested that intuitions of justice in the general public are consistent with principles of retributive justice. Retributive justice is one of the many different qualifiers of 'justice', for which a wrongdoing or immoral action necessitates punishment in proportion to the moral magnitude of the crime. However, an examination of a modern-day legal system indicated an increasingly utilitarian-based approach to punishing wrongs, in which sanction policies are designed according to principles of deterrence.

Moral philosophy traditionally debates the justification in permitting a community to punish its members, by distinguishing between two schools of thought: *retributivism* and *utilitarianism*. Kant [23] best captures the retributivist perspective by asserting that punishment in *proportion* to the severity of an offence is both just and necessary in order to maintain social control, with no pardon for future redemption of the offender. Under this notion of an offender facing retribution for what they deserve, no other principle is a legitimate ground for punishment. By contrast, Bentham [24] founded a utilitarian outlook on justice, which argued that punishment of a wrongdoer should be assessed by the overall gains and losses of utility for the concerned society. Punishment is thus deemed moral under the circumstance of society benefitting, with the loss of utility to an offender acting as a *deterrent* to recidivism, whilst discouraging other agents from replicating a similar offence.

Regardless of the justification chosen for sanctioning wrongdoers, there will be certain practical issues to address in designing a mechanism for punishment. These issues range from deciding when an action constitutes a violation of institutional regulations, what level of severity to impose in sanctioning such an action, and how to assign a form of punishment. For example, if a violation of institutional rules were to necessitate a penalty, then should this breach of 'law' be punished proportionately, or does the possibility for rehabilitation mandate forgiveness? Moreover, should the sanction applied in this scenario be a warning to the agent responsible, or is expulsion from the community a compulsory outcome?

In this context, Ostrom's view on self-governing institutions claims that a system of graduated sanctions must be enforced as one of the conditions for enduring CPR management (Principle 5 in Table 1). Graduated sanctions refer to a layered set of sanctions that advance in some determined scale of punishment upon recurrence of communal rule violations. For instance, if a member of the institution commits their first offence, then the level assigned to this player is set to one and their power to demand from the common pool is temporarily withdrawn. Likewise, if this same player commits a second offence, then the level is further raised and they may even be excluded from the institution altogether once a specific level is attained. This mechanism of punishment displays a strong correlation with a framework of retributive justice.

Principles 4 and 6 in Table 1 are closely linked with this framework. Principle 4 is concerned with assuring that appointed authorities of an institution monitor the process of resource allocation with respect to the local environment and observe individual agent behaviour. Any cost associated with the monitoring procedure is extracted from the common pool. Principle 6 declares that a fast, low-cost, and effective conflict-resolution mechanism should be in place for preserving sustainable resource management. Nonetheless, Ostrom notes that whilst there is no guarantee for enduring institutions in the presence of a dispute-resolution mechanism, the fact that this principle could impact the complex system of rules is evident [5, p. 101]. These features of retributive justice, alongside other categories of justice (e.g. distributive, procedural), are discussed in greater detail in the paper that introduces the discipline of computational justice for open self-organising systems [4].

2.5 Social Networks

A system that is represented as a network (or graph) consists of a set of nodes (or vertices) that are interconnected via edges. Given the key features of open MAS, a well-grounded proposal is to model these systems as agent societies or electronic institutions [25] [26]. Furthermore, the comparison of agent-based systems with human social structures inspires component interactions within these distributed systems to be modelled as social relations [27]. As a result, these social links between agents act as a formal representation of social networks.

The adaptive topological features of social networks have been widely explored in graph theory. Statistical quantities, such as the average path length between nodes, clustering coefficient [28], and degree distributions [29] of interconnections play a critical role in characterising the structure and dynamics of complex social networks [30]. Various approaches to modelling dynamic social networks that display these qualities in MAS exist, however this project primarily concerns itself with the collision-based models specified in the research of Gonzalez *et al.* [31][32]. These articles present a mobile-agent platform that attempts to capture multiple attributes of social networks, such as clustering, and community-like structure [30]. Segments of their design are exploited in the target system of this project to replicate the dynamic formation of institutions, as well as to manipulate the physical state of constituent agents.

Influenced by the properties of moving particles [31], mobile agents will form a link with one another upon collision i.e. create a social contact. The motivation behind this model lies in its similarity to online social networks, such as Facebook or Twitter, whereby the number of social interactions is directly correlated with the number of previous contacts. In other words, the more connected an agent is, the more likely for further acquaintance with other members of the community. For the project, this dynamic model will enable the segmentation of SOEI into isolated networks that avoid a fully connected graph situation, thereby enhancing the richness of these evolving social structures during simulation.

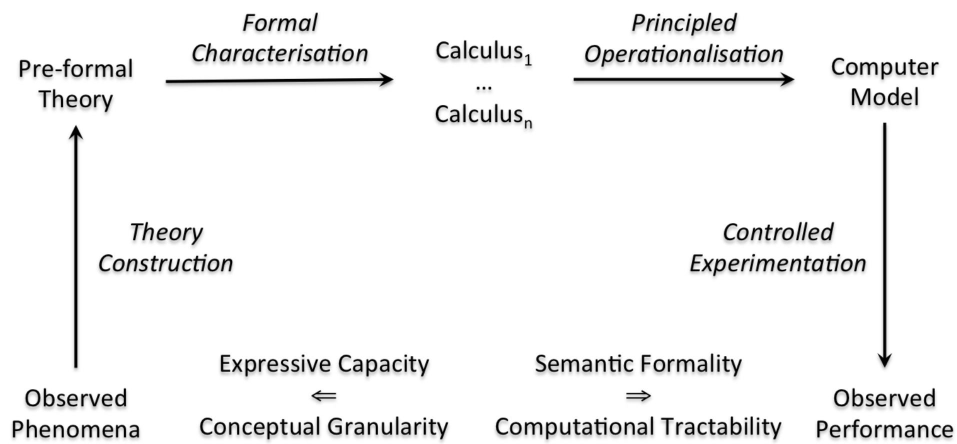


Figure 1: Methodology for sociologically inspired computing [34], with further clarifications taken from Macbeth *et al.* [6].

3 Simulating an Artificial Social System

A noteworthy reason for developing and operating a social system is to assemble computational systems that mimic human methodologies in order to address engineering challenges [33]. The issue of non-compliant behaviour in open MAS is an example of such a challenge. Therefore, this section first briefly describes a methodology for engineering artificial social systems. The multi-agent simulation platform, Presage2, is then examined from an architectural point of view, summarising a majority of the content covered in the chapter on principled operationalisation of social systems [6]. Lastly, this project's desired requirements in the design and execution of a self-organising MAS to explore a social system of retributive justice are outlined with regard to the discussed methodology and simulator.

3.1 Methodology

Modelling and simulating an artificial social system, such as one involving the self-organisation of retributive justice, commonly employs a *sociologically inspired computing* methodology [34]. A diagrammatic illustration of this methodology is shown in Figure 1 [34]. This methodology is fundamentally structured around the research underlying artificial societies and artificial life [35], the formalisation of social connections (e.g. reputation, trust, cooperation) in MAS [36][37], and several other attempts to form a link between concepts derived from the social sciences and the operational design of computational systems [33].

In order to apply the methodology demonstrated in Figure 1, an observation of a social phenomenon must initially take place. A simple example of observable social phenomena is gossip within a closed community. *Theory construction* is the process of making a hypothesis according to this observation, resulting in a pre-formal theory. *Formal character-*

isation then involves translating this theory into a form of 'calculus', where calculus refers to any abstract system of calculation or computation obtained from the manipulation of symbolic representations [34]. Formulations by means of calculus are represented in different phases, either theory-facing through expressive capacity or conceptual granularity; or implementation-facing through semantic formality or computational tractability. The *principled operationalisation* stage is concerned with converting a formal model of a social system into an operational computational framework that embeds intelligent agent behaviour under varied environmental circumstances. Finally, the computer model undergoes *controlled experimentation*, followed by a performance assessment.

3.2 Presage2 Simulator

Presage2 is an agent-based animation and simulation platform that was developed to support the process of *principled operationalisation*, which is defined in this context as the mapping between formal models of social systems and implemented systems [6]. This simulator encompasses a broad range of features suitable for the purpose of modelling social systems and subsequently generating simulation results. For instance, this Java-based tool offers unrestricted agent computational complexity for simulating physical environments in which there is inter-agent communication. Moreover, a comprehensive event-handling mechanism enables robust state representation and control over state changes.

Numerous other agent-based modelling tools possess characteristics similar to Presage2, as Nikolai and Madey evaluated in a general survey [38]. To name a few relevant and well-known examples: NetLogo [39], Repast [40], and MASON [41]. All three platforms differentiate themselves from one another through traits of complexity and graphical interface. NetLogo acts as an easy-to-use multi-agent modelling environment with an extensive range of graphical features, whilst Repast is a more comprehensive framework focused on creating, executing, and visualising agent-based simulations. Even further up the complexity scale is the MASON toolkit, which is a flexible and extensible simulation toolkit written in Java. Presage2 is most comparable with the MASON software in terms of their powerful capabilities and all code-based functionality [6].

From an architectural perspective, there are six fundamental packages collaborating together to control simulation execution (Figure 2), as listed in the referenced chapter [6]:

- Core — Controls the main simulation loop and any core functions.
- State Engine — Stores and updates the simulation state.
- Environment & Agent Libraries — Implementations of common-use cases that can be used in the environment and/or agent specifications.
- Communication Network Simulator — Emulates a dynamic, inter-agent communication network.
- Database — Enables storage of simulation data and results or analysis.
- Batch Executor — Tools to automate the execution of batches of simulations.

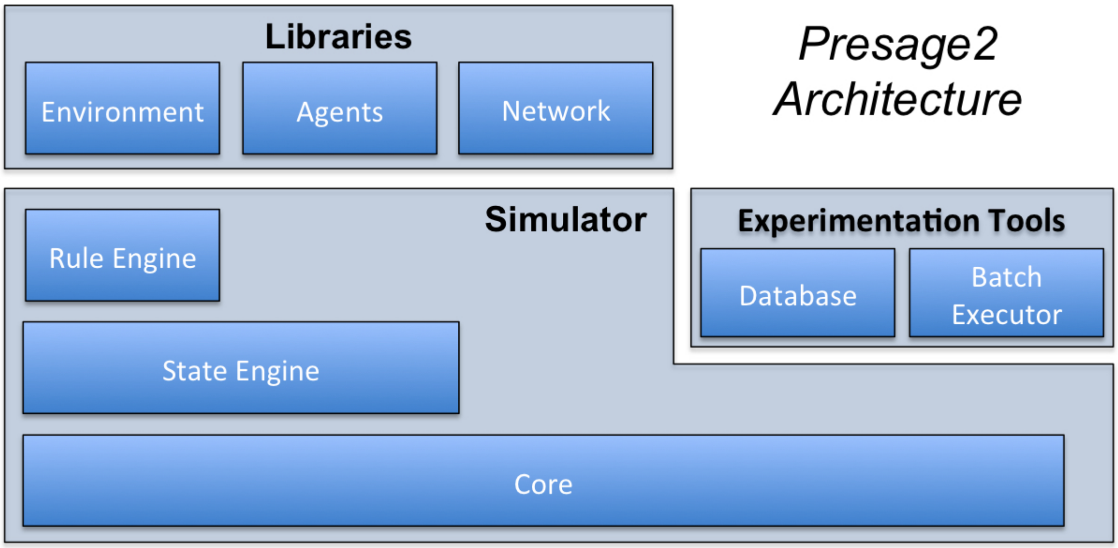


Figure 2: Presage2 architectural block diagram [6].

First and foremost is the simulator's core, which controls the main simulation loop and any functions related to the parameterised initialisation of a simulation. Running simulations in discrete time involves each loop representing a single time step. Within each time step, an agent can potentially act upon the environment and hence lead to an update in the simulation state. The unrestricted complexity of the architecture enables each agent's function to completely terminate before proceeding to the next time step. Furthermore, Presage2 exploits a multi-threaded implementation to utilise available computing resources efficiently during each time step. Consequently, agent implementations are executed in parallel using multiple threads wherever possible.

The state engine package is responsible for simulating a shared environmental state with every agent in the population. This package offers two functions for the user to manipulate: the capability for agents to perceive the current state of the environment, and the ability to determine a change in state following all agent actions performed in the preceding time step. These functions are denoted via modules, which are independently comprised of a set of rules for observability and/or detecting state updates. User-defined modules encapsulating the behaviour of agents can also be integrated into this composition of modules.

Environment modules can be defined in order to provide processed data to the observing agents. By querying the environment for information regarding other local agents, relationships between agents can be modelled in a dynamic manner. Figure 3 [6] displays the connectivity between agents and the environment modules, which mediate access to the environmental state.

An extension to the basic state engine implementation is the use of the JBoss Drools rule engine[2] as an underlying state storage machine. There are two powerful benefits de-

[2] `https://www.jboss.org/drools`

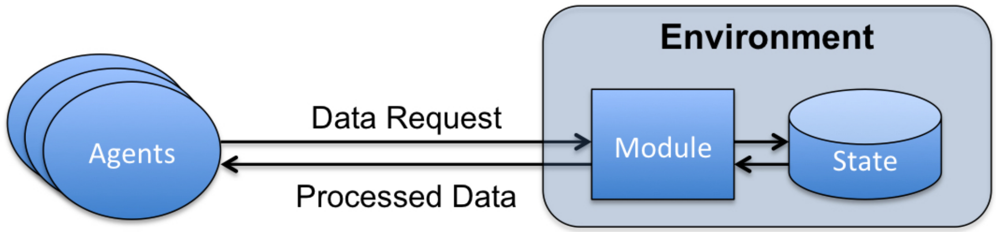


Figure 3: Relationship between agents and their shared environment through modules [6].

rived from the addition of Drools to this framework. Firstly, Drools provides a richer state representation, storing structured and relational data in the state, rather than storing raw data points with text strings for referencing. Secondly, Drools features a forward chaining rule engine, allowing sets of declarative rules to be created and triggered at each time step in order to modify the engine state during simulation runtime.

Implementations for common use cases are provided through the in-built environment and agent libraries, as well as the communication network simulator. The libraries include integrated functionality to create common structures for agents and environments within a simulation, such as mobile agents. On the other hand, the network simulator presents a basic physical communication layer for a small number of network types. This layered network allows agents to transmit structured data to one another in the form of a message.

A database package is included in the Presage2 platform to accommodate data visualisation and analysis. By persisting information of transient agent behaviour or environmental state changes, raw data is stored during every time step of a simulation. Several database management system (DBMS) drivers are available for use in this simulator, however this project will store data during experimentation using the object-relational PostgreSQL DBMS.

The batch executor is a tool for dealing with large batches of simulations, which avoids the need for supervised and repeated parameter instantiation. Automation of this process is accomplished by streamlining experimental runs over user-available network machines, or appropriately interfacing with a job scheduler. This allows simulations to be run in a parallel manner, making effective use of available resources and without resorting to high-performance computing infrastructure.

A clear illustration of how a simulation in Presage2 is created is demonstrated in Figure 4 [6]. This creation process is broken down into two phases. The design and implementation phase is responsible for collating an executable system specification layout that can be invoked by a string of parameters. A composition of available and user-defined modules are incorporated into this phase, which includes optional design choices such as selecting a network topology. This phase also requires the user to specify functional data types for their simulated agents. Following this procedure, the experimentation and evaluation phase involves running the simulation according to specification, whilst recording and generating a series of data sets to be analysed and visualised through use of external tools.

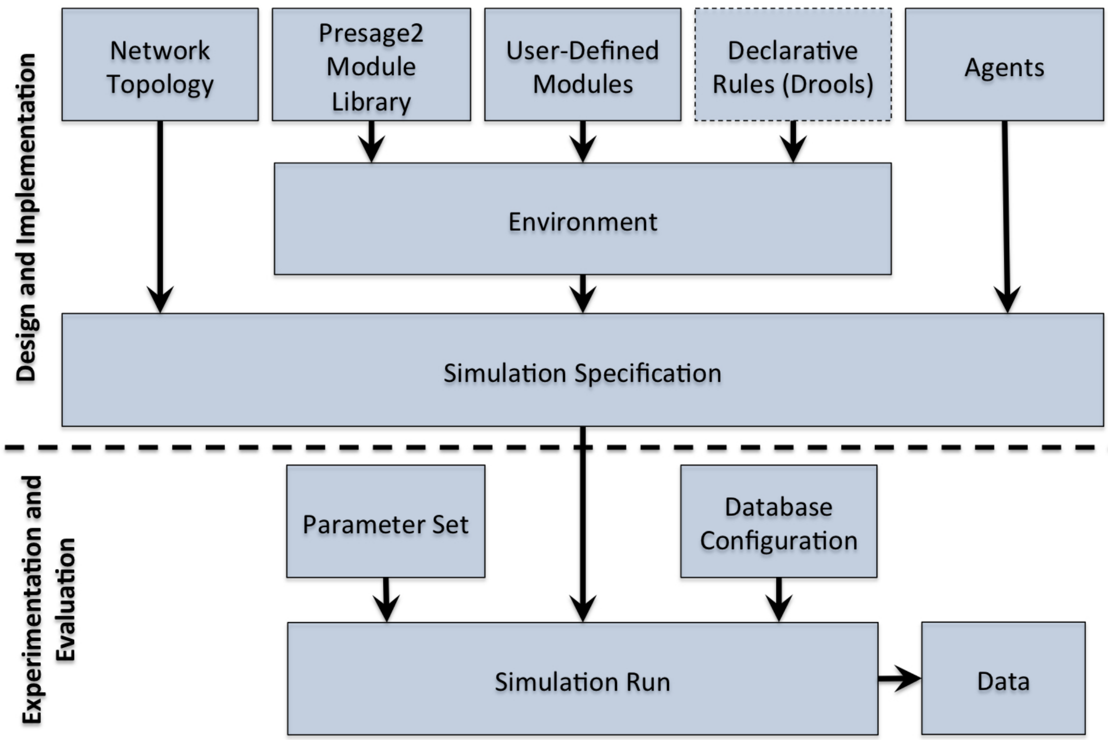


Figure 4: Creation process of a simulation in the Presage2 architecture [6].

3.3 Target System Requirements

The technical deliverable for this project is a social system of retributive justice that can be simulated using the Presage2 platform. To this end, the original challenge of non-compliant behaviour in open MAS is addressed through the evaluation of SOEI that contain rules for monitoring, sanctioning, and dispute-resolution, in order to avoid CPR depletion. This section of the report presents the desired set of requirements needed to obtain a queryable computer model with which to conduct this evaluation.

According to the sociologically inspired computing methodology (Figure 1), there are certain steps to follow before an operable computational model of an artificial social system may be acquired. Each of these different steps in development and analysis fulfil the expected specifications of the target system. A diagram summarising the stages of work-flow tailored towards this project is shown in Figure 5. This figure illustrates three core objectives:

1. *Formal Characterisation* — To propose a social system of retributive justice through formal tools of analysis, which will facilitate *expressive capacity* in articulation of the appropriate theory.
2. *Principled Operationalisation* — To generate a computer model that represents the artificial system; and can be built on top of the Presage2 platform.
3. *Controlled Experimentation* — To simulate the computational model in Presage2 for the assessment of sustainability in open electronic institutions.

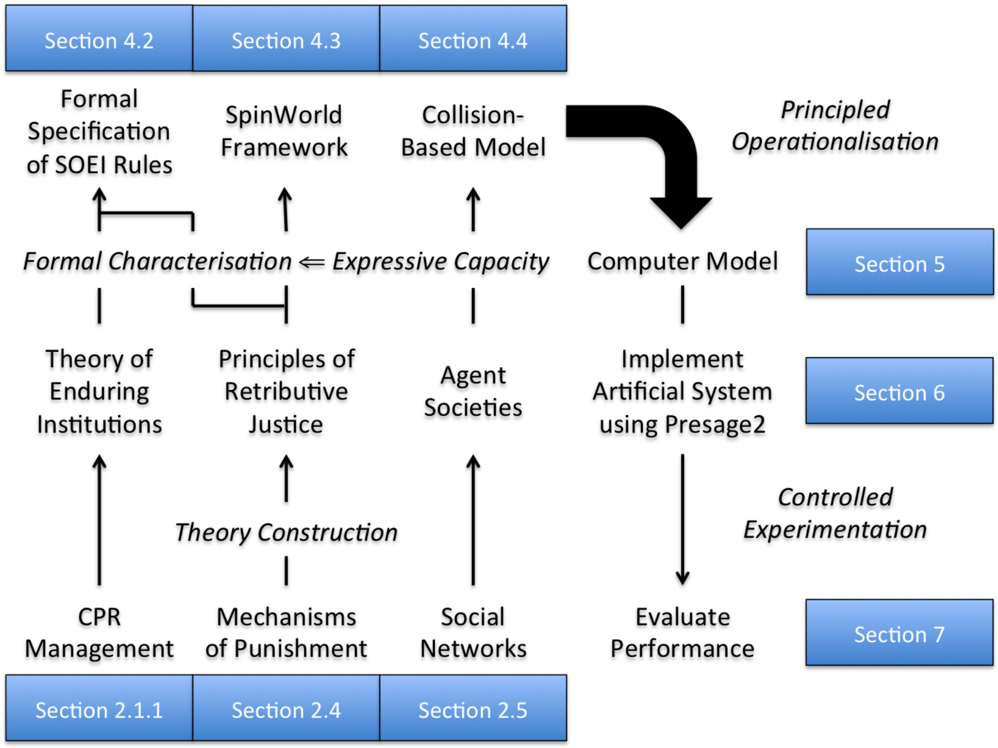


Figure 5: Project design-flow in terms of the sociologically inspired computing methodology (with reference to relevant sections of the report).

The first objective involves an in-depth analysis of the background theory presented in Section 2, for the purpose of delivering a comprehensive social system. To develop the proposed system, a formal model of a MAS must be defined. In an LPG′ game setting, this model will include a set of SOEI that are specified by their rules of resource distribution and retributive justice. Despite a strong focus on retributive justice in this project, mechanisms of punishment besides a system of graduated sanctions are also openly considered, such as the utilitarian principles of deterrence (Section 2.4). Members of the SOEI will use a framework, SpinWorld, to adapt their decision-making upon observation of institutional action against compliant and non-compliant behaviour. Lastly, the MAS should include a collision model for defining social network dynamics and agent interactions.

During the principled operationalisation step, the formal representations of the previous analysis must be embedded into a programmable format for simulation. For the LPG′ game, this will necessitate detailed implementations of agent and institutional structures, as well as declarative rules to govern the electronic institutions of retributive justice. Drools can be used to specify these rules in an executable manner that the environment can process in response to agent actions. At the LPG′ layer of implementation, the agents will be static and their actions purely devoted to resource management. On the other hand, the collision layer is concerned with the creation of mobile agent structures and user-defined environment modules that will embody a set of update equations for the formation of social networks. Figure 6 is a preliminary view of this two-layered computational architecture.

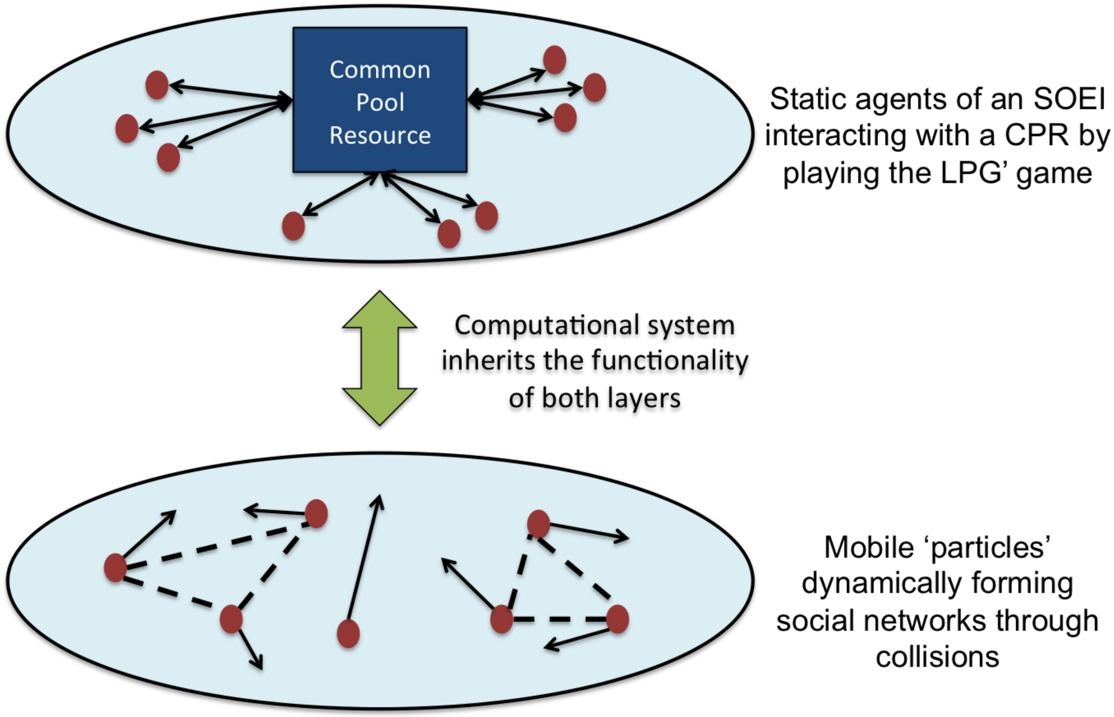


Figure 6: Forecasted computational model with a two-layered architecture composed of SOEI members playing the LPG′ game and dynamic social networks.

The network layer intended for this project should only represent a *conceptual* interface for subjective relations to be constructed between agents. A *physical* communication layer does not require development, as a number of libraries containing basic interfaces are currently available in the Presage2 toolkit.

Expressing agent societies as social networks represents a key aspect of the planned deliverable, however there is a more open-ended view on how this layer of the architecture should contribute to the primary investigation of this project. There is a vast range of experiments that can be simulated to explore the influence of social networks on sustainable institutions. One example would be to explore the effects of preferential attachment between agents, for which agents of authoritative power might be hypothesised to make unfair and lenient decisions on how to sanction close connections. Experiments that exploit the physical properties of networked agents can be exhaustively suggested. Hence, the project narrows the scope of investigation down to the set of rules used to govern an SOEI, as opposed to individual agent dynamics.

With regard to the final objective, a batch of coordinated simulations will be executed as a means to investigate the artificial social system under different parameterisations of error recognition and corrective action. These experiments will seek to validate hypotheses through visualisation of the system's performance over different configurations of independent variables. In all experiments, the main source of interest will lie in proving that a self-organising system of graduated sanctions can endure the resource allocation problem.

4 Formal Model

A formal model of the social system designed for this project will now be introduced through abstractions of calculus. Firstly, an explicit definition of a MAS is provided using SOEI, wherein constituent agents play n-player games and inform their decision-making under subjection to retributive rules. These institutional rules of retribution are then formally specified to obey Ostrom's principles. The SpinWorld framework is subsequently conveyed notationally as a framework of agent decision-making based on behaviour observation and propagation. Lastly, this section closes with a brief description of the collision model used to characterise network emergence in the target system.

4.1 Multi-Agent System

A generic model of an artificial system can be derived to address the challenge of resource management and non-compliance in open systems. Therefore, let the model of a MAS denoted as IC_t at discrete time step t be defined by the 6-tuple:

$$IC_t = \langle A, N, I, L, G, \delta \rangle_t$$

where:

- A is a set of agents.
- N is a *physical* network interface.
- I is a set of institutions.
- L is a norm-governed system specification.
- G is a set of games.
- δ is a dimension of the square shaped linear space.

A communication network, N, is included to provide an interface for agents A to send messages to other agents within their communication range. In addition to this physical network, a *subjective* layer of social networks that are represented as electronic institutions I are included. This institution set is instantiated as empty $\emptyset$ due to the dynamic design of building social networks in this model.

L is the dynamic specification space of the system, where each instance is defined by a different set of DoF *values* [8]. These DoF are configurable parameters that specify the operational- and collective-choice rules for a particular instance in the system. An example of a DoF is the method for resource allocation in an institution, by which its value could be configured as *random*.

In a generic context, G is a set of n-player games that are played by agents in the multi-agent simulation environment, for which the environment is constrained to $\delta \times \delta$ discrete cell locations.

Each institution (or social network) $i_t \in I_t$ is a 4-tuple:

$$i_t = \langle M, l, \epsilon, g \rangle_t$$

where:

- M is a set of members and a subset of A.
- l is a specification instance of L.
- ϵ is the local environment represented as the pair $\langle Bf, If \rangle$
- g is an n-player game.

The pair $\langle Bf, If \rangle$ corresponds to the set of brute and institutional facts. Brute facts Bf vary with the *physical* state of the environment, such as the size of the common pool. Institutional facts If are instead dependent on the *conventional* state, which are the state values asserted by agents of empowerment i.e. *institutionalised power* [17].

Designated roles of empowerment are one of the social constraints in the specification of a norm-governed system [18]. A role might refer to a member who is both an appropriator and provider of resources; or it might refer to an elected *head* of authority that can set values to If. For this project, all members appropriate and provision resources as *prosumers*, whilst the head role is simply assumed to be rational and unanimously approved across an institution's population. In other words, all operational-choice rules are modified at an institutional level of power, rather than by a specific agent of authority.

The game g can be any n-player game that an institution chooses to play. For this project, the focus will be on using the LPG′ game described in Section 2.3 as a model for analysing endogenous resource allocation in institutions.

Finally, we can define each agent (or 'particle') $a_t \in A_t$ participating in the system by the 6-tuple:

$$a_t = \langle R, E, H, swf, D \rangle_t$$

where:

- R is a set of quantities for resources allocation.
- E is a set of utility and satisfaction parameters.
- H is a set of physical properties.
- swf is the SpinWorld framework, formalised in Section 4.3.
- D is a decision function concerning compliance.

Every agent has a set of physical properties H that vary over the period of simulation, such as their velocity and 2D cell location. In an n-player game setting, the agents will also possess attributes R related to the resource distribution process. For the LPG′ game, this resource allocation set would include the amount of resources generated, needed, provisioned, demanded, allocated, and appropriated. A set of utility and satisfaction values E additionally describe the positive state of the agent with respect to achieving their individual goal in managing resources. This set may also include a threshold for which an agent will decide to leave an institution if the threshold is not met for a certain number of consecutive rounds.

Lastly, agents apply a decision function D on a round-by-round basis to formulate a strategy for maximising their E properties. D is a probabilistic function that agents refer to before determining whether or not to comply with the game rules in a given round. A driving force in this decision-making process is an agent's intrinsic propensity to cheat under subjection to retributive-based rules. Agents adapt their likelihood to cheat using the SpinWorld framework swf, which computes a factor to either reinforce or slacken the 'spin' on an agent's tendency for non-compliant behaviour.

4.2 Formal Specification of Institutional Rules

A formal specification of institutional rules situated in the LPG′ game will now be grounded on the design principles for self-governing the commons (Section 2.2.1). To meet the sustainable CPR requirements of this project, the multi-agent model is characterised to conform to Ostrom's system of retributive justice (Principles 4, 5, and 6). Additionally, each operational-choice rule of this system maintains congruence (Principle 2) with the prevailing state of ϵ, as well as with the agent community itself, M.

The resource allocation for an institution in the model follows a *Random* policy based on the argument that this method is no less 'unfair' than any other protocol (e.g. queue, priority, largest-first). This procedure entails a random mapping from a vector of demands by the members M to a pairwise vector of allocations, given by:

$$Random : \{d_a\}_{a \in M} \to \{r_a\}_{a \in M}$$

Another argument in favour of the notion of randomness in resource distribution is that there will always be the opportunity for non-compliance due to overpowering *individual* incentives to maximise utility, regardless of the chosen allocation rule [7]. Without the ability to reveal internal agent states in an open system i.e. lack of *full disclosure*, there is an obligation to monitor non-compliant behaviour. However, this raises the question of how to discern non-compliant actions from compliant ones such that a set of monitoring rules may be axiomatised.

Accordingly, the following operational-choices rules, ocr, are defined to regulate agent interactions and verify compliance or non-compliance with the CPR:

1. *Provision* rule — An agent a is required to provide the same amount as it generates:

$$ocr_{prov} : \forall a \in M, p_a = g_a$$

2. *Moderation* rule — An agent a is required to not demand more than it needs:

$$ocr_{mod} : \forall a \in M, d_a \leq q_a$$

3. *Appropriation* rule — An agent a is required to not appropriate more than it was allocated:

$$ocr_{approp} : \forall a \in M, r'_a \leq r_a$$

These rules were established in the first study of retributive justice as part of the computational justice programme [7], however the scope of their research fixated on *full disclosure* in open systems. Full disclosure is a feature of open systems dictating that an agent's private state cannot be audited. As a result, the system presented in the aforementioned study discarded the provision and moderation rules from the monitor function in correspondence with the fact that *internal* values g_a and q_a of agent a are unobservable.

For this project, non-compliant behaviour in open systems is the fundamental issue for which a system of retributive justice is to be developed. Consequently, the following monitoring functions encompass all three rules and insist on full disclosure at the loss of practicality in design:

$$monitor\_ocr_{prov} : \forall a \in M, (a, g_a, p_a) \rightarrow Boolean$$

$$monitor\_ocr_{mod} : \forall a \in M, (a, q_a, d_a) \rightarrow Boolean$$

$$monitor\_ocr_{approp} : \forall a \in M, (a, r_a, r'_a) \rightarrow Boolean$$

Taking the demand case as an example, its respective monitor function returns *true* if an agent a is caught 'cheating' by demanding more than it needs, and *false* otherwise.

In the event of mistaken or intentional 'cheating', the institution employs graduated sanctioning (Principle 5) to prevent future acts of recidivism. The logical axiomatisation of this rule is conditional on one of the monitoring rules being triggered:

$$sanction\_ocr : a \in M, a \times monitor\_ocr_*(\cdot) \rightarrow Boolean$$

A graduation in sanction level occurs whenever a member of the institution a is mapped to a *true* Boolean for any one of the operational-choice monitor rules. The condemned agent will then have their violation count incremented.

To apply a sanctioning mechanism, the institution distinguishes between three types of judgement: *no sanction*, *warning*, and *expulsion*. An institution decides to enforce one of these three judgements for a violation depending on its severity s with respect to harming the CPR. This severity metric is deduced quantitatively by how far the agent deviates from the intended quantity of provision, demand, or appropriation. By setting a lower bound and an upper bound on this severity scale at instantiation, an SOEI can decide how to sanction a member who has breached a set of rules.

Finally, a conflict-resolution mechanism is proposed to enable agents to appeal the result of their sanction (Principle 6). The operational-choice rule used to recognise an appeal event is resolved after sanctions have been appointed:

$$appeal\_ocr : \forall a \in M, a \times sanction\_ocr(\cdot) \rightarrow Boolean$$

If the agent has been assigned a sanction and is about to endure a penalty, then the condition for an appeal is triggered.

Congruence between the specified rules and the nature of the autonomous population (Principle 2) is maintained by ensuring that only institution members entitled to access the CPR can be affected by the rules, hence the iteration over M. Given the additional consideration of congruence with the prevailing environment, the rules discussed in this section are restricted to not disobey the physical constraints of the environment i.e. its sum of pool resources.

The computational design of SOEI adhering to the formal specifications provided in this section is discussed in Section 5.2, with executable specifications of institutional rules presented in Section 6.1.4.

4.3 SpinWorld Framework

SpinWorld is a framework used to express behaviour observation in agents that are subjected to rules of retribution. The framework *swf* is formally denoted by a 3-tuple:

$$swf = \langle o, \rho, \mu \rangle$$

where:

- o is a set of observation update functions.
- μ is a set of observation metrics.
- κ is a set of observation data structures.

Members observe different institutional choices to monitor and sanction other agents. These observations represent information events for the framework, whereby the functions o react to these events by updating stored data structures κ for a particular individual. Using these data structures to feed the framework, agents analyse metrics μ (propensity to cheat indicators) to estimate how appealing a non-compliant strategy would be. Combining the weighted and normalised values of these metrics produces the 'spin' factor, which is essentially a reinforcing or diminishing factor of an agent's propensity to cheat.

The *decision-to-cheat* function D of an entity is a probabilistic method for choosing what strategy to employ i.e. whether to comply or not with institutional rules. This function is entirely driven by the spin factor, for which a high spin will increase the likelihood of an agent cheating on their next action; and vice versa for a low value. Section 5.4 includes a more detailed explanation of how the spin factor and result of the decision-to-cheat function are obtained.

In this section, the proposed SpinWorld framework has been described in a generic structure that can be extended to incorporate various observation functions, data structures, and metrics for deriving an agent's propensity to cheat. For this project, the observations are concerned with estimating an institution's monitoring frequency and harshness in applying sanctions. Consequently, the propensity to cheat indicators of the framework are narrowed down to an agent's 'perception' of their risk, likelihood to be caught, and estimated benefit in rolling utility of pursuing a non-compliant strategy.

4.4 Modelling Social Networks

In order to design a dynamic system of institutions, a model is required to characterise the emergence of community structure. Gonzalez *et al.* [31][32] proposed a system of mobile agents to reproduce the evolution of social networks and their typical characteristics, such as clustering and community structure. Their findings proved that by representing mobile agents as nodes of networks, and their collisions as edges, it was possible to simulate dynamic networks. The target system includes equations from this research to formulate the physical nature of 'particle' motion and collision. By integrating an aspect of mobility into the MAS, electronic institutions can be dynamically formed throughout simulation runtime without the need for an instantiated set of I (initialise instead with $\emptyset$).

To model the properties of mobile agents and naturally encapsulate the emergence of institutional structure, a collision model is critical for its application in capturing particle dynamics. Social links associated with acquaintance are formed whenever two particles collide by intercepting one another, where interception refers to an overlap in the discrete cell location of two or more agents. Upon collision, agents have their trajectory and velocity altered in a random direction according to the degree of their nodes. Figure 7 demonstrates an example of the collision procedure.

Figure 7 displays two notable topological traits about the social networks. First and foremost, every network is modelled with a fully-connected topology. The rationale behind this design choice is that the dynamic nature of institutions can lead to unnecessary complexity in the redirection of links following node removal. Node removal occurs in the system whenever an agent is expelled from an institution, or whenever the agent opts out of an institution due to consecutive rounds of dissatisfaction. Secondly, there is a wrap-around effect interpreted for agents that surpass the coordinate space of the multi-agent environment. Deflections at the borders are removed to prevent the possibility of excessive clustering into small institutions and allow the entire grid to be traversed. In this way, an optimally sustainable institution containing a large population might arise.

The event-handling mechanisms and set of update equations for this collision model are presented in Section 5.5.1. A detailed description of the networking function utilised to resolve collision scenarios between agents is also supplied in Section 5.5.2.

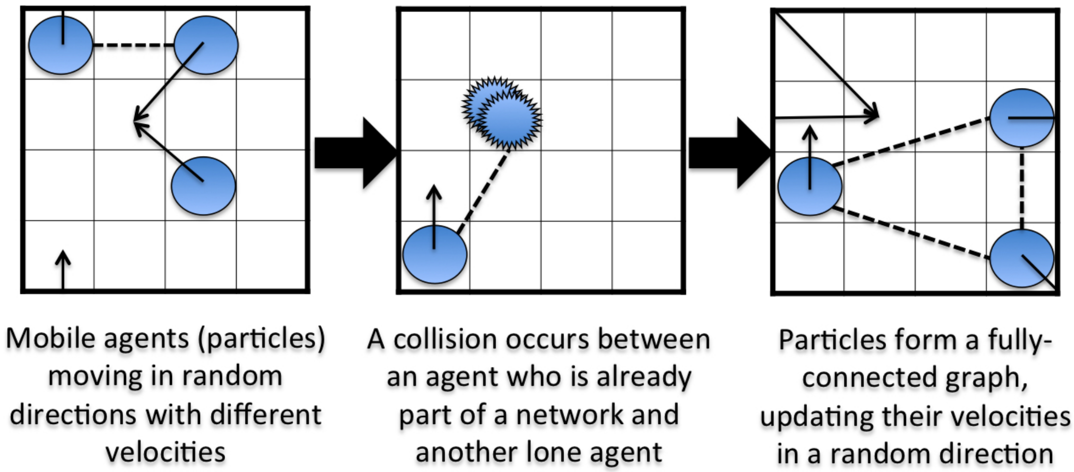


Figure 7: Process of building social networks (institutions) dynamically through collisions between particles, which are the basis behind forming connections (dashed lines).

5 System Design - Simulation Specifications

In this section, the previously presented formal model is mapped into an operational system that can be simulated in Presage2. A high-level system overview is first provided in accordance with the Presage2 simulation specifications illustrated in Figure 4. Each component of the MAS is then analysed in terms of its computational design and functionality.

5.1 System Overview

By adhering to the simulation creation process displayed in Figure 4, this project was able to design and implement an operational model of the target system in Presage2. A high-level overview of the final system design is provided in Figure 8.

In general, the fundamental components required to conduct the study of automated error detection and corrective-action were all developed successfully. Due to the lack of contribution to the project investigation, the physical network topology included in the final design holds no actual purpose in the implementation. The communication interface between agents will thus be excluded in the breakdown of each component presented throughout this section. Moreover, the SpinWorld framework realised by prosumer agents is also explained beyond what was formally defined in Section 4.3, such that it may be translated into a computational framework.

The current section is purely concerned with the translation of each formally specified component into a realisable computer design for simulation through the Presage2 platform. Section 6 of the report continues this analysis by providing implementations and executable specifications of each core component.

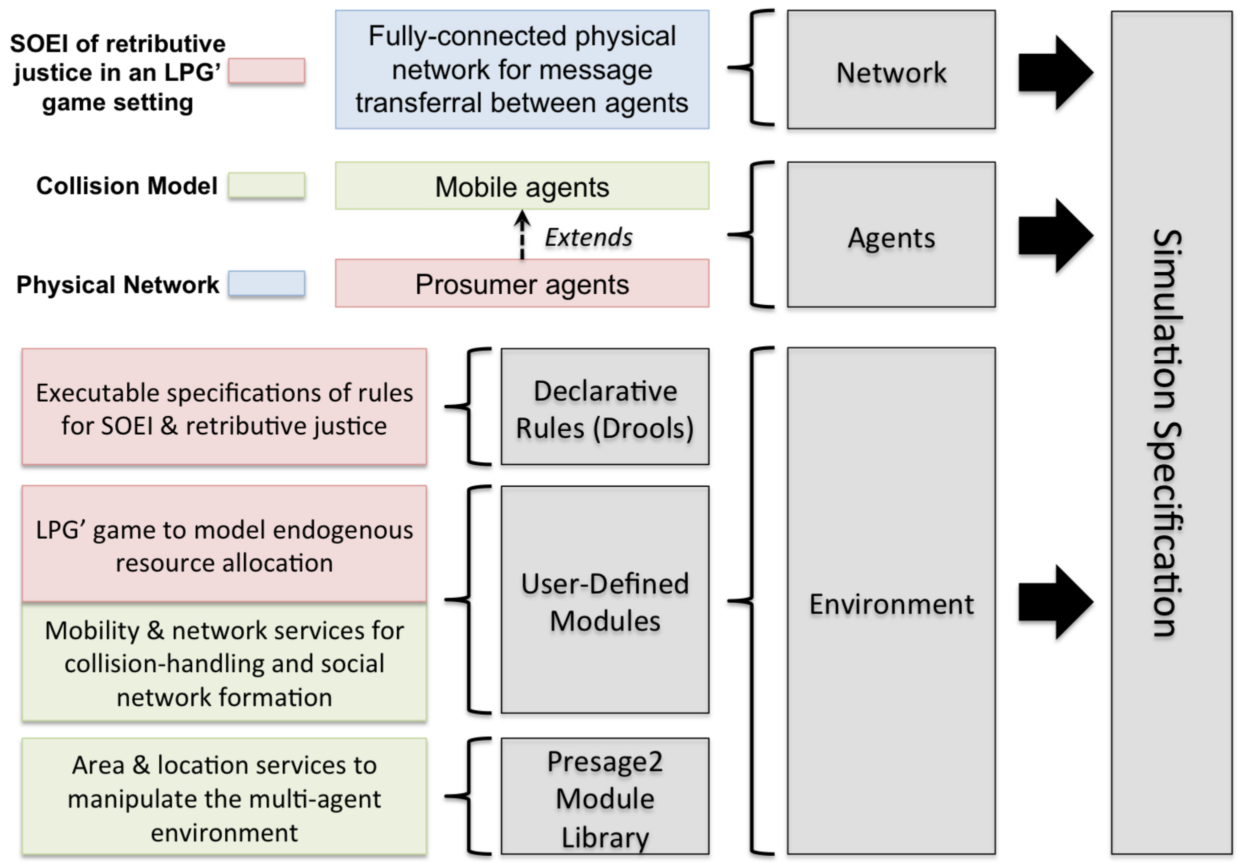


Figure 8: High-level system overview of the two-layered computational model simulated in Presage2.

5.2 Institutions

There are two key steps in the principled operationalisation of institutions within the system. First, the SOEI need to be translated into data structures that can be queried by the multi-agent environment. Once a computational object is obtained, the next step is to implement the logical axioms of institutions presented in Section 4.2 via the Drools rule engine. This section of the report is concerned with the encapsulation and initialisation of institutions, whilst Section 6 addresses the second step in principled operationalisation of SOEI.

A class diagram for an institution of the system that is classified as a social network is shown in Figure 9. Every institution in the MAS can be specified to use a different rule-set for error detection and correction upon creation. These rule-sets can be instantiated with the following DoF:

- Monitor: Choose to monitor or ignore non-compliance. If the institution is configured to have a monitoring function, set a cost and maximum frequency for its operation.
- Sanctioning: Choose to sanction violations or ignore them. If the institution employs a sanctioning function, set a value to determine what constitutes punishment or not.

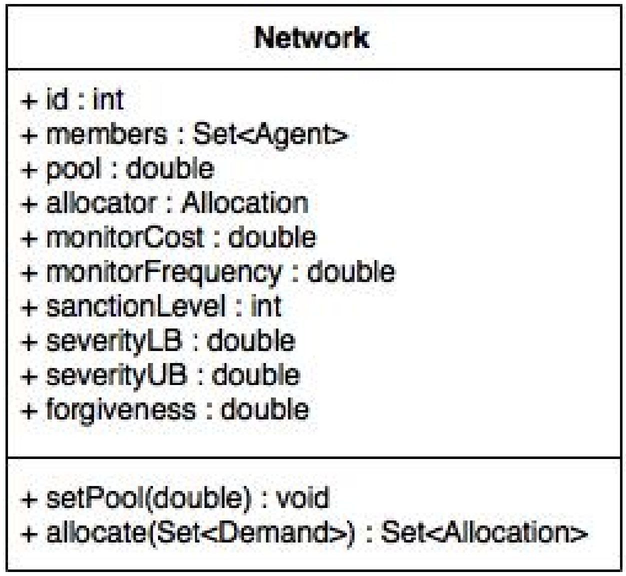


Figure 9: Network class diagram

.

- Graduated Sanctioning: Choose whether to sanction violations in graduated levels or not. If the institution applies a system of graduated sanctions, set a number of levels.
- Deterrence: Choose whether to apply a mechanism of deterrence in sanctioning or not. If the institution uses a deterrent judgement system, set bounds of leniency for punishing non-compliance.
- Conflict-Resolution: Choose whether to have a conflict resolution mechanism or not. If the institution has a dispute-resolution mechanism, set a forgiveness factor in dealing with sanction appeals.

Although SOEI can incorporate one or more of these DoF, there are certain obvious dependencies between them. For instance, an institution cannot enforce sanctions if there is no monitor to pick up on violations. Likewise, appeals are redundant if no sanctions took place. An SOEI that combines a system of monitoring, graduated sanctioning, and conflict-resolution abides by Ostrom's principles of retributive justice (Principles 4, 5, and 6) for enduring institutions.

Aside from the DoF that define which techniques are used to deal with error in an SOEI, another configurable parameter is the choice of resource allocation method. Excluding the allocator, none of the other parameters are mandatory in the instantiation of an institution. Consequently, institutions at the most basic layer of construction simply allocate resources and sustain congruence with the local environment. In the context of this project and its investigation, the allocator value has been fixed to *random* across all institutions for the reasons provided in Section 4.2. As a result, the only distinguishable factors between institutions of the system are the DoF listed above.

SOEI that enable monitoring have a certain fixed expense associated with their application, which is gathered from the CPR. This cost informs a decision to the institution about the frequency at which monitoring rules can afford to happen. A ceiling value for this frequency is also included in the parameterisation of this DoF, such that an institution does not waste all its pool resources unnecessarily on funding the monitor. Through self-determination, an SOEI uses collective-choice rules to adapt its maximum frequency level in accordance with the number of offences detected in any particular round of the LPG$'$ game. If no violations occur, the institution proportionately diminishes its ceiling limit by a fixed factor. In the opposite case, an institution will deem it necessary to increase the recurrence of monitoring if wrongdoing is brought to attention.

In a system of sanctions, a single lower bound is assigned to an institution as a threshold for acceptable behaviour. Anything above this value justifies that the non-compliant member is sanctioned with a 'warning' and is thus rendered incapable of demanding resources from the commons for a single round. This threshold value is compared to the severity of a violation, which is calculated as the deviation in an agent's provided, appropriated, or demanded quantity from the expected amount. Using the Provision rule, $p = g$ as an example, the severity s with respect to this rule can be computed as $s = g - p$. The s value is also correlated with the self-determination of a monitoring frequency by weighting non-compliance of greater severity with a higher reinforcement factor than that of lower significance.

An institution has the option of exploiting a retributive mechanism of punishment. By applying a system of graduated sanctions, there are a parameterised number of levels in retribution assigned to the system. Each stage in the graduation scale authorises a penalty that is proportional to an agent's wrongdoing. In other words, as a player is repeatedly sanctioned and escalates up the levels, their ability to demand from the CPR is withdrawn for the number of rounds corresponding to their assigned graduation. At the highest level of retribution, the player is expelled from the institution.

On the other hand, the deterrent mechanism extends the sanctioning DoF by additionally specifying an upper bound. As a penalty is entirely contingent on these bounds, an institution could adhere to a utilitarian justice system by setting bounds that benefit the current state of the CPR. For example, in a scenario where resource depletion is imminent, an institution could set very harsh bounds on the expulsion rule as a way to kill incentive for non-compliant behaviour. It is possible to construct an institutional object that resorts to both a retributive and utilitarian system of punishment during implementation.

Finally, a dispute-resolution device is activated if a forgiveness factor is set as an input parameter. This tool states whether an institution does or does not offer its members a chance to reverse the result of their conviction by appealing against the 'court'. Taking a probabilistic approach, the forgiveness metric is basically a range in which a randomly generated value dictates whether the institution allows a sanction to be reconsidered, or simply rejects the appeal.

Beyond encapsulating a self-organising system of retributive justice (or potentially deterrence), an SOEI is responsible for managing the state and access rights of the CPR.

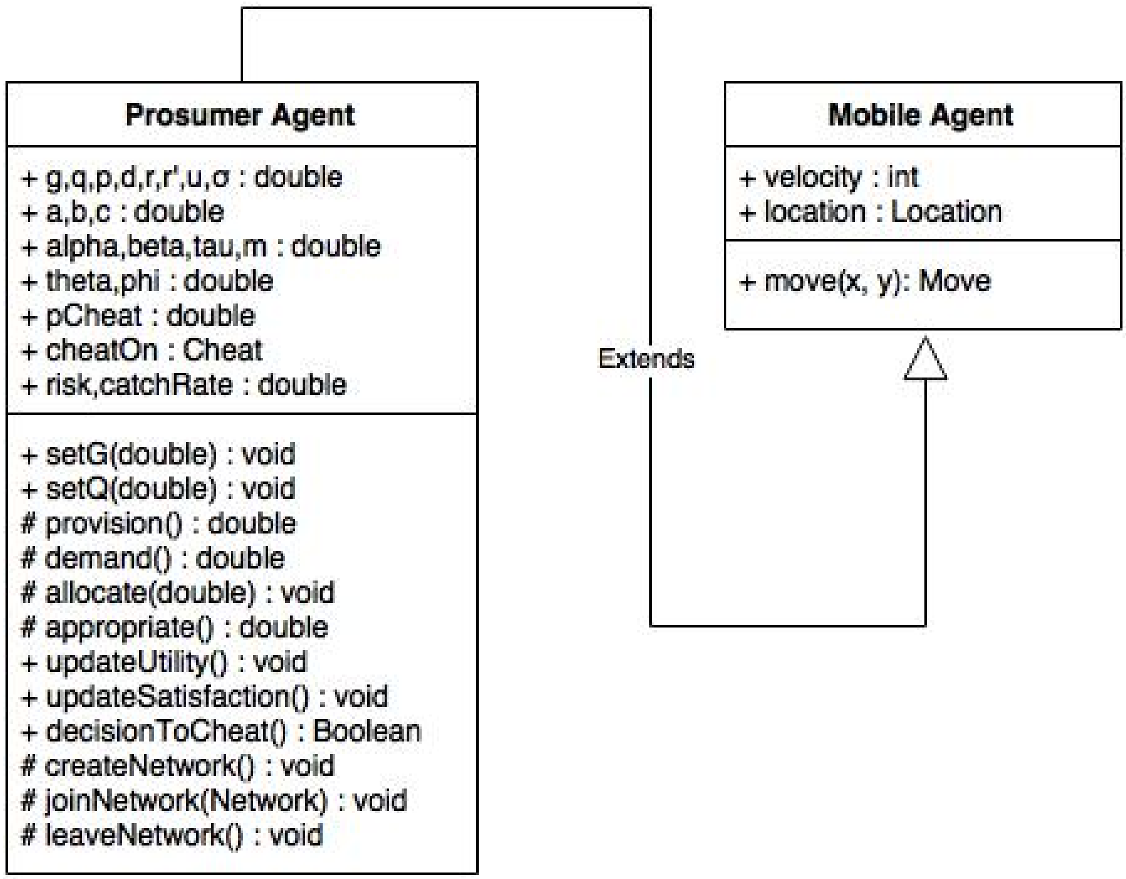


Figure 10: Mobile and Prosumer agent class diagrams.

Therefore, an institution of the MAS is also comprised of a data field that signifies the current quantity of the common pool, as well as a set of members who can provision, demand, or appropriate resources. Those permitted to act upon the state of the pool are entitled to receive allocations. Methods and data structures concerned with maintaining the state of the environment and regulating its agent community are instances of Ostrom's principle of congruence (Principle 2).

Protocols of retributive justice are analogous to appropriation and provision rules when it comes to ensuring a congruent state. An example of the direct relationship between Principles 2 and 5 is when an agent that is normally empowered to make a *valid* demand from the CPR has their entitlement rights temporarily revoked upon prosecution [18]. Conclusively, any implementations of Principles 4, 5, and 6 in SOEI are designed to always conform to the prevailing state of the multi-agent environment.

5.3 Agents

5.3.1 Mobile Agents

Mobile agents are designed in terms of the basic structure shown in Figure 10. At simulation start time, agents are initialised with fixed velocities at random 2D locations throughout the local environment. By behaving like particles in their motion, the agents act upon the environment at every time step to traverse around the grid of cells. Each agent has the ability to make a single transition in location per time step.

Two types are required for the environment state to perceive agent movement: Location and Move, which are both available to the in-built Presage2 libraries. In essence, Location is a 2D vector in the environment space that keeps track of an agent's relative position, whilst the Move action represents a change in this position based on two input coordinates that specify the target location. An action handler for Move actions validates that this target location is within the domain of the parameterised simulation grid space. During any movement operations, the mobility service keeps track of collisions and updates the physical states of particles using a set of formulae.

5.3.2 Prosumer Agents

The Prosumer agent class extends the Mobile agent class as defined in Figure 10. Each prosumer is an individual player of the LPG' game in which there are six actions: *provision*, *demand*, *appropriate*, *allocation*, *generate*, and *need*. These actions are tuples that contain the current round of the game and the quantity of the resource associated with the invoked action. In each time step, an agent privately spawns random quantities for their generate and need actions. When a prosumer is a member of an institution, they are permitted to participate in the game by deciding how much to demand and provision based on the generate and need values. An allocation is then bestowed to the agent by an institution of the environment. Upon receiving an allotted quantity, the prosumer finally decides on an amount to appropriate. Figure 11 [6] demonstrates this interplay between actors, networks, and the LPG' game.

Before agents pursue the actions of the LPG' game, they are first notified by the environment of any agents within the vicinity of their collision space. In the event of a collision with one or more agents, a networking function is referred to and dictates whether the collided particles should create a network, or whether one should join the other's institution. The environment service for network management is responsible for coordinating this process and updating prosumer membership to institutions.

After a prosumer has re-established who the current governing SOEI is, they determine whether to infringe the institutional rules or not in the next game round. Using the Spin-World framework, prosumers weight different observation metric values (e.g. *risk*, *catchRate*)

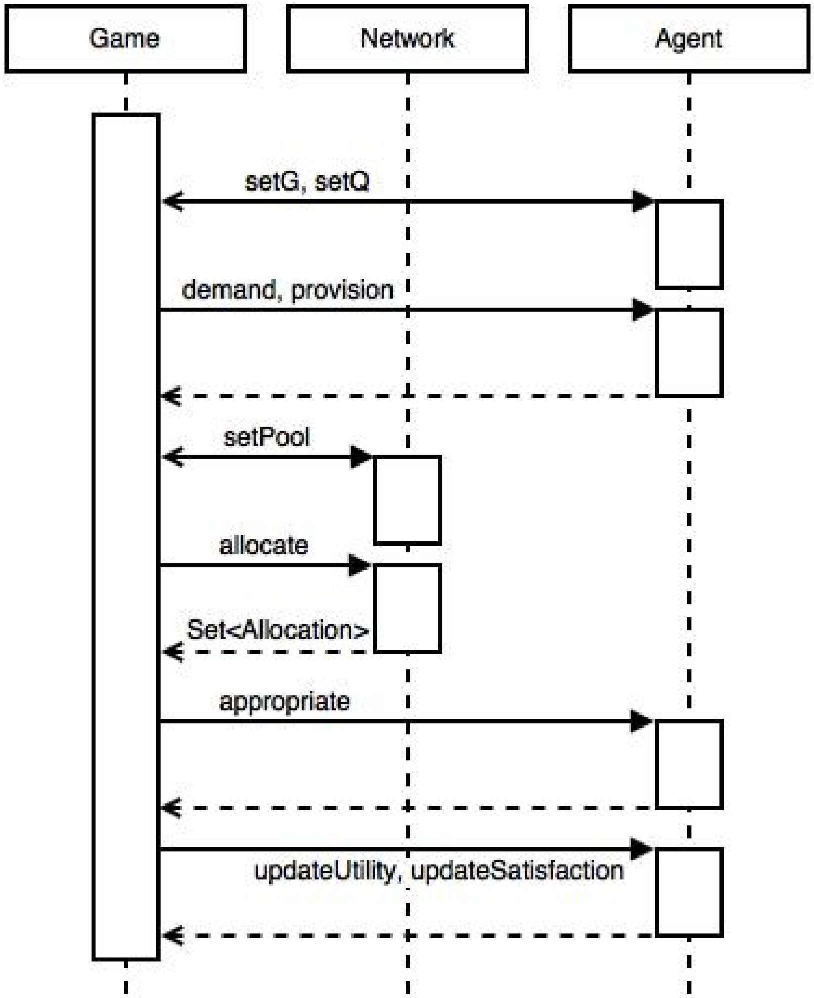


Figure 11: Sequence diagram of a single LPG′ game round [6].

and compute a spin factor with which to modify their propensity to cheat, $pCheat \in [0, 1]$. Once this property is evaluated, the agents probabilistically decide to non-comply if the decision to cheat function returns *true*. Compliant prosumers conform to the LPG′ game rules by demanding what they need, provisioning what they generate, and appropriating what they are allocated. On the other hand, non-compliant prosumers breach the rule corresponding to a pre-defined *cheatOn* parameter. This *Cheat* enumerable states whether to defy either the moderation, provision, or appropriation rules of the institution.

The agent data structures shown in Figure 10 include constant coefficients that are specified at compile time. As described in Section 2.3, the coefficients a, b, and c are used to calculate utility (Equation 1); the coefficients α and β compute satisfaction (Equation 2); and the satisfaction threshold is dependent on the coefficients τ and m. Coefficients θ and ϕ are also introduced in the SpinWorld framework for the purpose of manipulating the momentum of the spin factor.

5.4 SpinWorld

5.4.1 Observation-Based Behaviour

Prosumers are restricted from viewing the mechanisms of retributive justice at an institutional layer (e.g. the monitoring frequency), however they can make self-evaluated estimations of these variables based on observation. In the final system design, behavioural choices of agents in response to monitoring and sanctioning have been modelled using two elements: perceived likelihood of apprehension and perceived risk of punishment. To derive the estimated variables, the agents use stored data structures to keep a record of witnessed apprehensions and penalties. These data structures are fed into the SpinWorld framework as information events, which are updated on an observational basis.

With regard to the perceived chance of 'arrest', the agent attempts to estimate the monitoring frequency of an institution. To accomplish this, an agent tracks their own success at avoiding capture by non-complying with institutional rules and recording whether their action was monitored and sanctioned, or simply disregarded. The agent also takes into account any incidents where other members in the network underwent sanctioning, but they cannot observe whether these members have previously managed to bypass capture. Utilising a data structure that holds a set of Booleans, the agent computes their own perceived $catchRate \in [0, 1]$ at a given time step by taking the percentage of occurrences where the monitor triumphed in comparison to the number of successful evasions.

For the perceived risk indicator, an agent speculates what type of retribution would be enforced in the case of an unsuccessful violation. Similarly to the apprehension case, an agent tracks both the outcomes of their own sanctions and the assigned penalties of other members in the institution. A map of punishments is the data structure used to compute the perceived $risk \in [0, 1]$, where an agent compares the number of occasions when the sanction resulted in no judgement (as a consequence of an institutional decision, or an appeal) to the number of warnings and expulsions. Prosumers weight expulsions more heavily than warnings due to the assumption that agents perceive excommunication as a more significant form of retribution.

A final property that agents self-evaluate to compete against the former variables is the benefit acquired from the non-compliant strategy they applied in the past round of the game. This is simply computed as the change in rolling utility between the previous and current round.

5.4.2 Compliance Decision

In each round of the LPG$'$ game, an agent refers to the SpinWorld framework before deciding whether to pursue a compliant or non-compliant path. A key component in this compliance decision is an agent's inherent propensity to cheat, for which a driving force is the spin factor.

Accordingly, the spin factor $SF(a)$ is computed for an agent a at discrete time t as:

$$SF_a(t) = \theta \times mb_a(t) - \phi \times (mr_a(t) + mc_a(t)) \tag{3}$$

where the $\theta, \phi \in [0, 1]$ coefficients respectively control an agent's tendency towards mischievous and precautious behaviour; and the propensity to cheat indicators m (b for benefit, r for risk, and c for catch rate) have been normalised such that:

$$mb + mr + mc = 1$$

In a generic framework that uses an alternative and more expansive set of observation metrics, the equation for $SF(a)$ would be expressed as:

$$SF_a(t) = \sum_{i=1}^{n} w_i m_i \tag{4}$$

where n are the number of observation metrics, m are the spin indicators (reinforcing values are positive and slackening values are negative), and the weights w are again normalised to sum up to one.

Once the spin factor has been inferred by an agent, the following equation is used in the computer model to reinforce or diminish their propensity to cheat:

$$pCheat_a(t+1) = \begin{cases} pCheat_a(t) + SF_a(t) \times (1 - pCheat_a(t)), & \text{if } SF_a(t) > 0 \\ pCheat_a(t) + SF_a(t) \times pCheat_a(t), & \text{otherwise} \end{cases} \tag{5}$$

where the propensity to cheat for agent a at time t is reinforced by the spin factor if the benefit of a violation outweighs the risk and likelihood to be caught.

The updated *pCheat* field is finally tested by the agent in the decision-to-cheat function against the condition $pCheat < rand(0, 1)$, where $rand(0, 1)$ is a random float factor between 0 and 1. If this condition returns *true*, the agent chooses to breach the LPG′ game rules in the upcoming round, else if *false* they comply. Furthermore, in the scenario where compliance with rules has continually benefitted the agent in terms of utility, a fixed factor is used to decrease the propensity to cheat.

5.5 Environment

5.5.1 Mobility Service and Collision-Handler

The mobility module is a core abstraction layer for handling collisions and physical properties of the environment during simulation. This service describes the prevailing state of the environment and uses an event-detection mechanism to recognise collisions within the grid space. Another crucial design feature of the mobility service is the use of update equations construed by Gonzalez at al. [31] to modify the external state of agents involved in one or more collisions at a particular time step. Mobile agents query this module after executing their Move action in order to alter their physical dynamics for the following round of motion.

With regard to collision detection, an event-handler is triggered to check for collisions whenever a Mobile agent performs a Move. A collision is recognised as an event when two or more agents overlap in cell location i.e. have the same coordinates (assuming particles are of one dimensional radius). As a result, the mobility service checks for this condition whenever the operation of a particular Mobile agent has completed. If the condition is met, this environment module will store a structure that maps the relevant particle to a set of agents involved in the collision. This structure is the basis for Prosumer agents to be informed on whether they need to update their social network at the beginning of a game round.

For the purpose of simulating the dynamic behaviour of mobile particles in a MAS, a pair of update equations for the positions and velocities of constituent agents is required. The simulation setup for this project instantiates agents at time $t = 0$ in random locations throughout the environment. All agents initially have the same velocity modulus v_0 and are set to propagate in a random direction. In each time step t, position x_a of agent a is updated as:

$$x_a(t+1) = x_a(t) + v_a(t) \times (randInt(3) - 1) \tag{6}$$

This can be interpreted as an agent setting a target location $x_a(t+1)$ relative to their current location $x_a(t)$ based on the offset of a randomly generated direction $[-1, 0, 1]$ that is scaled by the agent's state of velocity $v_a(t)$.

After the occurrence of a collision, the velocity modulus v_0 of an agent a is altered in proportion to the degree k, which represents the number of social links connected to this agent at time interval t:

$$|v_a(t)| = v_0 + \bar{v}k_a(t) \tag{7}$$

where $\bar{v}$ is a constant arbitrary unit of velocity that can be parameterised as an influential factor for reinforcing the momentum of a collision.

5.5.2 Network Service

Given that institutions are dynamically created as social networks during implementation, the agents of the multi-agent environment are required to perform additional decision-making regarding the adaptation of networks. In particular, an agent faces the decision of whether to create, join, or leave a social network after colliding with another particle. This is termed the networking function and is a critical feature of the network service, which is a module concerned with managing the environmental state of membership and access rights in institutions.

In the scenario of a collision, an agent has four options of networking. Firstly, if both particles involved in the collision have no membership with an institution, then they unquestionably choose to create a network together. Secondly, if one of the particles is already associated with an institution, then the external agent decides to form a bond and join the institution. The third and forth cases of networking are more complex, whereby both particles are already affiliated with networks and must assess the two institutions out of preference.

Evaluating an institution for preference is conducted using a satisfaction-based method. This method entails that the agent compares their own governing institution against another by taking into account current dissatisfaction, as well as the satisfaction distributed across the other network. If the prospect of leaving their institution is appealing, then the agent requests membership to join the network. However, if the current state of the agent is satisfactory and the other institution is not especially enticing, then this agent ignores the option and simply updates their physical properties in accordance with the collision dynamics.

Despite the simplicity of this method, implementing its functionality was one of the most challenging aspects of the project due to the multi-threaded nature of the Presage2 simulator. Numerous concurrency issues had to be considered as a consequence of this parallelism, which were all resolved using Java synchronisation methods and statements.

5.5.3 Resource Allocation Game

A module for managing the rules of the LPG′ game is not explicitly required as the environment functionality for processing these rules is encapsulated using the Drools rule engine [6]. The ability to store the game state and institutional rules as part of the environment is one of the powerful benefits of this rule engine, which was mentioned in Section 3.2.

The overall procedure of Drools encapsulating an operational model of the LPG′ game and SOEI is as follows. Actions performed by agents of the MAS are generated by the Presage2 simulator and registered as facts within the state storage machine. These factual data structures then trigger declarative rules written in Drools, which provides a way of translating formal axioms of institutions and LPG′ game rules into a compatible representation for modification at simulation runtime. This concludes the basis behind designing protocols in a computationally executable format for Presage2 to trigger iteratively at every discrete time step. The following section will present the implementation of executable specifications using the Drools rule syntax.

6 Implementation

This section completes the principled operationalisation stage of the sociologically inspired computing methodology [34] by presenting the MAS implementation that was used as a testbed for this project's investigation. Each of the key units that comprise this testbed are developed as a continuation from the previous sections, in which these components were formally specified and analytically designed. Development processes will regularly refer to the prototyping Presage2 platform [6] and its role in enabling simulation of the multi-agent environment. The closing remarks address concerns of data storage and animation as a means to support the controlled experimentation phase described in Section 7.

6.1 Testbed

6.1.1 Overview

Testbed development took place in Presage2, an agent-oriented Java platform that offers users the ability to systematically model and simulate artificial social systems. Embedded in this simulator is the Drools rule engine, which is consistently exploited in this project to accommodate the norm-governed nature of the MAS model. These are the fundamental programming tools employed in the target system implementation. Full source code implementation for this project is available online[3] (see Appendix A for user guide on set up).

There are four major components associated with the testbed designed for experimentation. The first is the main simulation loop of Presage2, which constitutes a single discrete time step in operation. For this project, the control loop is primarily modelled according to the LPG′ game and its sequence of steps (Figure 11). The second component is the collection of agent implementations for interfacing with the simulator, which are based on the data structures provided in Section 5.3. Third is the translation of formalised institutional rules and the theory of retributive justice into a Drools executable specification. Finally, to complete the specification illustrated in Figure 8 for simulation, an initial set of parameters are required to instantiate the game state and operate the multi-agent environment. This last procedure is exemplified in Section 7 of the report.

6.1.2 Operationalisation of the Linear Public Good Game

The core simulation loop for the project testbed is detailed in Algorithm 1 using an adaptation of the LPG′ game narrative for modelling endogenous resource allocation. This algorithm shows the procedure for an institution in an iteration of the loop, or the equivalent of a round in game terms.

[3] `https://github.com/mazrk7/SpinWorld`

Algorithm 1 Main simulation loop: The LPG$'$ game.

1: $A \leftarrow$ set of n agents; $I \leftarrow \emptyset$;
2: $mf \leftarrow mFreq$; $mc \leftarrow mCost$; /* Monitor parameters */
3: $t \leftarrow 0$
4: **repeat**
5:     $m\_count \leftarrow 0$
6:     update institution membership
7:     **for** each agent $a \in A$ **do**
8:         $g_a \leftarrow a$.generate; $q_a \leftarrow a$.need;
9:         demand d_a; provision p_a;
10:     **end for**
11:     compute common pool $P = \sum_{a \in A} p_a$
12:     compute allocation $R : \{a | a \in A\} \rightarrow \mathbb{R}$ using *Random*
13:     **for** each agent $a \in shuffle(A)$ **do**
14:         **if** $m\_count < mf$ **and** $P > mc$ **then**
15:             $monitor(a)$
16:             $P = P - mc$
17:             $m\_count \leftarrow m\_count + 1$
18:         **end if**
19:     **end for**
20:     **for** each agent $a \in A$ **do**
21:         $r'_a \leftarrow a$.appropriate
22:         update u; update σ;
23:     **end for**
24:     impose sanctions
25:     manage appeals
26:     update mf
27:     $t \leftarrow t + 1$
28: **until** $t == T_{lim}$

A run of the testbed is initialised at time step $t = 0$ with a random agent population size n and an empty set of institutions I. The *mFreq* and *mCost* parameters are specified DoF for dynamic networks that perform monitoring functions. Similarly, a set of empirical metrics for sanctioning and conflict-resolution mechanisms are configured in the parameter specifications in order to impose variations in the principles applied for different institutions.

At the beginning of each round, the institution's members are updated by the collision model and networking function described in Section 5.5. Every agent within a network plays the LPG$'$ game by performing a sequence of actions with respect to the CPR. Once these actions are processed, the institution aggregates the provisions received to compute the resource pool and subsequently allocates a quantity from this pool amongst its internal agents (lines 11-12). A *Random* allocator is used such that each member is selected at random and allotted its demanded amount until either every demand is met or the pool is inevitably drained.

In a system of self-organised error detection and corrective action, institutions additionally perform monitoring, sanctioning, and conflict-resolution mechanisms over the typical LPG′ narrative. For the monitoring function, institutions have a certain fixed cost mc and a maximum frequency mf. The combination of these metrics determines how many agents can be monitored in a single round of the game given the pool resource constraints (lines 13-19). By using the *shuffle* method in the monitoring loop, agents are selected at *random* to be monitored until the CPR cannot cover the costs any longer. An extension to this project could investigate how a structured algorithm would prioritise monitoring agents who are more likely to commit a crime in an economy of scarcity.

At the end of the round, an institution detects whether there have been any rule violations e.g. an agent demanding more resources than it needs ($d_a > q_a$). If an offence was monitored, the network enforces a sanction scheme and manages any appeals made by agents feeling undeserving of their judgement (lines 24-25). For sanction schemes, the levels of punishment diverge according to the severity of the violation and number of offences previously committed. Finally, the institution self-determines a new monitoring frequency depending on whether any infringement of rules was detected (line 26). In the scenario where a number of consecutive rounds consisted of fully compliant behaviour, the monitoring frequency is diminished. The overall testbed procedure is repeated until a number of rounds T_{lim} have been iterated over.

6.1.3 Agent Players

Section 5.3 described two types of agents: Mobile and Prosumer. Both classes implement an in-built Presage2 interface that is designed for participants in the simulation to interact with the state of the environment at each discrete time step. Provided with this layer of abstraction, agents can keep track of the testbed state on a round-by-round basis and call methods in turn with the simulation loop. In this section, the core behaviour of Prosumer players is explained in terms of how these agents interface with the control loop shown in Algorithm 1.

Prosumer agents have three principal functions in their implementation, all of which play a crucial role in the narrative of the main simulation loop. First and foremost, dynamically forming networks are one of the key design features in the system. Therefore, the mobility and collision-based qualities of Prosumer agents about the grid space are of paramount importance in the establishment of institutions. Secondly, if an agent is a network member and is thus entitled to play the LPG′ game, then their contribution of resources to the CPR is an essential influence on the sustainability of the collective. However, the final function employed by these players in the game is to devise a strategy that maximises their individual goals at the expense of contradicting with the common goal. Prosumers refer to the SpinWorld framework in order to compose this strategy.

Algorithm 2 outlines the functionality of a Prosumer agent for a particular round of the game. At the beginning of a round, the Prosumer first moves in a random direction to a

Algorithm 2 Prosumer agent operation in a single time step.

1: $C \leftarrow$ set of c agents collided with
2: $E \leftarrow$ sequence of observation events
3: *move* in random direction
4: **for** each agent $c \in C$ **do**
5:     $networking(c)$
6: **end for**
7: update velocity
8: generate $g \leftarrow rand(0, 1)$
9: need $q \leftarrow rand(0, 1)$
10: **for** each observation event $e \in E$ **do**
11:     update observation data structure κ_o
12:     compute propensity to cheat indicators m
13: **end for**
14: compute SF
15: update $pCheat$
16: $cheat \leftarrow decision\_to\_cheat(pCheat)$
17: **if** $cheat$ **then**
18:     $cheatOn$ either d, q, or r'
19: **else**
20:     comply with d, q, and r'
21: **end if**
22: update u; update σ;

target location using Equation 6. After transitioning to this location, the agent is informed by the collision-handler of any overlapping agents in the new grid cell. If any collisions took place, the agent applies a *networking* function in collaboration with the collided particle, which updates their social connections as detailed in Section 5.5.2. The agent's velocity is then modified with respect to its latest network degree using Equation 7. Careful consideration for concurrency issues during this event-recognition step was taken into account via synchronisation idioms.

Given the latest physical properties and network links, agents subsequently participate in the LPG′ game according to the steps listed in Section 2.3. Before an agent provisions, demands, or appropriates resources, they use the SpinWorld framework to compute metric values in response to triggered observation events regarding sanctioning and monitoring (lines 10-13). An agent then adapts its 'spin' in accordance with the latest information received from the propensity to cheat indicators (Equation 4). The *pCheat* property incorporates the updated spin factor into Equation 5 and the probabilistic *decision_to_cheat* function is called on to determine the agent's strategy for the upcoming round (lines 15-16).

An agent's strategy for non-compliance can involve breaking the rules of demand, provision, or appropriation, as classified by the *cheatOn* field. If an agent chooses to comply with the LPG′ game rules, then it demands what it needs ($d = q$), provisions what it generates

($p = g$), and appropriates what it is allocated ($r' = r$). For the non-compliant strategy selected by *cheatOn*, the agent will perform the following actions:

- $d = q + rand(0, 1) \times (1 - q)$ — Increases demand to be a proportionately random amount more than they actually need.
- $p = g \times rand(0, 1)$ — Reduces the provision quantity by a random amount proportionate to what they generate.
- $r' = r + rand(0, 1) \times (1 - r)$ — Increases the appropriation quantity by a random factor of what they are allocated.

At the end of each round, the agents reassess their utility and satisfaction values using Equations 1 & 2 respectively.

6.1.4 Executable Specification of Institutional Rules

For the target system, the Drools rule engine translates logical axioms of Ostrom's institutional principles to rules that can be recognised by the Presage2 simulator. The events modelled by this engine address both player actions and several institutional related events concerning resource distribution, as well as error detection and corrective-action.

Rules are specified in the Drools rule syntax using a 'when' clause that will execute the logic described in a 'then' clause. These rules are event-driven and triggered for every newly inserted, modified, or retracted 'fact' of the system. In this way, the rule engine stores a logical framework that describes the environmental state of the system. For this project, the factual-based description of the environment incorporates the current state of the game, its internal institutions, and the actions performed by agents throughout the simulation.

An example of an institution detecting a *Monitored* player cheating on their *Demand* action is demonstrated in Figure 12. This rule is signified by its name in the first line and will fire for every permutation of the combined set of facts specified in the 'when' clause. Colons are used to bind values on the right-hand side of the colon (*agent*, *network*) to variables on the left-hand side. These variables are denoted using a \$ convention ($\$ag$, $\$n$) and can be accessed by the 'then' clause to execute the specified operation.

The operation of the example in Figure 12 is as follows. The *Round* fact first affirms which time step the simulator is running in and the *MemberOf* fact validates membership rights of the agent performing a *Demand*. *Generated* is the fact that confirms whether an agent has cheated by demanding a quantity larger than they need ($quantity > \$q$), hence breaching the moderation rule. Finally, *Monitored* simply states that the agent was caught by the institution in the act of a violation with rules. Under these conditions, a *Violation* fact is declared by the institution with an associated severity of harm to the CPR, as well as the type of action that the agent chose to cheat on.

Consequently, by developing rules in the Drools syntax using a similar approach to the

```
rule "Detect cheat on demand"
     when
          Round($t : number)
          MemberOf($ag : agent, $n : network)
          Monitored($ag, $n, $t)
          Generate(t == $t, $g : g, $q : q, agent == $ag)
          $demand : Demand(agent == $ag, t == $t, quantity > $q)
    then
          Violation v = new Violation($ag, $n, $t);
          v.setSeverity($demand.getQuantity() - $q);
          v.setOn("demand");
end
```

Figure 12: Drools rule example for detecting an agent cheat on the moderation rule.

presented example, a set of institutional rules can be applied over any action triggered by the Presage2 simulator. Therefore, this framework offered the implementation of formal rules in an executable specification, which is the basis behind integrating Ostrom's design protocols of retributive justice (Principle 4, 5, and 6) and congruence (Principle 2) into an SOEI. In the source code of this project, a number of rules have been developed to apply monitoring, sanctioning, and conflict-resolution mechanisms, whilst pertaining to the local state of the environment and its autonomous population.

6.2 Local Data Storage

Simulation data has been stored locally for the purposes of this project. Presage2 offers a basic programming interface to persist data gathered during the simulation in a locally defined database, which was configured using a PostgreSQL implementation. Given this setup, transient agent and environmental properties were the primary data types sent to a PostgreSQL server for analysis. By logging these properties during experimentation with the testbed, the social system developed in this study was verified in terms of its performance for modelling mechanisms of punishment in SOEI.

6.3 Animation

Visualisation and animation play a key role in the controlled experimentation phase for engineering 'smart' social systems. This is especially true when modelling networks that are constructed dynamically and display volatile behaviour. Prior work with the Presage2 platform's predecessor addressed the benefits of simulating and animating intelligent systems with 'smarter' solutions using visualisers as plugins for the core simulator [42]. The results presented in this article were a key inspiration for the development of an interface to visualise the networking dynamics applied in this project.

Consequently, this section of the report simply presents a graphical interface that was used to obtain a richer view of the networking solution developed throughout this project. Figures 13, 14, and 15 demonstrate an interface of the solution developed during implementation of the networked social system. This user interface was programmed in Java using the JUNG framework[4] and can be operated to show a step-by-step evaluation of the network at every discrete time point in the simulation.

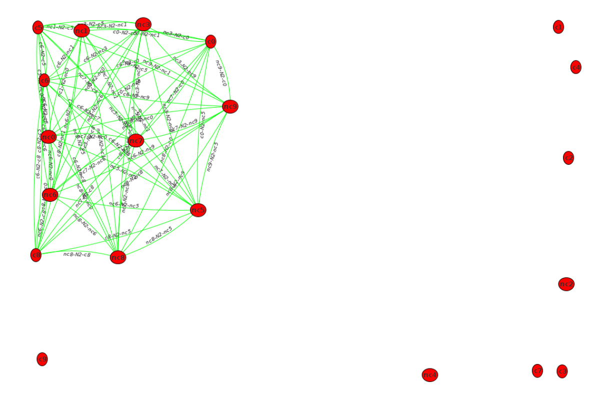


Figure 13: Social network at time t.

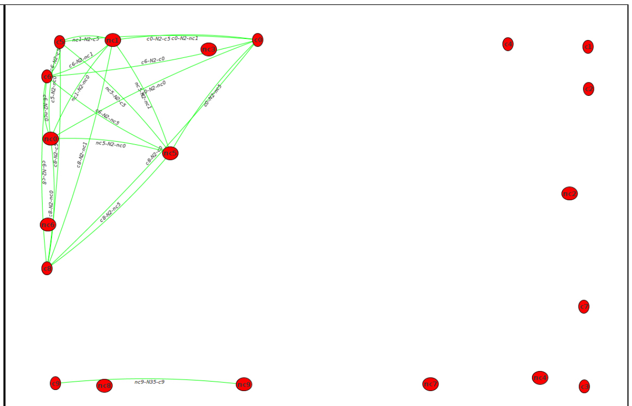


Figure 14: Social network at time $t + 1$.

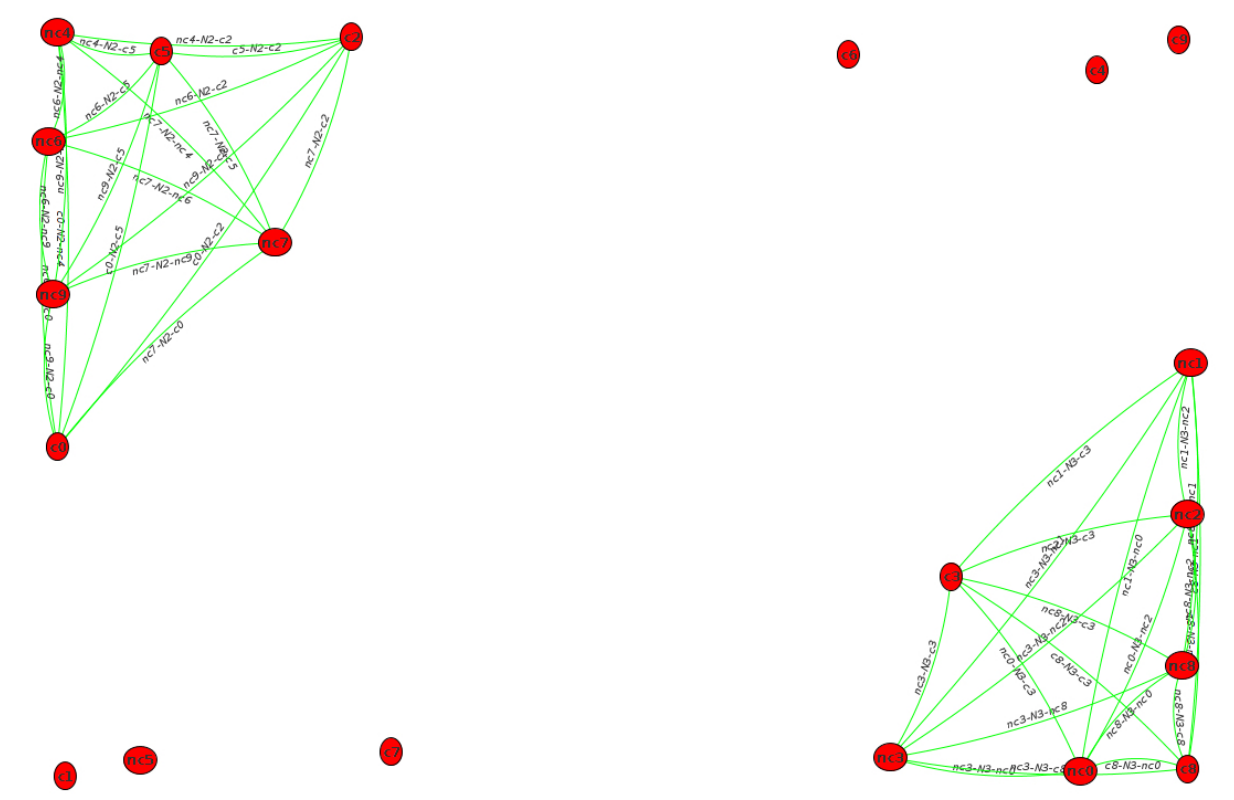


Figure 15: Two isolated social networks.

[4] `http://jung.sourceforge.net/`

7 Experimentation

Using the developed testbed, this section of the report describes a range of experiments that were conducted to assess the performance of an SOEI incorporating mechanisms of error detection and corrective-action. The experiments are first introduced in terms of their goals and setup. An analytical and graphical evaluation of the results obtained from each experiment is subsequently presented. Finally, an overall summary is provided at the end of this section to draw conclusions about the influence of self-organised retributive justice in a social system that aims to endure the resource allocation problem.

7.1 Experimental Setup

Experimental results were derived from simulations of the testbed discussed in Section 6.1. Each experiment seeks to explore how institutions sustainably manage the resource allocation problem under a specific parameterisation of their instances. A total of four experiments have been conducted with the following objectives:

(A) To investigate how variation in monitoring costs can affect the total utility of agents and longevity of institutional networks.
(B) To examine how different sanctioning schemes influence the sustainability of networks, as well as the agent population's utility and average propensity to cheat.
(C) To analyse the impact of different probabilistic conflict-resolution settings on the average longevity of networks and tendency for non-compliance, as per a system of retributive justice.
(D) To explore the comparative trade-offs between different institutional mechanisms of error detection and corrective-action.

In these experiments, the simulations were parameterised as follows (unless stated otherwise):

- $T_{lim} = 1000$: The total number of game rounds assigned per experiment is set to 1000.
- $\delta = 7$: All experiments were constrained to a 7×7 multi-agent environment space i.e. 49 discrete cell locations.
- $n = 40$ agents (20 compliant & 20 non-compliant): This was the standard population distribution selected for all experiments. In order to strictly investigate the effectiveness of different mechanisms of punishment for dealing with non-compliance in an SOEI, an evenly split population of compliant and non-compliant agents is proposed. Moreover, the population size is configured with respect to the 2D grid space such that agents are neither in an overcrowded environment, nor are they overly dispersed.
- $pCheat = 0.025$ (compliant) & $pCheat = 0.25$ (non-compliant): A compliant agent's likelihood for non-compliance is at a value indicating a scenario of accidental behaviour, whilst a non-compliant agent is parameterised to represent malicious actions.

- *cheatOn* $= Provision$: Every agent will cheat on their *provision* actions in the game. An analysis of different cheating strategies is outside the scope of this project.
- $a = 2, b = 1, c = 3$: The utility parameters for Equation 1 are set to maintain the constraint $c > a \geq b$, whereby agents have the incentive to defect but should display full compliance in order to meet optimal conditions for the common collective [20].
- $\alpha = 0.1, \beta = 0.1, \tau = 0.1, m = 3, \sigma_{i,C}(0) = 0.5$: The satisfaction parameters for Equation 2 are configured to represent moderate personality traits of agents. A threshold value of 0.1 is set to give a reasonable tolerance for dissatisfaction in a network. Furthermore, an agent i will choose to leave their network C if the threshold value is met for three consecutive rounds. Agents creating or joining a network are initialised to a mediocre state of satisfaction.
- $\theta = 0.1, \phi = 0.1$: The SpinWorld framework parameters used in Equation 4 are instantiated to model agents as well-balanced in their reasoning about the observations they make in an institution. That is to say, they will evenly weigh out the negative and positive aspects for deviating towards non-compliant behaviour.
- $mFreq = 0.4, mCost = 0.2$: Institutions are instantiated with a fixed cost of 0.2 for each monitoring action and the initial monitoring frequency is also set to 0.4. In the event of a violation, an institution will increase their monitoring frequency proportionately by a fixed factor of 0.1 as $mFreq = mFreq + 0.1 \times (1 - mFreq)$. If an institution has chosen to expel a member for whatever reasons, they will become more vigilant and instead increase their level of monitoring by a factor of 0.2 using the same update equation. However, if three rounds go by with fully compliant behaviour, then the institution will diminish their monitoring frequency by 0.1 as $mFreq = mFreq - 0.1 \times mFreq$.

Other sanctioning related parameters are introduced independently per experiment due to their variable instantiation values.

The following metrics are used to evaluate the results generated in different experiments over the course of a game:

- **Ut. Total/Sum**: Total agent utility generated, which can be split into compliant and non-compliant groups.
- **Ut. Avg**: Mean utility of all agents.
- **Ut. Std**: Standard deviation in utility across all agents.
- **Avg. Prop. to Cheat**: Mean propensity to cheat of all agents.
- **Longevity (%)**: Sustainability of a network expressed as a percentage of the total simulation lifetime.

Given a fair source of randomness associated with the simulations (agent processes for computing resources, deciding whether to cheat etc.), a controlled study should run a large number of repetitions for a single experiment initialised at different random seeds. As this project relied on a personal computer and local database, the number of simulation repeats that could be run for a single experiment and averaged over was limited to five due to memory constraints. However, executing five repetitions would impose a memory restriction on the number of game rounds down to approximately 250.

For the experiments analysed in this section, only a single repetition was run in order to acquire graphical plots of data for game durations of 1000 rounds. In addition to obtaining longer simulations, choosing one repetition also offered the ability to make finer grain analysis of the networks studied over the simulation, in particular with regard to isolating network properties for sustainability. Another argument against using more repetitions is that five alone would not have a significant influence on the removal of randomness. Appendix B includes an extended set of plots in addition to the ones displayed in this section. In this appendix, there are examples of simulations run for more repetitions, which illustrate a moderate smoothing effect on the time series plots.

Overall, a further extension to this project should include use of high-performance computing infrastructure to execute approximately 50 or 100 repetitions and thus obtain results that are less volatile to the random behaviour.

7.2 Experiment Results

7.2.1 Experiment A: Variations in Monitoring Costs

Experiment A analyses the total utility of agents and average longevity of networks for different values of *mCost*. Using the standard setup, this experiment was run for different $mCost \in \{0, .2, .4, .6, .8, 1.0\}$. Each monitoring cost is normalised against the average provision of an agent, such that an *mCost* of 1.0 would require the CPR to be completely depleted in order to monitor the entire agent population. However, the *mFreq* parameter can limit the extent of monitoring, which is initially the case for this experiment as *mFreq* is set to 0.4.

As the process of monitoring agents is effectively pointless unless some type of sanctioning mechanism is in place, a graduated sanction method L_5 has been applied across all simulations. Using this sanction scheme, agents are expelled from a network on their fifth offence, whilst at lower stages they have their rights to demand from a CPR revoked for a consecutive number of rounds that correspond to their currently assigned level. The decision upon institutions on whether or not to sanction a player using this scheme is at a fixed lower bound of 30%. In other words, whenever a non-compliant resource action deviates from the norm quantity by more than 30% of the intended amount, then the institution will enforce a sanction as a 'warning'.

The results for the aggregate utility of compliant and non-compliant groups of agents are shown in Figure 16. In this chart of bar triplets, the red bar corresponds to the compliant group, the blue to non-compliant, and green is the summation of both classifications. As monitoring costs increase up from *MC_000* (where there are no monitoring costs incurred), there is a downward trend for utility across all participants, with the exception of *MC_020*. Generally, this is to be expected as the system is strained by its endogenous nature and requirement to cover for any monitoring expenses directly from its own CPR. The difference

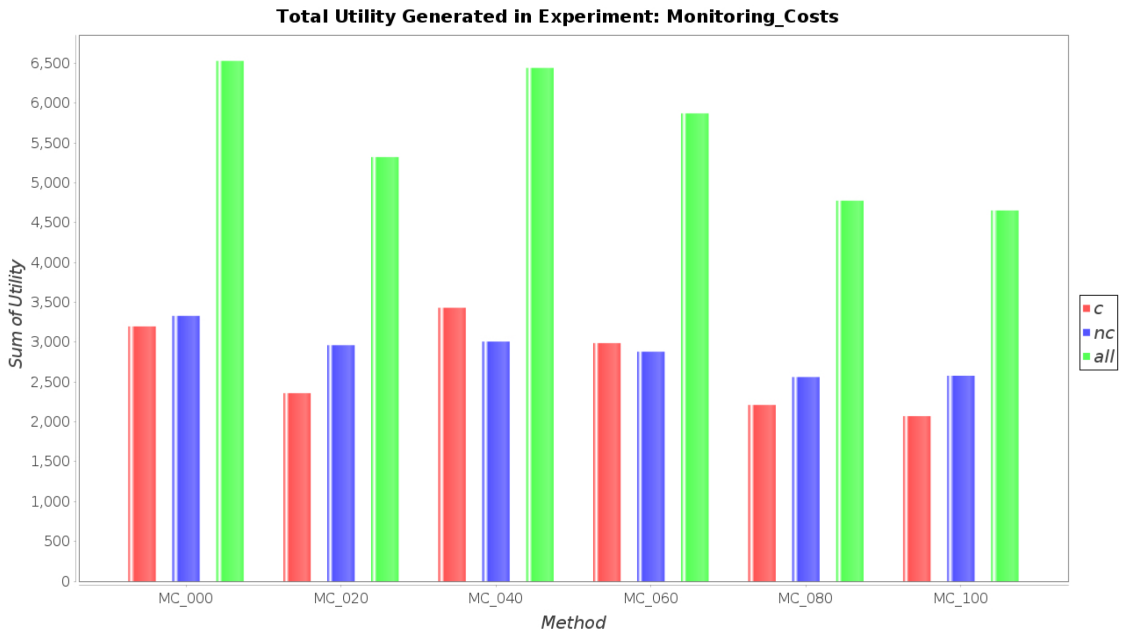


Figure 16: Experiment A – Total utility bar plot.

in utility points between the cheaper *MC_000* end of the chart and the expensive *MC_100* end is just under 2000, which indicates a significant difference relative to the total utilities.

Furthermore, the similarity between both groups of agent types in their utility illustrates that monitoring costs have little effect on the behavioural nature of the agent. In essence, higher costs will negatively affect all participating players, regardless of any heterogeneity.

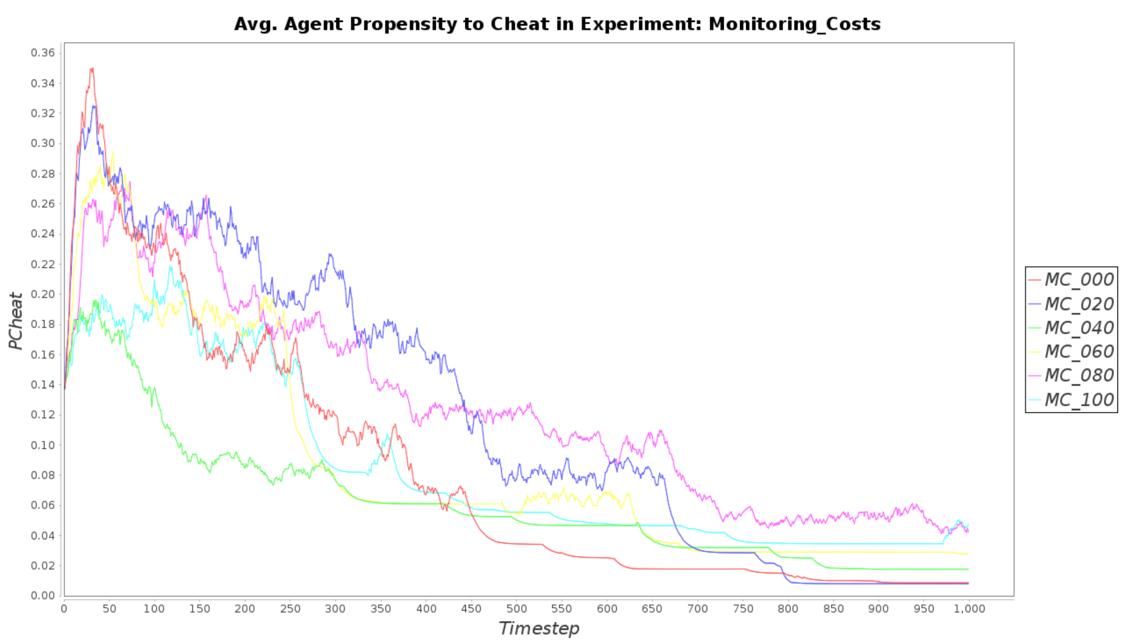


Figure 17: Experiment A – Time-series chart of average agent propensity to cheat.

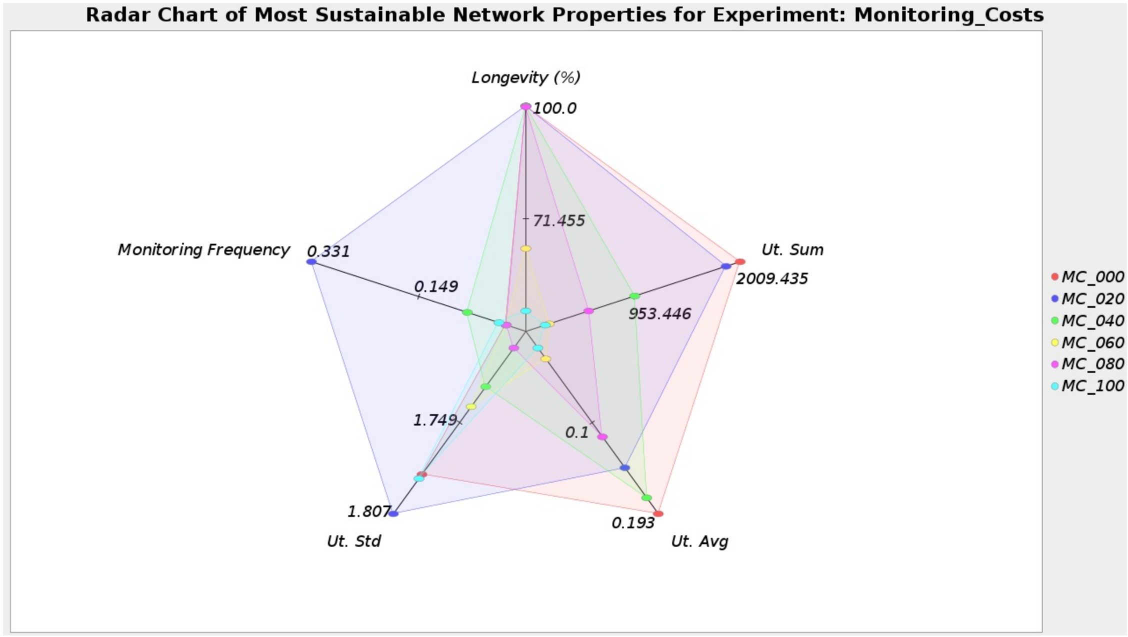


Figure 18: Experiment A – Radar plot of properties for the most sustainable networks.

This argument is reinforced by the similar trends forming in the average propensity to cheat time-series plot shown in Figure 17. Despite the fluctuations early on in the game, low propensity to cheat values begin to develop across all monitoring costs and appear to stabilise around 0.00-0.06 towards the end of the simulation. These low and stabilising lines are presumed to be a consequence of the effective sanctioning mechanism in place.

The last part of analysis for this experiment was to isolate the most sustainable network obtained in each *mCost* simulation and display its properties using a radar chart, as shown in Figure 18. For the cases where the expenses associated with monitoring were lower, *MC_000* and *MC_020*, the generated networks persisted for the entire duration of the game and had significantly higher utility totals in comparison to the more costly monitoring functions. Despite these benefits, the high standard deviations in utility of these networks relative to their mean suggests that there is a large spread between the utilities of each institution member. As a result, both the utility and longevity of the network could be driven by only a few lucky members in the institution, whilst the rest possible stray into negative states.

More crucial than the costs assigned to monitoring is the frequency at which it occurs. By examining the self-enduring institution generated during the *MC_020* simulation (Figure 18), an optimal monitoring frequency can be considered given the nature in which the institution self-determined and adapted this property. One design choice for a system of justice that could be extracted from this experiment would be to state that a monitoring level of approximately 0.33 is a reasonable parameter to configure. However, the autonomous nature of the population may cause this value to vary from simulation to simulation and thus disregards the ability to make such a claim about an empirically configurable frequency limit for monitoring.

7.2.2 Experiment B: Different Degrees of Sanctioning Schemes

Experiment B is concerned with how different degrees of sanctioning schemes affect the longevity of networks, as well as average agent utility and propensity to not comply with rules. The typical setup is used for this experiment, with monitoring frequency and costs fixed at 0.4 and 0.2 respectively. Two sanctioning methods were employed in this experiment: *L_x* and *S_lb_ub*.

For *L_x*, an agent is sanctioned in levels up to x, where levels are incremented by one for each offence detected. At each level, the sanctioned agent is prohibited from demanding in the game for the number of rounds that matches with their current violation count. In the case where an agent escalates to level x, the institution will choose to expel the non-compliant agent and ban them from re-entering this particular institution again. The simulations were parameterised with different stages $x \in \{0, 1, 2, 4, 6, 10\}$, in an attempt to model a system of graduated sanctions (Principle 5) through a punishment scheme that is proportionate to the number of recurring offences. The decision to sanction a player is set to a lower bound threshold of 30%.

On the other hand, the *S_lb_ub* mechanism explores the different bounds surrounding an institution's decision to sanction a player without any consideration for proportional punishment. For these simulations, lower bounds and upper bounds of tolerance for severity were set using different combinations of $lb \in \{0, .3, .5\}$ and $ub \in \{.3, .5, .8\}$. As mentioned earlier, the lower bounds represent the severity scale at which institutions determine whether or not to impose a demand withdrawal sanction for non-compliant behaviour. In this experiment setup, the removal of demand rights as a form of punishment is always fixed to be for the duration of a single game round, as opposed to the number of historical violations associated with a player. With regard to the upper bound, this sanction scheme attempts to model principles of deterrent justice by assigning very harsh punishments, such as expulsion, to all violations that exceed the upper constraint. That is to say, if the *ub* were set to 0.3, then any severity of crime exceeding 30% in scaled damage to the CPR will constitute expulsion.

Figure 19 shows the bar plot of total agent utility results for both sanctioning schemes. The bar triplets for system of graduated sanctions *L_x* display a clear downward trend in utility as the number of layers associated with the scheme increases. In other words, the expansiveness in a structure of graduated sanctions is indirectly correlated with the overall utility of constituent agents affected by these sanctions. This can be exemplified by comparing the two extreme cases of a 0-levelled and 10-levelled graduated sanctioning scheme. Agents are banned upon their first non-compliant action when governed under the 0-levelled sanction policy, yet they obtained a total utility of approximately 8000 units. The more stretched out sanction policy that includes 10 different stages in escalation has a far lower overall agent utility of about 3000 units. This substantial difference between the two systems can be attributed to the nature of prolonging demand withdrawal, which may have more influence on the utility of agents than a direct expulsion does.

A similar argument extends to the deterrent justice systems, where even a very strict

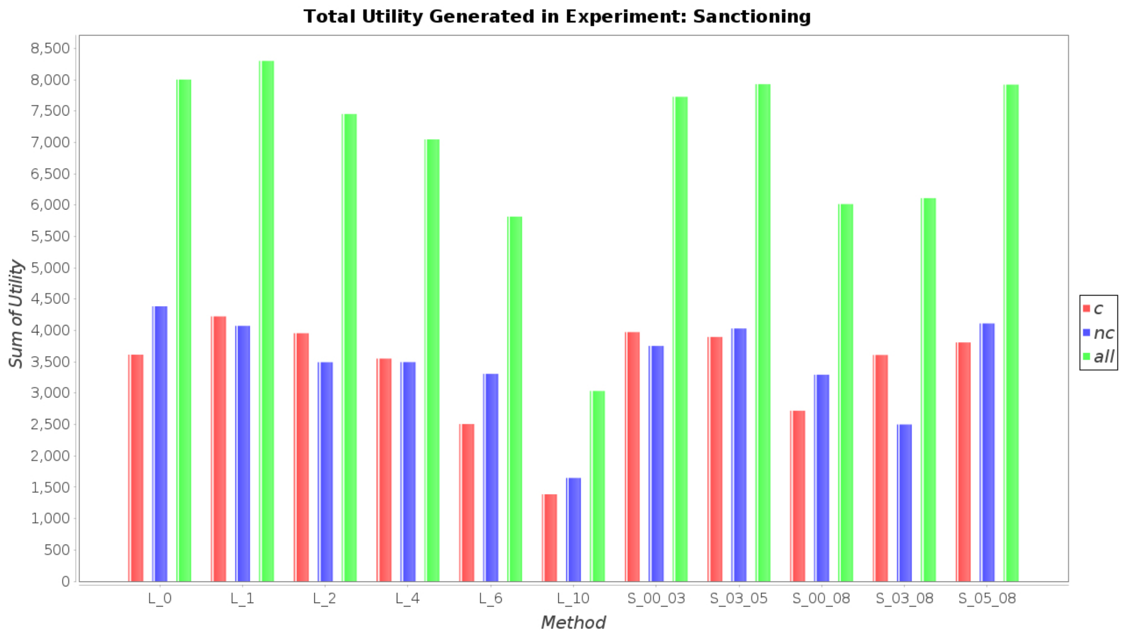


Figure 19: Experiment B – Total utility bar plot.

set of sanction bounds like the S_00_03 policy has significantly higher utility rates than any selection of bounds focused on accommodating a higher frequency of lesser punishments, such as demand restriction.

The average sustainability of networks for different sanctioning strategies is shown in Figure 20, which displays a nearly inverse effect to the utility plots. For both sanctioning

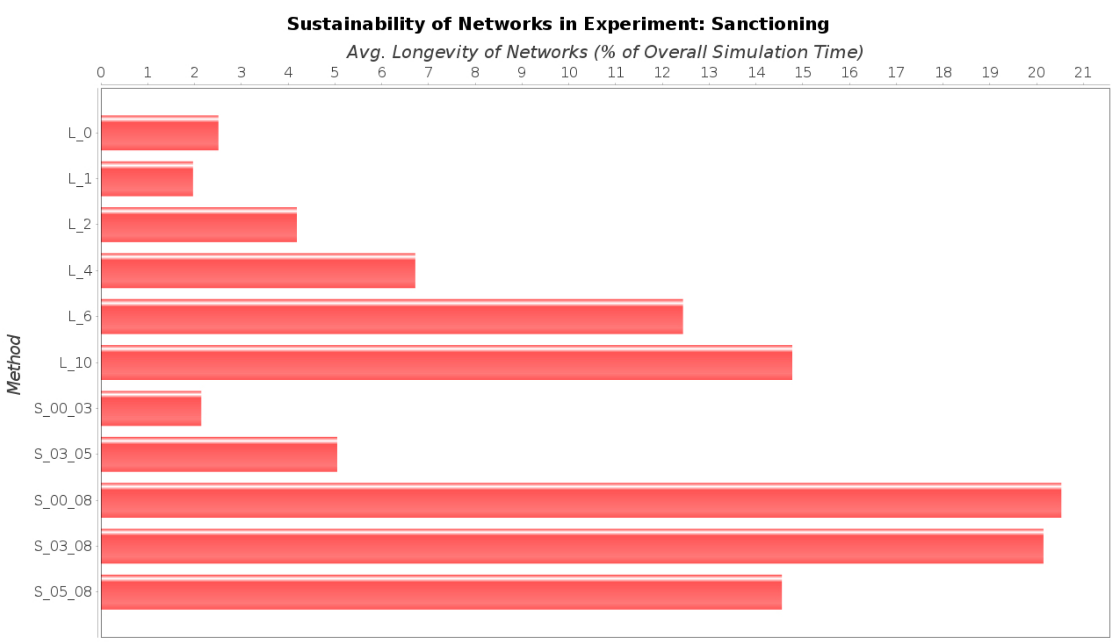


Figure 20: Experiment B – Average longevity of networks.

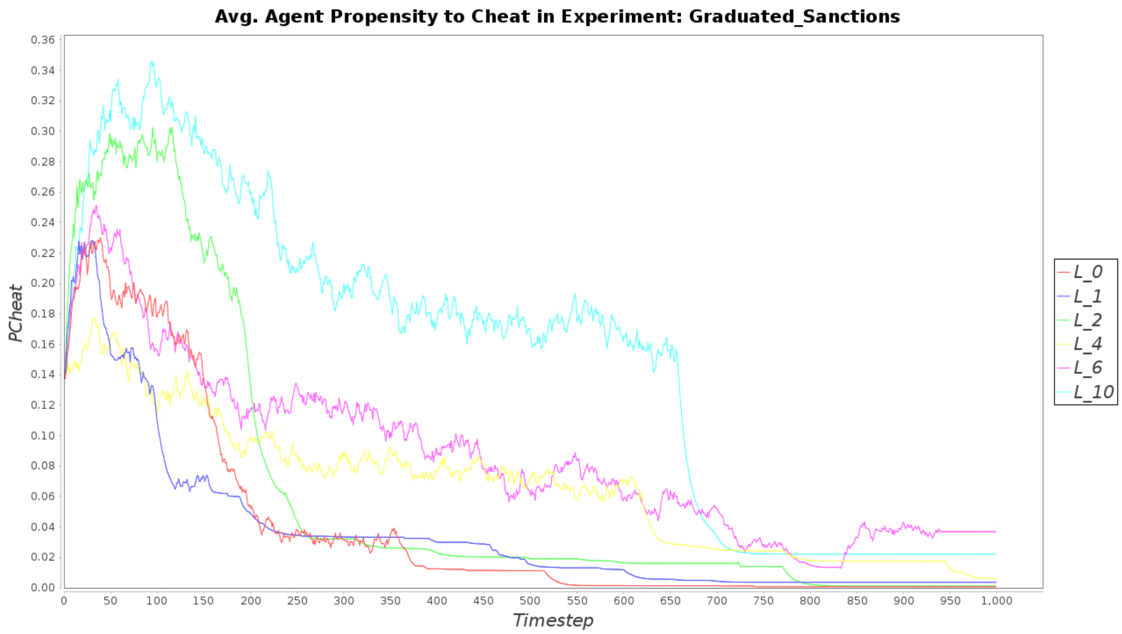


Figure 21: Experiment B – Average agent propensity to cheat in a system of graduated sanctions.

schemes, the influence of expulsion on network longevity suggests that there is a predictably strong correlation between one and the other. In particular, deterrent justice systems perform particularly well by setting the upper bound on expulsion to 0.8. This relatively high threshold for deciding to punish via expulsion enables agents to probabilistically avoid the likelihood of excommunication in a single accidental or intentional violation. At the other end of this severity scale, the institutions with low tolerance for malicious behaviour that can be detrimental to the CPR tend to obtain very low average longevities in the range of $1 - 5\%$ of total simulation time.

Both graphical bar charts for the different institutional sanction policies suggest that there is a strong trade-off between sustainability and maximising overall utility in the design of a punishment mechanism. The key aspect of this trade-off lies in the choice of punishment between assigning extended periods of demand withdrawal and invoking institutionalised power to expel players. Finding an adaptive balance in this decision-making process is comparable to certain human legal courts and their judgement to sentence criminals to either life imprisonment or the death penalty.

With regard to the state of compliance in the population, Figures 21 & 22 illustrate the respective trends in average agent propensity to cheat for a graduated sanctioning and deterrent justice system. In both time-series, the harsher schemes of sanctioning diminish the average tendency for non-compliant behaviour more rapidly than other policies.

For instance, the systems with fewer graduations and thus quicker mechanisms for expulsion L_0, L_1, L_2 stabilise the likelihood for non-compliance to very low values in the range 0.00-0.02, which indicate that agents will no longer cheat out of malicious behaviour but

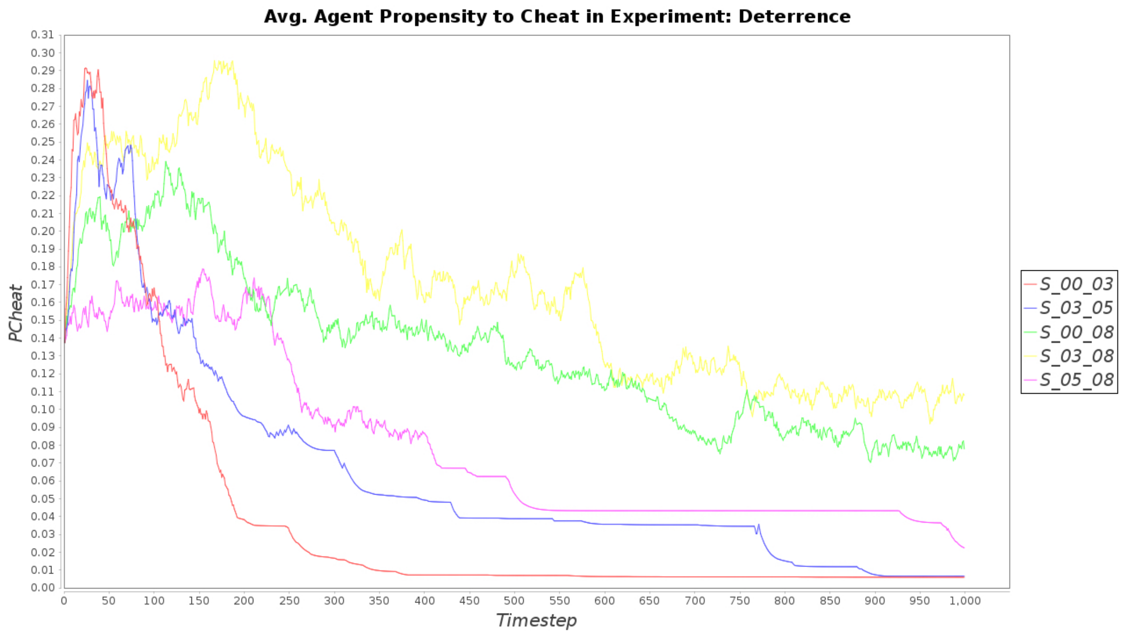


Figure 22: Experiment B – Average agent propensity to cheat under principles of deterrent justice.

rather mistaken resource management. The other higher levels of graduation do eventually tend towards lower average propensity to cheat, however it takes around 700 game rounds before this stabilisation develops.

Deterrent justice systems have similar patterns in the drop rates of agent propensities to cheat, but are more spread out in their effectiveness across the different mechanisms. Notably, the S_03_05 and S_03_08 severity scales for assigning punishment tend to flatten out around 0.08-0.11, which represents a range of values that would still lead to malicious and unpredictable acts of behaviour with respect to the game rules. Nevertheless, there is the possibility for further decline in a longer game.

7.2.3 Experiment C: Varying Forgiveness in a Conflict-Resolution Mechanism

Experiment C seeks to analyse the impact of different leniencies in a conflict-resolution mechanism on the performance of a system of retributive justice that encapsulates Ostrom's principles. Once again, the standard setup is used to operate this experiment. Configurations for monitoring are the same as in previous experiments, with *mFreq* initialised to 0.4 and *mCost* fixed at 0.2. In order to abide by a system of retributive justice, the graduated sanctioning mechanism is applied using L_5.

The conflict-resolution mechanism tested in this experiment concerns a probabilistic factor for determining whether an appeal should lead to corrective-action. This factor is represented as a forgiveness feature of institutions, whereby institutions with a high tend-

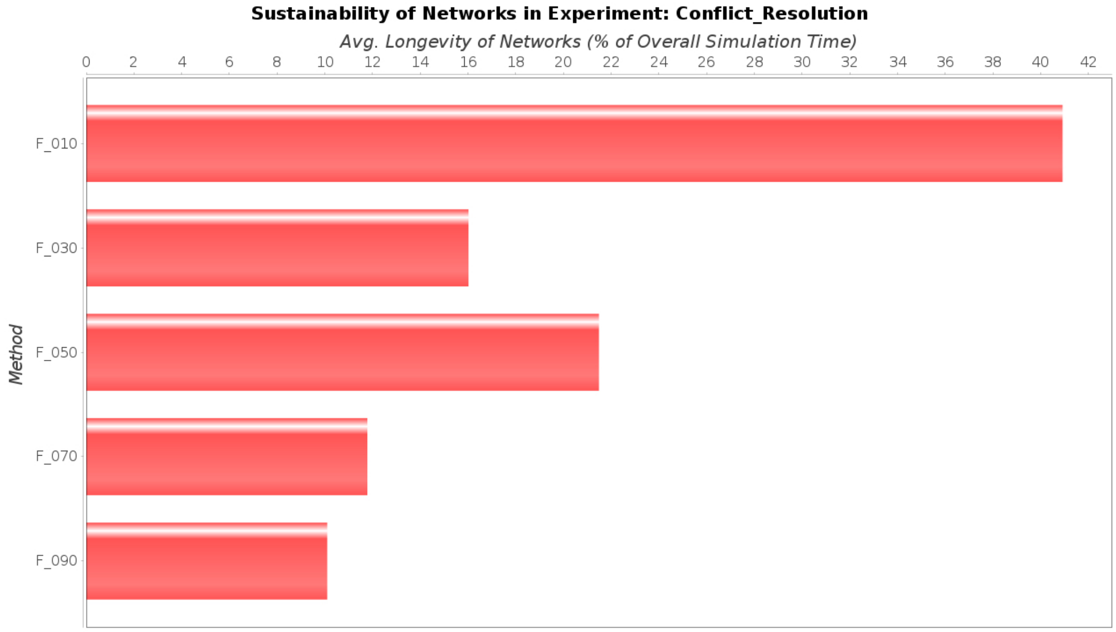


Figure 23: Experiment C – Average longevity of networks.

ency to forgive players that were previously sanctioned will allow the reversal of their assigned punishment. In this experiment, variations in the value of this factor are examined for their effects on sustainability in networks, as well as average agent properties of utility and propensity to cheat. The range of values used to represent the forgiveness factor F_y were $y \in \{.1, .3, .5, .7, .9\}$, where low y implies a very forgiving institution that enables formal changes to be made to a declared sanction on a regular basis, whilst a high valued network will reject the majority of appeals.

Figure 23 demonstrates the effects of different factors of forgiveness on longevity of networks. As would be expected, an institution that manages appeals and is likely to reverse sanctioned outcomes will generate long-lasting networks for around 40.5% of the simulation lifetime. However, a careless conflict-resolution mechanism with a low ability to carry out punishments will negate the influence of a well-designed sanction system and subsequently damage the state of the common collective. Figure 24 includes a radar chart of the best set of dynamically formed networks acquired across the different simulated conflict-resolution methods. In this plot, the low F_010 factor is revealed to have had less success in sustaining the resource allocation challenge than the institutions with forgiveness factors specified at F_030 and F_050. As a result, the argument for a probabilistically even method in handling appeals is demonstrated to be a superior form of dispute-resolution.

7.2.4 Experiment D: Comparative Trade-Offs in Design Principles

Experiment D is a wider evaluation of the comparative trade-offs between different design principles for error monitoring and corrective-action. The scope of this evaluation sought to

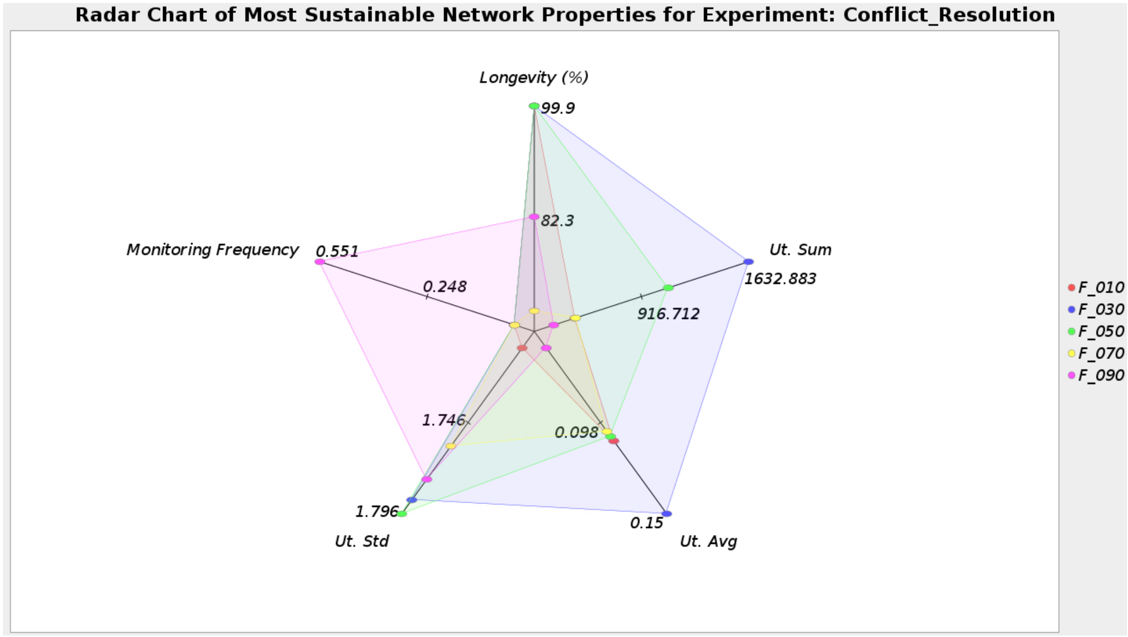


Figure 24: Experiment C – Radar plot of properties for the most sustainable networks.

compare the following five strategies in institutions:

- **Base**: The base case where no mechanisms of monitoring, sanctioning, or conflict-resolution are exploited.
- **Principles_4_5**: Abide by Ostrom's Principles 4 and 5 by incorporating functions to monitor and sanction (graduated scheme) agents.
- **Deterrence**: Apply simple sanctioning and monitoring schemes according to mechanisms of punishment that treat each individual case of negligence independently.
- **Principles_4_5_6**: Extends Principles 4 and 5 to include probabilistic dispute-resolution over appeals.
- **Combined**: Explore parameter space by mixing a system of graduated sanctions with a notion of deterrence.

All simulations were run using the same previously outlined setup. Monitoring was parameterised as $mFreq = 0.4$ and $mCost = 0.2$. Sanctioning parameters were as follows: L_5, $lb = 0.3$, and $ub = 0.8$. Lastly, the forgiveness factor configured for the last two simulation runs was F_070.

Figure 25 shows the total utility generated for each simulation run in this experiment. These results demonstrate that the *Base* case generates the lowest total utility over its simulation. Given that there are few comparative differences between the other four strategies, the results suggest that a minimal requirement to pursue the aim of maximising overall utility in an autonomous population is to adapt a monitoring and sanctioning mechanism into the set of institutional rules. As with previous experiments, there are also no distinguishing features of utility between compliant and non-compliant agents. This project argues that

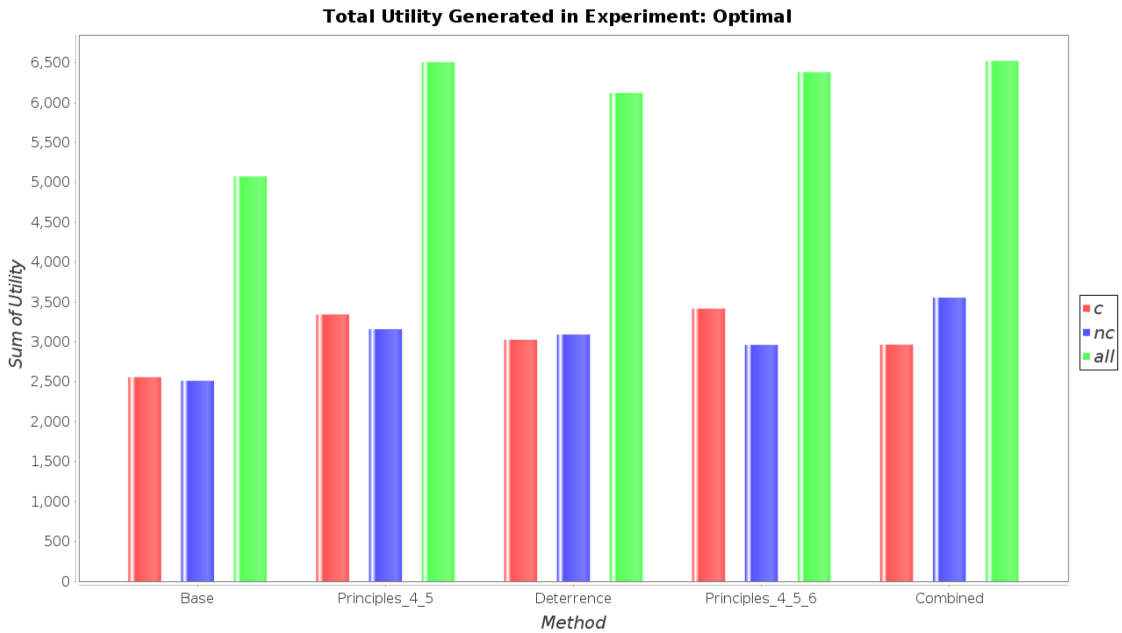


Figure 25: Experiment D – Total utility bar plot.

the SpinWorld framework's involvement in the system design leads to both types of agents being treated identically and therefore assists unanimously in adapting agents propensities to cheat, regardless of the initialisation.

Figure 26 adds more to the performance analysis of each strategy by displaying the bar chart for average longevities of networks. At the expense of utility, the *Base* strategy prevails

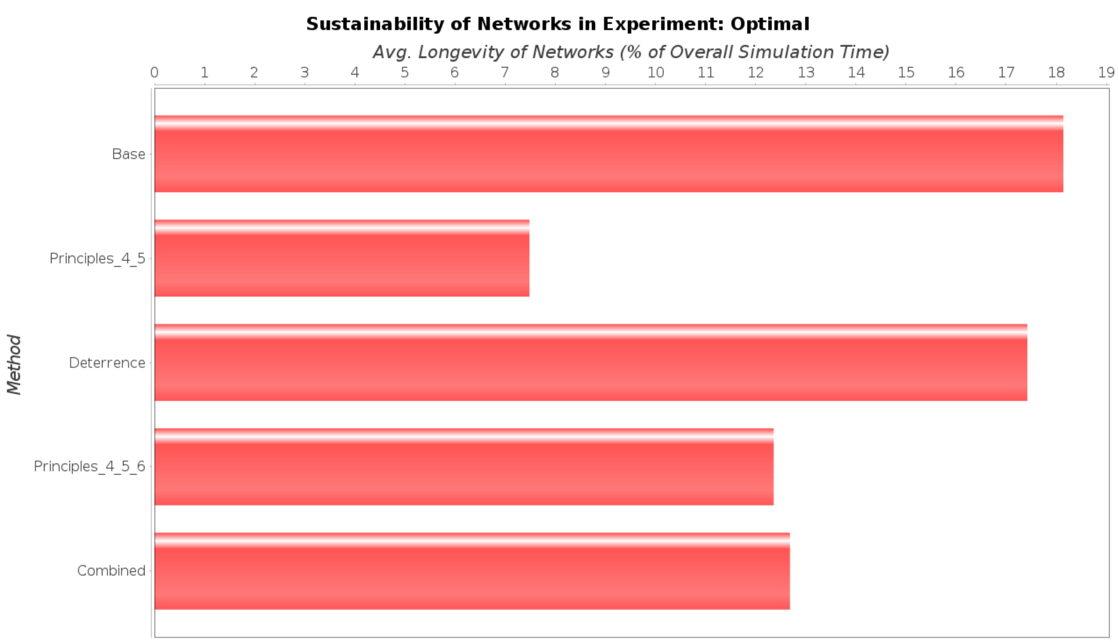


Figure 26: Experiment D – Average longevity of networks.

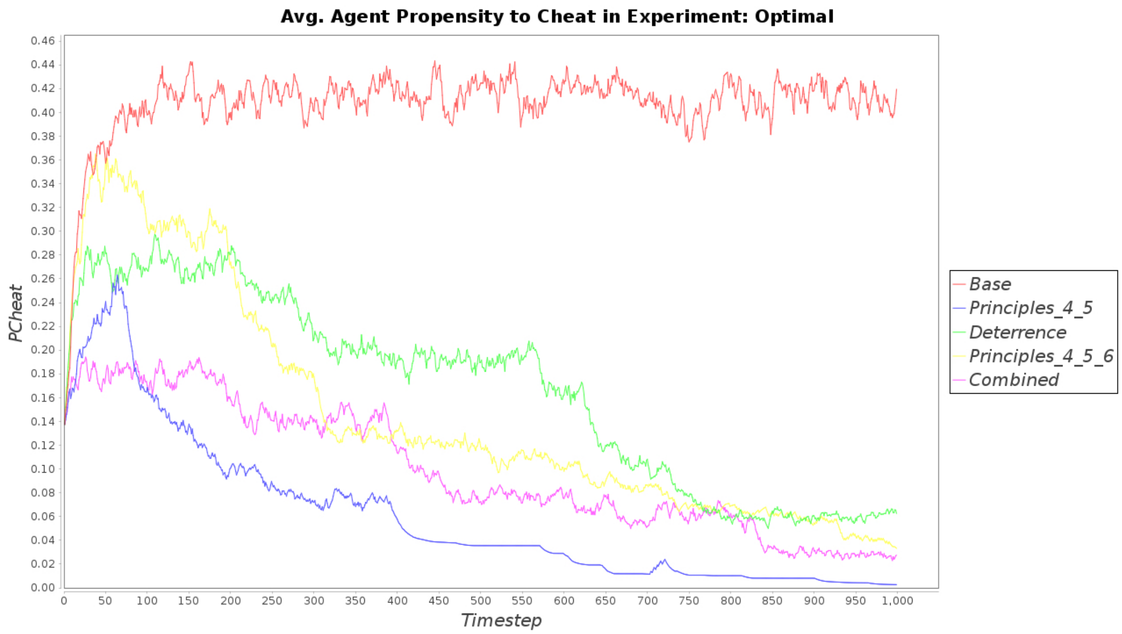


Figure 27: Experiment D – Average agent propensity to cheat under exposure to variations in institutional design principles.

in a longevity evaluation due to the lack of existing sanction policies. Principles of *Deterrence* in the design of a punishment system obtain a longevity average across networks that is most comparable to the *Base* case and without the drawback of lower utility. Another observation is that the addition of a conflict-resolution mechanism on top of Principles 4 and 5 leads to an increase in average sustainability of networks by about 5%. This result suggests that the simulation of the SOEI in this testbed experiment is a potential validation of Ostrom's set of principles for retributive justice.

Finally, this experiment closes with an analysis of the time-series charts shown in Figure 27 for average agent propensity to cheat. The most significant flaw of the *Base* case is debatably its inability to deal with non-compliant behaviour of autonomous agents in a resource allocation setting. Unlike other strategies, the constituent agents of *Base* institutions quickly adapt to their environment after using the SpinWorld framework to compute their spin in an observational state of no perceived risk or likelihood to be caught. After 100 rounds of the game, the agents have already learnt their system and fluctuate around an average propensity to cheat in the region of 0.39-0.44.

The other strategies display varying rates and degrees of drops in the average agent propensity to cheat. Although Ostrom's set of principles for retributive justice in an SOEI excelled during the evaluation of utility and longevity metrics, there is a less compelling argument for their design to tackle non-compliant behaviour. By losing the abstraction of conflict-resolution, Principles 4 and 5 are alone sufficient to completely kill off any individual incentive for non-compliance, with the likelihood to cheat value reduced down to almost zero.

7.3 Evaluation and Related Research

All four experiments presented in this section addressed the issue of sustainable decentralised resource management in a system prone to unpredictable agent behaviour. Initially, the aim of the project investigation was to find an adaptive balance between rules for error monitoring, sanctioning, and conflict-resolution in an SOEI. The results of these experiments conclude that there is a design choice to be made on whether the primary objective of the system should be to maximise total utility in the population or downgrade non-compliant behaviour in agents. Establishing a system of justice to target one of these objectives is further complicated by the critical dependency on the nature of the autonomous agent population.

Seeking an 'optimal' strategy to maximise longevity of governing institutions through the formal design of a system of justice across all experiments has introduced various trade-offs for increasing collective utility and diminishing propensity for non-compliance. During the investigation of variations in monitoring expenses, a sustainable network with generally desirable properties was generated using low-cost and limited frequency monitoring. Exploring a diverse set of sanction policies indicated that more 'drastic' forms of punishment (e.g. expulsion) should be enforced under deserving conditions, rather than allowing for excessive or prolonged use of less harsh mechanisms (e.g. temporary demand withdrawal). This could be contextualised into a typical debate about the deterrent effect of capital punishment [43]. Finally, different conflict-resolution mechanisms were examined for their effects on longevity of institutions and indicated that an equally probabilistic mechanism would maximise the formation of long-lasting networks with high levels of utility.

By implementing institutional rules to encapsulate design principles for self-governing the commons, a comparative evaluation of the trade-offs between different social systems of justice was also conducted for the purpose of acquiring an optimal model. Nonetheless, the results presented in this experiment only provided an empirical validation of Ostrom's principles of retributive justice as a sufficient and necessary condition in the self-regulation of non-compliant behaviour, which was put into perspective using a 'base' case system.

Modelling a social system to restore a 'normative' state in a population where agents adapt their behaviour selfishly in accordance with institutional rules proves to be an extremely problematic issue in practice. Evidence-based legislation is concerned with formulating technical implementations of law based on empirically derived data [44]. Unlike evidence-based medicine, whereby the best practice in gathering evidence to make decisions about the modern healthcare of patients is clinically accurate and explicitly defined [45], the same cannot be easily transferred to a legal study. Multiple controlled trials over a particular institution and its patterns of social behaviour may be used to derive a set of applicable rules to govern this institution, but they cannot be imposed upon another institution with a different type of social behaviour [44].

In the context of this project, the same behavioural issues that arise in evidence-based legislation apply to the procedure of simulating social systems in a controlled experimentation setting. As a result, the random characteristics of agent behaviour presupposes that the experiments conducted in this project could generate a different set of outcomes for each repeated study. Indeed, the discrepancies that exist across some of the plots included in this section (see Appendix B for more examples) demonstrate this behavioural dilemma in modelling a social system of justice.

Questioning why agents choose to cheat or not is an integral matter that should be further addressed in this study of non-compliant behaviour. A recent article in behavioural ethics demonstrated that 'winning' in a competition provoked players to be more dishonest in their subsequent actions, for reasons such as a sense of entitlement [46]. Likewise, 'losing' has been examined for its effects on victims and their sense of loss necessitating an act of selfishness or dishonesty in order to avoid further emotional or physical disadvantage [47]. By enhancing the SpinWorld framework to include notions of winning and losing in an LPG' game, a more accurate encapsulation of agent behaviour may be proposed.

Despite the influence of unpredictability in agent actions, Ostrom's principle of congruence with the system environment provides an essential means for modelling adaptive SOEI. This project has made a conscious effort to develop a social system that adheres to this design principle, however a more detailed approach for measuring the "fitness for purpose" [21] of an 'optimal' set of self-organising rules of justice is outside the scope of this investigation.

8 Limitations and Future Work

One of the fundamental limitations to this project's investigation is the lack of *role* empowerment included in the final system design. For this project, rules of retributive justice were imposed by an *institutionalised power* representing the whole community, as opposed to a self-determined authority. Every agent modelled in the MAS was thus portrayed as a 'prosumer', without any members undertaking the position of a 'monitor' and a 'head', or possible even both. Without any self-determination performed by the agents to elect a monitor and a head, the SOEI are fixed in their core algorithms for monitoring and sanctioning. The function to determine which agents to monitor is randomly organised, and the sanctioning decision on a case-by-case basis is set within certain bounds of severity that are not altered over simulation time.

Consequently, a further extension to this project would be to advance the incorporation of Ostrom's principle of self-determination (Principle 3) into the social system model. By developing agent functionality to elect and vote members into power, the specification of rules for retributive justice can be embellished with the complexity of individual behaviour. This would offer a broader perspective on the study of non-compliance in open systems. For example, experimentation could seek to explore the impact of different algorithms for electing a monitor or a head on the effectiveness of the rules for error recognition and corrective-action. Moreover, the drawback of unpredictable agent behaviour for evidence-based legislation can be used in this proposed context advantageously by self-adapting the rules concerning monitoring, sanctioning, and dispute-resolution.

Another major limitation associated with this project was the procedure for experimentation. As a consequence of using a local database to manage all store simulation data, the memory restrictions imposed on storing attributes of agents prevented coordination of large batches of simulation data. Furthermore, time constraints of this project and the development phase prevented re-configuration of the database schema to a smarter model without the occurrence of tables overfilling with millions of rows of persisted data.

An optimal option and suitable extension for future work would be to use high-performance computing infrastructure to run batches of simulations in parallel over a cluster of computers. A reinforcement learning algorithm could then also be considered to operate over this infrastructure for the purpose of obtaining an optimal parameter set to define an 'ideal' strategy for self-regulating and self-enduring institutions.

An additional direction for further research could involve extending the fields of justice covered in this investigation to also include concepts of restorative justice. Rehabilitation of offenders who have committed past violations is an area where the social system of justice developed in this project could choose to move towards. A proposal for encapsulating this type of justice might involve developing strategies in which agents who have breached institutional rules make amends for their wrongdoing by cooperating with other members of the institution to reintegrate back into the community.

Furthermore, the notion of forgiving sanction-worthy behaviour is briefly covered in this project and could provide another platform for future work. This may be particularly effective in modelling a more complex conflict-resolution mechanism using a reputation system that captures ideas of *trustworthiness* and *forgiveness* [48]. In this reputation system, an offender who has unintentionally defied resource allocation rules might be able to apologise for their action and thereby regain the trust of members in the institution. By building the idea of trust and forgiveness between agents, an authority of an institution could build a better decision-making process for revoking sanctions.

Finally, the SpinWorld framework that was introduced in this report could be further tested and enhanced to take into account other social aspects related to cheating, aside from behaviour observation and propagation indicators. As was briefly mentioned in the evaluation section (Section 7.3), a potential continuation with this framework could involve developing additional metrics and data structures for victory and loss in n-player games. This would enrich the decision-to-cheat function of agents and thus better-capture the social system explored in this project.

9 Summary and Conclusions

This project addressed the problem of expected component error in open MAS, which are required to collectively organise and distribute computing resources. With no centralised controller in open systems, internal agents self-determine the process for decentralised resource management by cooperating and synchronising with one another to make local decisions. Under these adaptive conditions, there is the possibility of sub-ideal component behaviour due to misunderstanding, necessity, or malice. To compensate for these errors, the open system performs corrective action as a means for restoring a 'normative' state. Therefore, mechanisms for monitoring, sanctioning, and conflict-resolution must be developed for the purpose of restoring compliance in the system. However, excessive monitoring expenses and harsh sanctions pose a viable threat to the survival of the collective.

To investigate the various trade-offs that these mechanisms impose on the sustainability and utility of the common collective, this project has designed and implemented a self-organising MAS with which to conduct this investigation. A formal model of the MAS has been specified using SOEI based on Ostrom's theory for self-governing the commons, which includes a set of principles to encapsulate a system of monitoring, graduated sanctioning, and dispute-resolution. Using these formal tools of analysis to express a social system of retributive justice, a computational model has been subsequently generated for simulation and operation using the Presage2 platform.

Using an LPG$'$ game to model resource allocation, a testbed has been developed to examine this model's performance through controlled experimentation. The results obtained from these experiments have indicated that an optimal strategy for defining a set of enduring retributive-based rules to regulate SOEI is difficult to derive from a system prone to unpredictable agent behaviour. However, by employing a concise and well-defined system of graduated sanctions with controlled monitoring costs and an overlaying conflict-resolution mechanism, the overall utility can be improved and the tendency for non-compliant behaviour can be significantly downgraded.

In conclusion, whilst an optimal set of rules for defining the automation of retributive justice in open systems remains a challenging prospect, this project has paved forth a means for further analysis of autonomous agent behaviour. Whilst establishing an ideal scheme of rules that conform to the environment and fit a theoretical model, implementation in practice represents the fundamental limitation of evidence-based legislation. As a result, any consideration of the trade-offs between mechanisms for error detection and corrective action in open institutions must equally maintain congruence with the adaptive nature of the agent population.

References

[1] Yuan Xue, Baochun Li, and Klara Nahrstedt. Optimal resource allocation in wireless ad hoc networks: A price-based approach. *IEEE Trans. Mob. Comput.*, 5(4):347–364, 2006. doi: 10.1109/TMC.2006.52. URL http://doi.ieeecomputersociety.org/10.1109/TMC.2006.52.

[2] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando. A game theoretic formulation of the service provisioning problem in cloud systems. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, pages 177–186. ACM, 2011. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963433. URL http://doi.acm.org/10.1145/1963405.1963433.

[3] B. HomChaudhuri and M. Kumar. Market based allocation of power in smart grid. In *Proceedings of the 2011 American Control Conference*, pages 3251–3256, June 2011. doi: 10.1109/ACC.2011.5991486.

[4] Jeremy V. Pitt, Dídac Busquets, and Régis Riveret. The pursuit of computational justice in open systems. *AI Soc.*, 30(3):359–378, 2015. doi: 10.1007/s00146-013-0531-6. URL http://dx.doi.org/10.1007/s00146-013-0531-6.

[5] OSTROM Elinor. Governing the commons: The evolution of institutions for collective action, 1990.

[6] Sam Macbeth, Dídac Busquets, and Jeremy V. Pitt. System modeling: Principled operationalization of social systems using presage2. In Daniele Gianni, Andrea D'Ambrogio, and Andreas Tolk, editors, *Modeling and Simulation-Based Systems Engineering Handbook*, pages 43–66. CRC Press, 2014. ISBN 978-1-4665-7145-7. URL http://www.crcnetbase.com/doi/abs/10.1201/b17902-4.

[7] Jeremy Pitt and Julia Schaumeier. Provision and appropriation of common-pool resources without full disclosure. In Iyad Rahwan, Wayne Wobcke, Sandip Sen, and Toshiharu Sugawara, editors, *PRIMA 2012: Principles and Practice of Multi-Agent Systems - 15th International Conference, Kuching, Sarawak, Malaysia, September 3-7, 2012. Proceedings*, volume 7455 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2012. ISBN 978-3-642-32728-5. doi: 10.1007/978-3-642-32729-2_14. URL http://dx.doi.org/10.1007/978-3-642-32729-2_14.

[8] Alexander Artikis. Dynamic specification of open agent systems. *J. Log. Comput.*, 22(6):1301–1334, 2012. doi: 10.1093/logcom/exr018. URL http://dx.doi.org/10.1093/logcom/exr018.

[9] Simon Gaechter. Conditional cooperation: Behavioral regularities from the lab and the field and their policy implications. Discussion Papers 2006-03, The Centre for Decision Research and Experimental Economics, School of Economics, University of Nottingham, April 2006. URL https://ideas.repec.org/p/cdx/dpaper/2006-03.html.

[10] Carl Hewitt. Open information systems semantics for distributed artificial intelligence. *Artif. Intell.*, 47(1-3):79–106, February 1991. ISSN 0004-3702. doi: 10.1016/0004-3702(91)90051-K. URL http://dx.doi.org/10.1016/0004-3702(91)90051-K.

[11] Mahadevan Venkatraman and Munindar P. Singh. Verifying compliance with commitment protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, 1999. doi: 10.1023/A:1010056221226. URL http://dx.doi.org/10.1023/A:1010056221226.

[12] Marek Sergot. *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, chapter Modelling Unreliable and Untrustworthy Agent Behaviour, pages 161–177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32370-9. doi: 10.1007/3-540-32370-8_11. URL http://dx.doi.org/10.1007/3-540-32370-8_11.

[13] M. A. Sirbu. Credits and debits on the internet. *IEEE Spectrum*, 34(2):23–29, Feb 1997. ISSN 0018-9235. doi: 10.1109/6.570823.

[14] Garrett Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968. ISSN 0036-8075. doi: 10.1126/science.162.3859.1243. URL http://science.sciencemag.org/content/162/3859/1243.

[15] Mancur Olson. The logic of collective action cambridge. *Mass.: Harvard*, 1971, 1965.

[16] Michael Cox, Gwen Arnold, S Villamayor Tomás, et al. A review of design principles for community-based natural resource management. *Ecology and Society*, 15(4):38, 2010.

[17] Andrew J. I. Jones and Marek J. Sergot. A formal characterisation of institutionalised power. *Logic Journal of the IGPL*, 4(3):427–443, 1996. doi: 10.1093/jigpal/4.3.427. URL http://dx.doi.org/10.1093/jigpal/4.3.427.

[18] Jeremy Pitt, Julia Schaumeier, and Alexander Artikis. The axiomatisation of socio-economic principles for self-organising systems. In *5th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2011, Ann Arbor, MI, USA, October 3-7, 2011*, pages 138–147. IEEE Computer Society, 2011. ISBN 978-1-4577-1614-0. doi: 10.1109/SASO.2011.25. URL http://dx.doi.org/10.1109/SASO.2011.25.

[19] Jeremy Pitt, Julia Schaumeier, Dídac Busquets, and Sam Macbeth. Self-organising common-pool resource allocation and canons of distributive justice. In *Sixth IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2012, Lyon, France, September 10-14, 2012*, pages 119–128. IEEE Computer Society, 2012. ISBN 978-1-4673-3126-5. doi: 10.1109/SASO.2012.31. URL http://dx.doi.org/10.1109/SASO.2012.31.

[20] Jeremy V. Pitt, Dídac Busquets, and Sam Macbeth. Distributive justice for self-organised common-pool resource management. *TAAS*, 9(3):14:1–14:39, 2014. doi: 10.1145/2629567. URL http://doi.acm.org/10.1145/2629567.

[21] Jeremy Pitt, Dídac Busquets, and Régis Riveret. Procedural justice and 'fitness for purpose' of self-organising electronic institutions. In Guido Boella, Edith Elkind, Bastin Tony Roy Savarimuthu, Frank Dignum, and Martin K. Purvis, editors, *PRIMA 2013: Principles and Practice of Multi-Agent Systems - 16th International Conference, Dunedin, New Zealand, December 1-6, 2013. Proceedings*, volume 8291 of *Lecture Notes in Computer Science*, pages 260–275. Springer, 2013. ISBN 978-3-642-44926-0. doi: 10.1007/978-3-642-44927-7_18. URL http://dx.doi.org/10.1007/978-3-642-44927-7_18.

[22] Kevin M. Carlsmith and John M. Darley. Psychological aspects of retributive justice. In M. P. Zanna, editor, *Advances in Experimental Social Psychology*, volume 40, pages 193–236. Elsevier, San Diego, CA, 2008. doi: http://dx.doi.org/10.1016/S0065-2601(07)00004-4. URL http://www.sciencedirect.com/science/article/pii/S0065260107000044.

[23] Immanuel Kant. The science of right (W. Hastie, trans.). In R. Hutchins, editor, *Great Books of the Western World*, volume 42, pages 397–446. T&T Clark, Edinburgh, Scotland, 1952. Original work published 1790.

[24] Jeremy Bentham. Principles of penal law. In J. Bowring, editor, *The Works of Jeremy Bentham*, volume 1, page 396. W. Tait, Edinburgh, Scotland, 1962. Original work published 1843.

[25] Alexander Artikis, Jeremy Pitt, and Marek J. Sergot. Animated specifications of computational societies. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings* DBL [49], pages 1053–1061. doi: 10.1145/545056.545070. URL http://doi.acm.org/10.1145/545056.545070.

[26] Marc Esteva, David de la Cruz, Bruno Rosell, Josep Lluís Arcos, Juan A. Rodríguez-Aguilar, and Guifre Cuní. Engineering open multi-agent systems as electronic institutions. In Deborah L. McGuinness and George Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose,*

California, USA, pages 1010–1011. AAAI Press / The MIT Press, 2004. ISBN 0-262-51183-5. URL `http://www.aaai.org/Library/AAAI/2004/aaai04-153.php`.

[27] Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings* DBL [49], pages 475–482. doi: 10.1145/544741.544854. URL `http://doi.acm.org/10.1145/544741.544854`.

[28] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684): 409–10, 1998.

[29] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, Jan 2002. doi: 10.1103/RevModPhys.74.47. URL `http://link.aps.org/doi/10.1103/RevModPhys.74.47`.

[30] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. doi: 10.1137/S003614450342480. URL `http://dx.doi.org/10.1137/S003614450342480`.

[31] Marta C. González, Pedro G. Lind, and Hans J. Herrmann. System of mobile agents to model social networks. *Phys. Rev. Lett.*, 96:088702, Mar 2006. doi: 10.1103/PhysRevLett.96.088702. URL `http://link.aps.org/doi/10.1103/PhysRevLett.96.088702`.

[32] Marta C. Gonzlez, Pedro G. Lind, and Hans J. Herrmann. Networks based on collisions among mobile agents. *Physica D: Nonlinear Phenomena*, 224(12):137 – 148, 2006. ISSN 0167-2789. doi: http://dx.doi.org/10.1016/j.physd.2006.09.025. URL `http://www.sciencedirect.com/science/article/pii/S0167278906003770`. Dynamics on Complex Networks and Applications.

[33] B Edmonds, N Gilbert, S Gustafson, D Hales, and N Krasnogor. Socially inspired computing. In *Proceedings of the Joint Symposium on Socially Inspired Computing, University of Hertfordshire, Hatfield, UK*, 2005.

[34] Andrew J. I. Jones, Alexander Artikis, and Jeremy Pitt. The design of intelligent socio-technical systems. *Artif. Intell. Rev.*, 39(1):5–20, 2013. doi: 10.1007/s10462-012-9387-2. URL `http://dx.doi.org/10.1007/s10462-012-9387-2`.

[35] Luc Steels and Rodney Brooks, editors. *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1995. ISBN 080581518X.

[36] Brendan Neville and Jeremy Pitt. A computational framework for social agents in agent mediated e-commerce. In Andrea Omicini, Paolo Petta, and Jeremy Pitt, editors, *Engineering Societies in the Agents World IV, 4th International Workshop, ESAW 2003, London, UK, October 29-31, 2003, Revised Selected and Invited Papers*, volume 3071 of *Lecture Notes in Computer Science*, pages 376–391. Springer, 2003. ISBN 3-540-22231-6. doi: 10.1007/978-3-540-25946-6_24. URL `http://dx.doi.org/10.1007/978-3-540-25946-6\_24`.

[37] Patricio E. Petruzzi, Dídac Busquets, and Jeremy V. Pitt. A generic social capital framework for optimising self-organised collective action. In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems, Cambridge, MA, USA, September 21-25, 2015*, pages 21–30. IEEE Computer Society, 2015. ISBN 978-1-4673-7535-1. doi: 10.1109/SASO.2015.10. URL `http://dx.doi.org/10.1109/SASO.2015.10`.

[38] Cynthia Nikolai and Gregory Madey. Tools of the trade: A survey of various agent based modeling platforms. *Journal of Artificial Societies and Social Simulation*, 12(2):2, 2009. ISSN 1460-7425. URL `http://jasss.soc.surrey.ac.uk/12/2/2.html`.

[39] Uri Wilensky. Netlogo. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.*, 1999. URL `http://ccl.northwestern.edu/netlogo/`.

[40] Nick Collier. Repast: An extensible framework for agent simulation. *Natural Resources and Environmental Issues*, 8(4), 2001. URL e.

[41] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, July 2005. ISSN 0037-5497. doi: 10.1177/0037549705058073. URL http://dx.doi.org/10.1177/0037549705058073.

[42] Jeremy Pitt, Brendan Neville, Sam Macbeth, and Hugo Carr. Animation of open multi-agent systems. In Levent Yilmaz and Tuncer I. Ören, editors, *2011 Spring Simulation Multi-conference, SpringSim '11, Boston, MA, USA, April 03-07, 2011. Volume 1: Proceedings of the 2011 Workshop on Agent-Directed Simulation (ADS).*, pages 100–107. SCS/ACM, 2011. ISBN 1-930638-56-6. URL http://dl.acm.org/citation.cfm?id=2048369.

[43] Isaac Ehrlich. The deterrent effect of capital punishment: A question of life and death. Working Paper 18, National Bureau of Economic Research, November 1973. URL http://www.nber.org/papers/w0018.

[44] Ann Seidman and Robert B Seidman. Iltam: Drafting evidence-based legislation for democratic social change. *BUL Rev.*, 89:435, 2009.

[45] David L Sackett, William M C Rosenberg, J A Muir Gray, R Brian Haynes, and W Scott Richardson. Evidence based medicine: what it is and what it isn't. *BMJ*, 312(7023):71–72, 1996. ISSN 0959-8138. doi: 10.1136/bmj.312.7023.71. URL http://www.bmj.com/content/312/7023/71.

[46] Amos Schurr and Ilana Ritov. Winning a competition predicts dishonest behavior. *Proceedings of the National Academy of Sciences*, 113(7):1754–1759, 2016. doi: 10.1073/pnas.1515102113. URL http://www.pnas.org/content/113/7/1754.abstract.

[47] Emily M Zitek, Alexander H Jordan, Benoît Monin, and Frederick R Leach. Victim entitlement to behave selfishly. *Journal of personality and social psychology*, 98(2):245, 2010.

[48] Asimina Vasalou, Astrid Hopfensitz, and Jeremy V. Pitt. In praise of forgiveness: Ways for repairing trust breakdowns in one-off online interactions. *Int. J. Hum.-Comput. Stud.*, 66(6):466–480, 2008. doi: 10.1016/j.ijhcs.2008.02.001. URL http://dx.doi.org/10.1016/j.ijhcs.2008.02.001.

[49] *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, 2002, Bologna, Italy, Proceedings*, 2002. ACM.

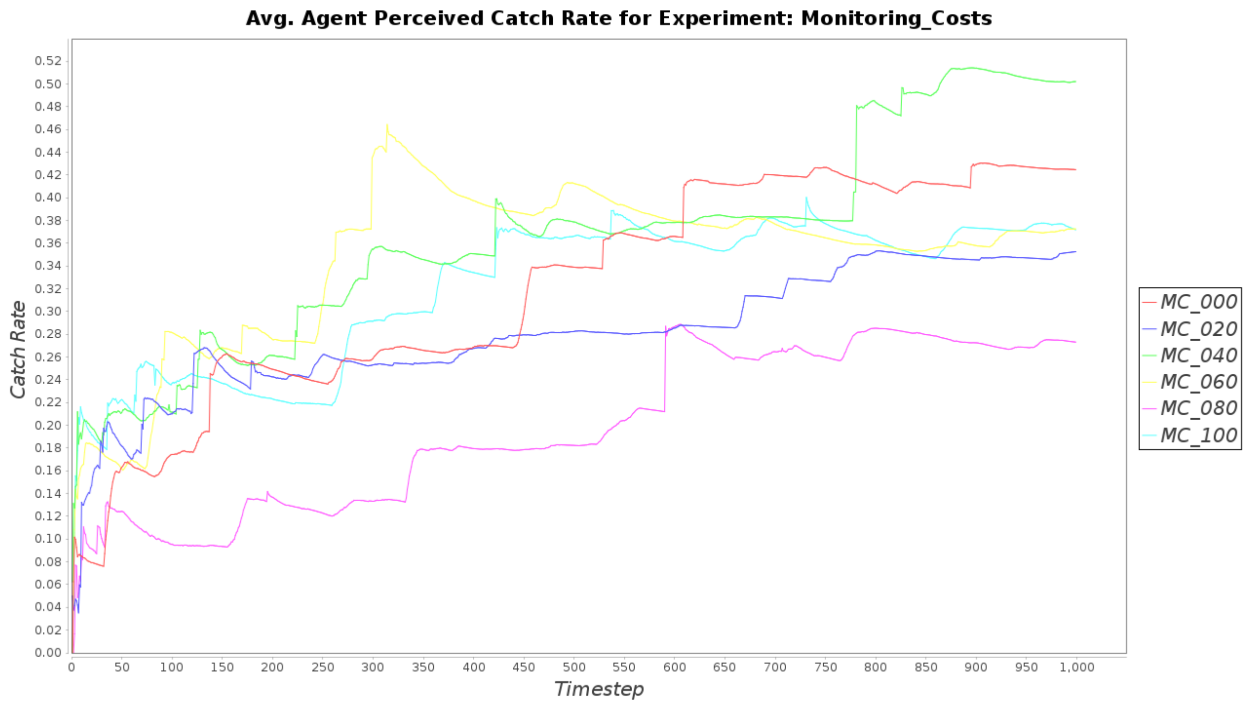
Appendices

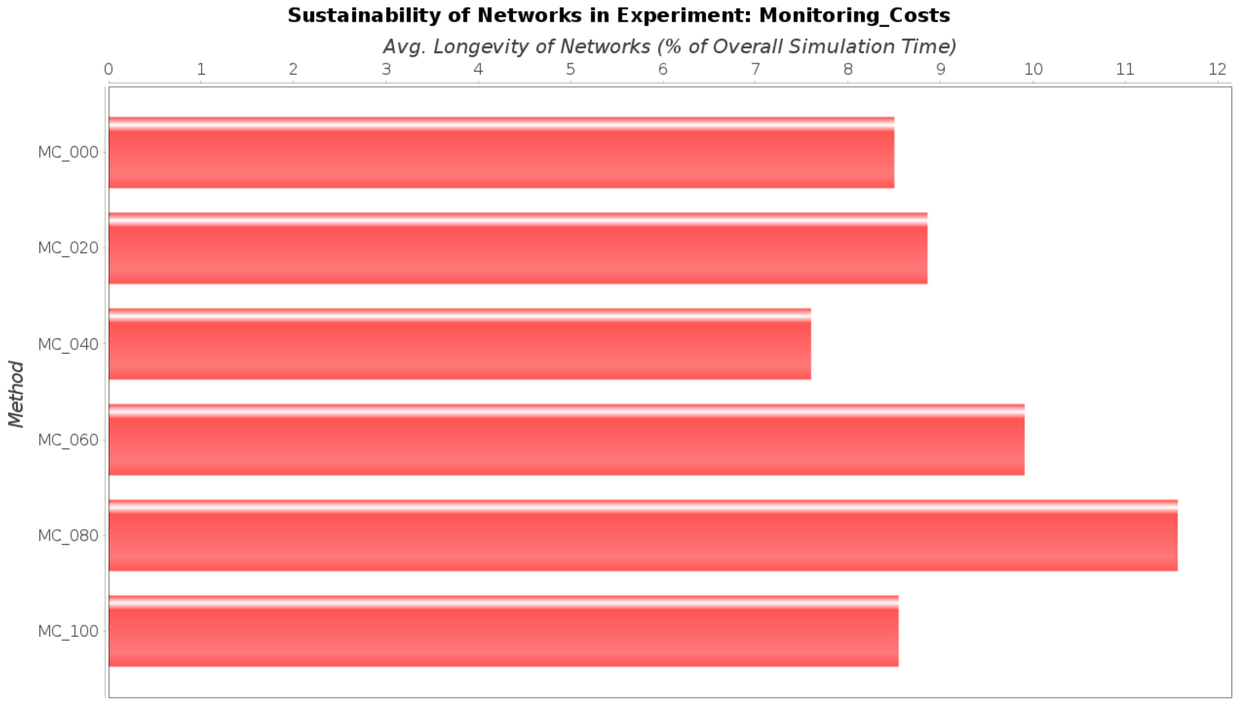
A User Guide

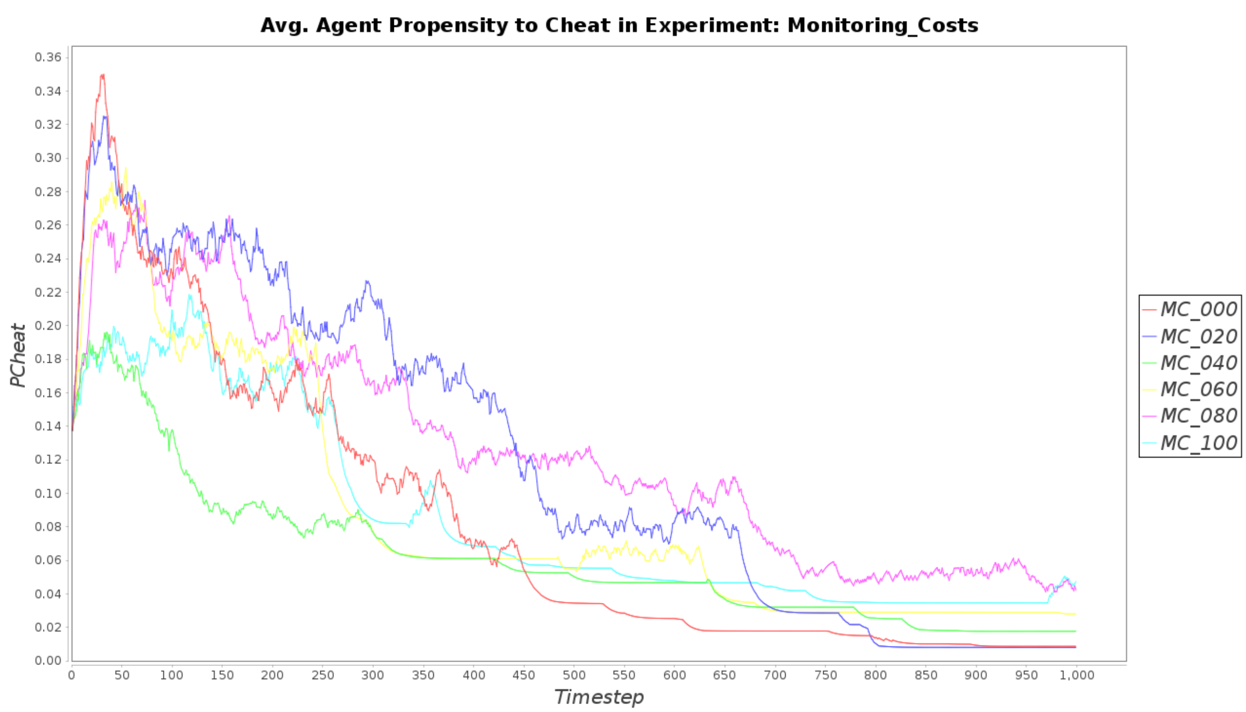
1. Git Repository Link (Public) with full source code implementation:
   - `https://github.com/mazrk7/SpinWorld`
2. A complete guide for setting up this project in the Eclipse IDE to use with Presage2 is available at the following link:
   - `http://www.presage2.info/w/Getting_Started_Guide`
3. Other sources of code that influenced development:
   - `https://github.com/sammacbeth/LPG--Game`
4. Packages that are open-source and were used for development:
   - `http://jung.sourceforge.net/`
   - `https://github.com/mkrauskopf/jfreechart-patches`
   - `http://www.jfree.org/jfreechart/`

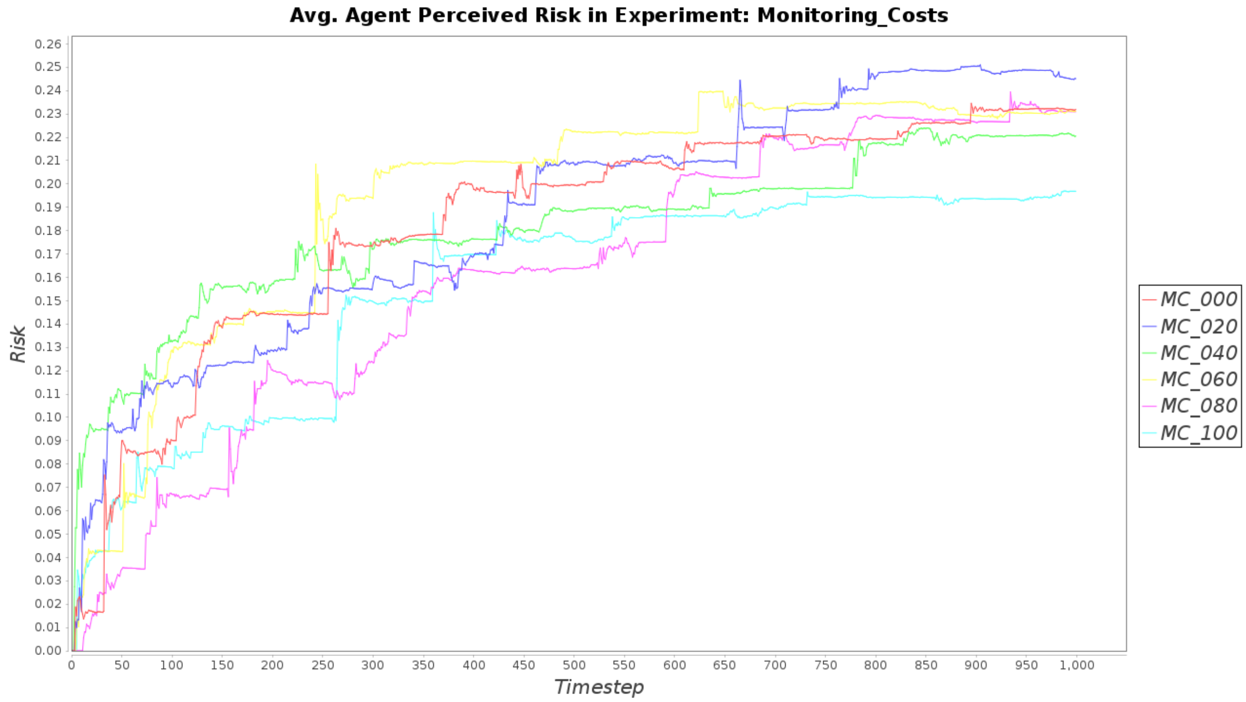
B Extended Experimental Results

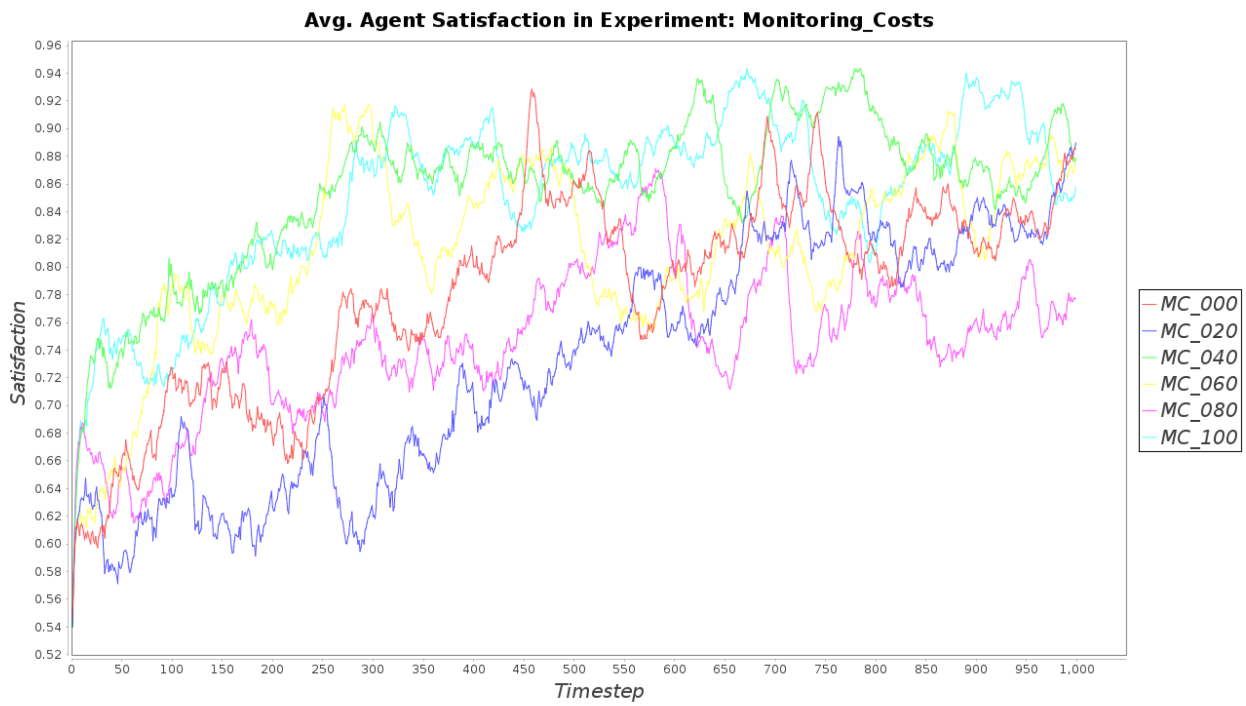
All Experiment A plots:

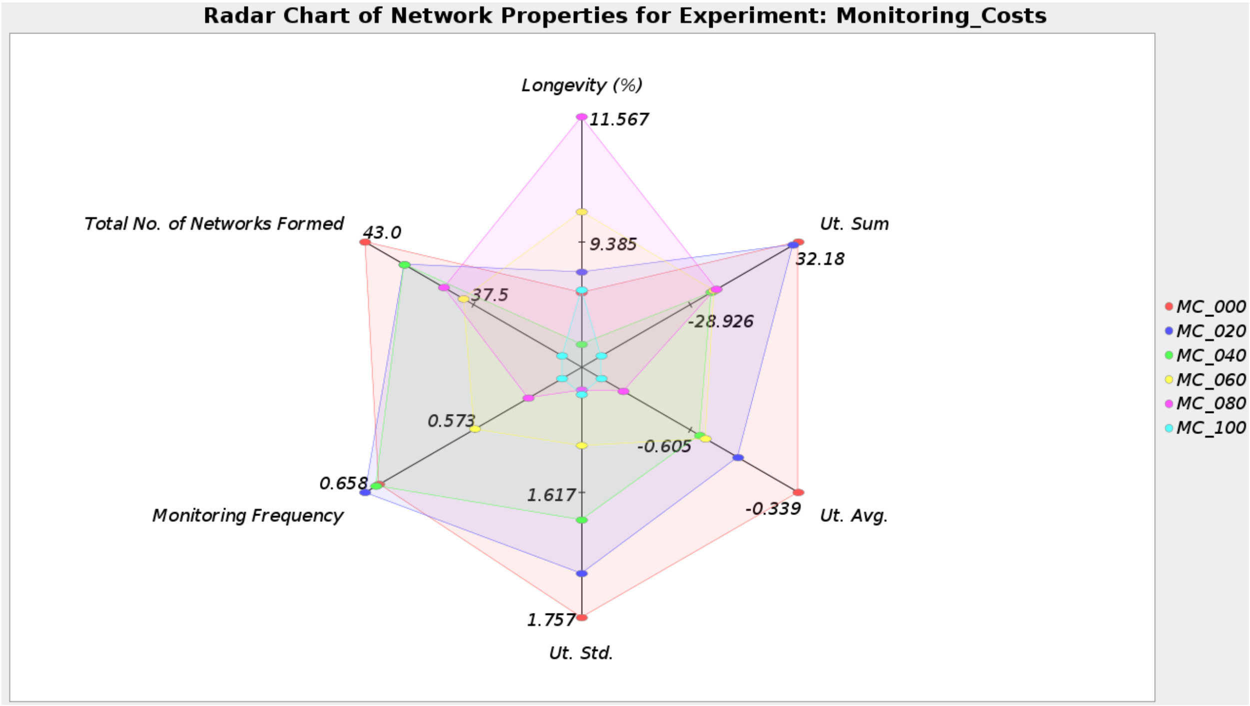


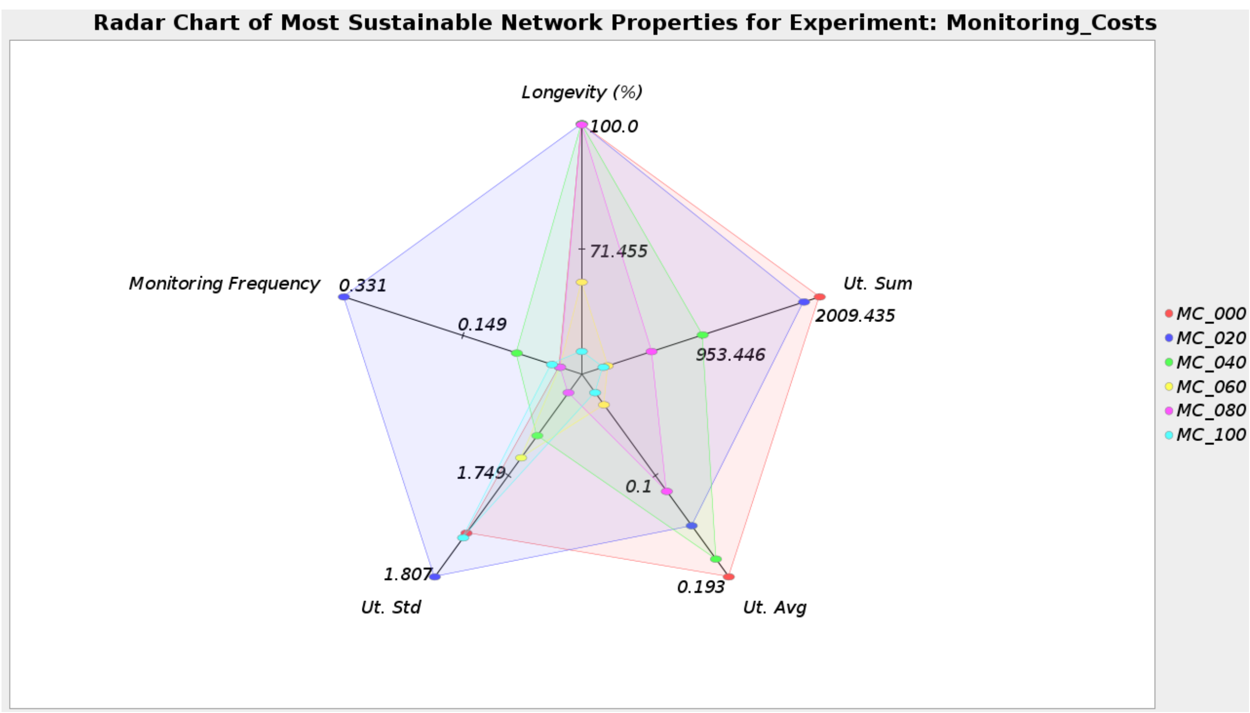

Sustainability of Networks in Experiment: Monitoring_Costs
Avg. Longevity of Networks (% of Overall Simulation Time)
Method
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100

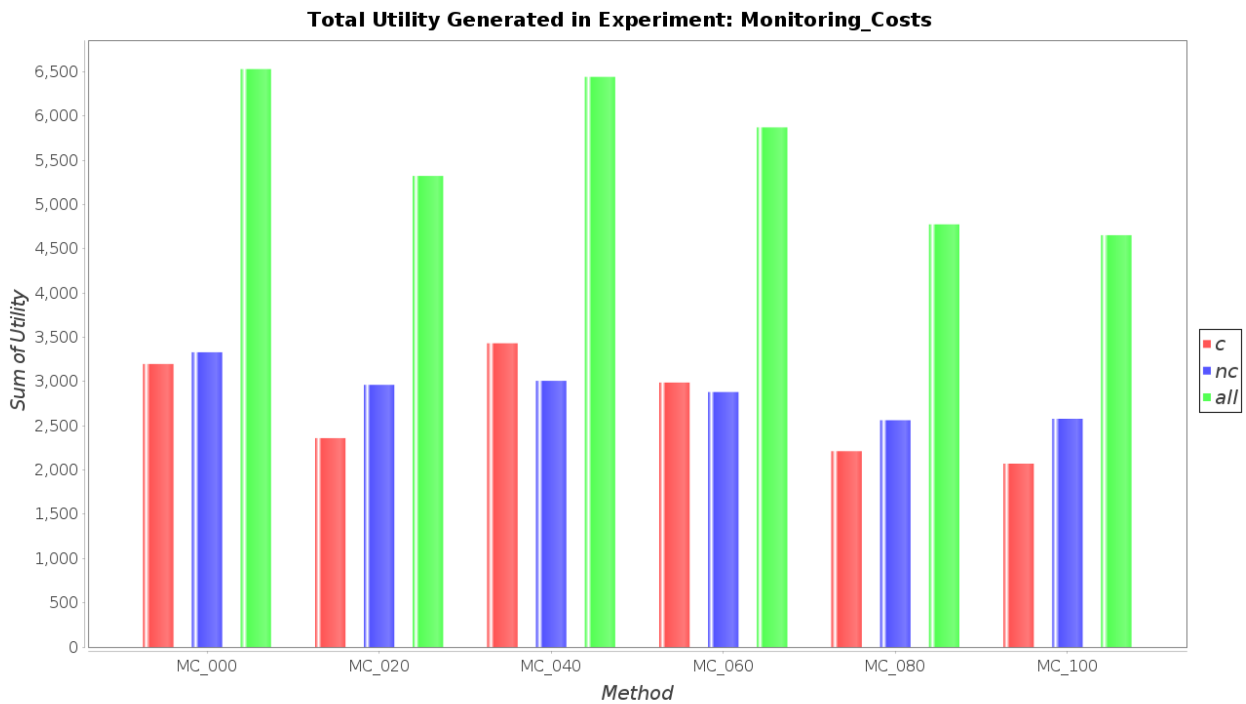

Avg. Agent Propensity to Cheat in Experiment: Monitoring_Costs
PCheat
Timestep
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100


Avg. Agent Perceived Risk in Experiment: Monitoring_Costs
Risk
Timestep
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100


Avg. Agent Satisfaction in Experiment: Monitoring_Costs
Satisfaction
Timestep
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100

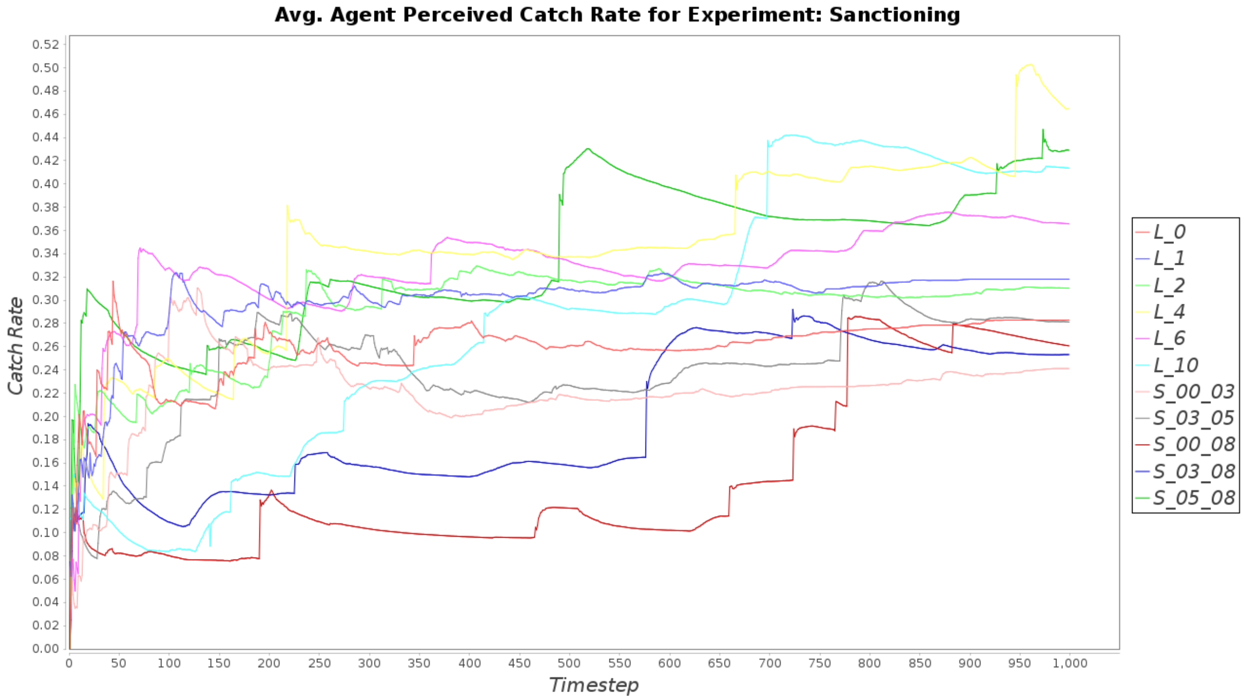

Radar Chart of Network Properties for Experiment: Monitoring_Costs
Longevity (%)
11.567
9.385
Ut. Sum
32.18
-28.926
Ut. Avg.
-0.339
-0.605
Ut. Std.
1.757
1.617
Monitoring Frequency
0.658
0.573
Total No. of Networks Formed
43.0
37.5
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100


Radar Chart of Most Sustainable Network Properties for Experiment: Monitoring_Costs
Longevity (%)
100.0
71.455
Ut. Sum
2009.435
953.446
Ut. Avg
0.193
0.1
Ut. Std
1.807
1.749
Monitoring Frequency
0.331
0.149
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100

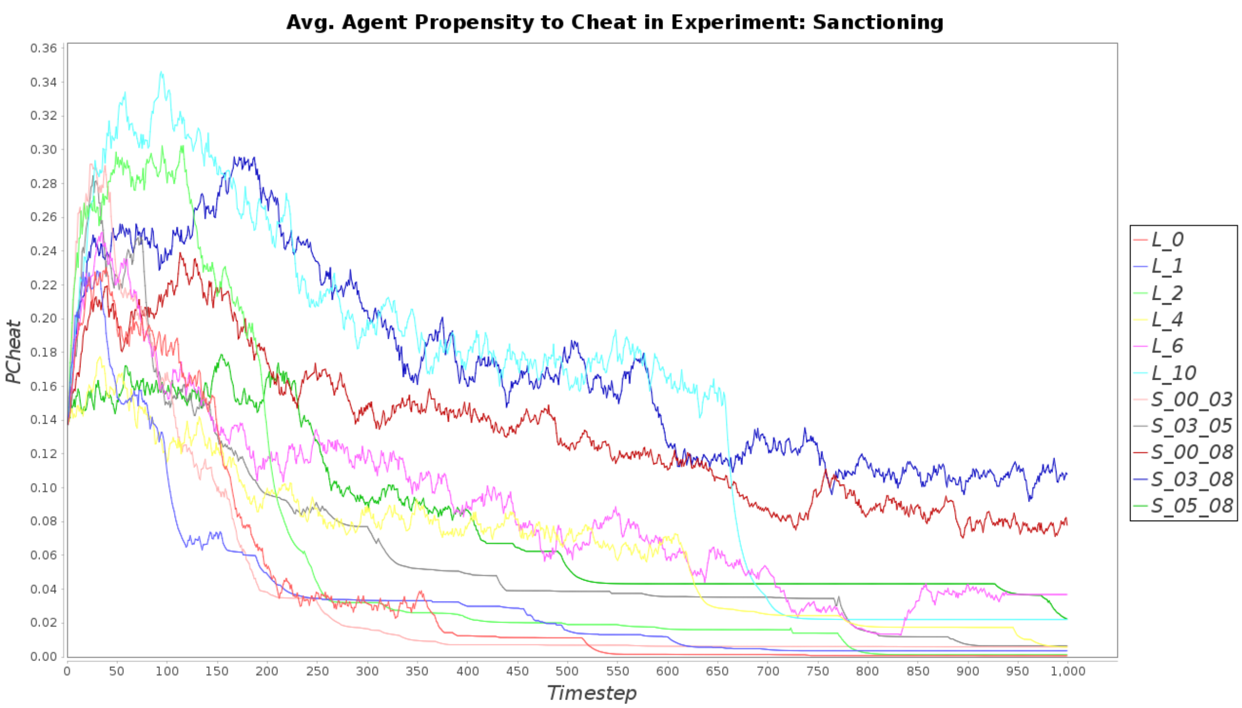


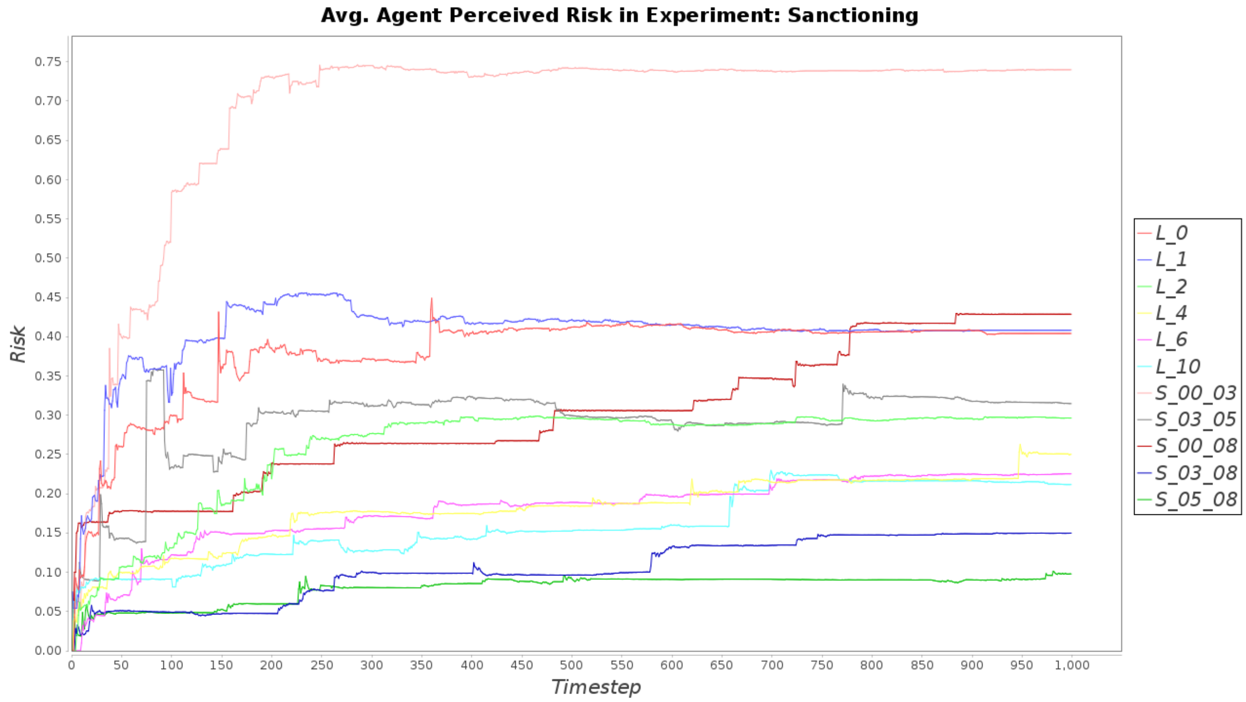
Total Utility Generated in Experiment: Monitoring_Costs
Sum of Utility
6,500
6,000
5,500
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100
Method
c
nc
all

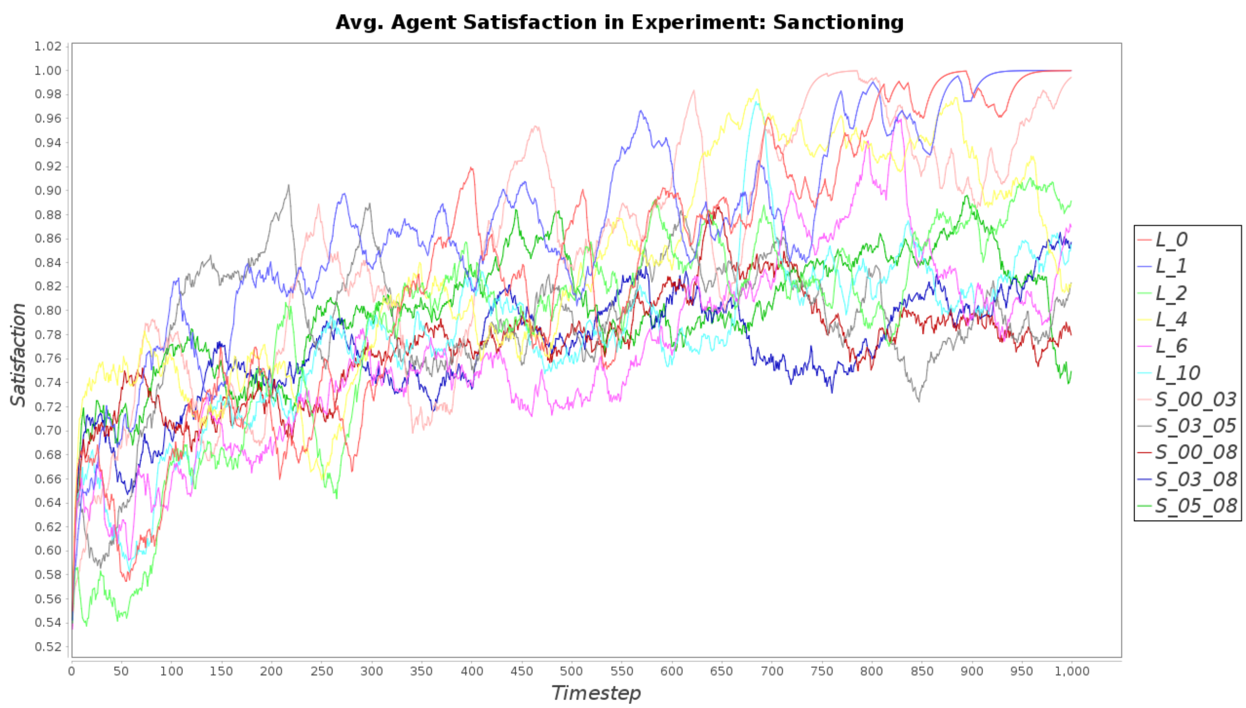
All Experiment B plots:

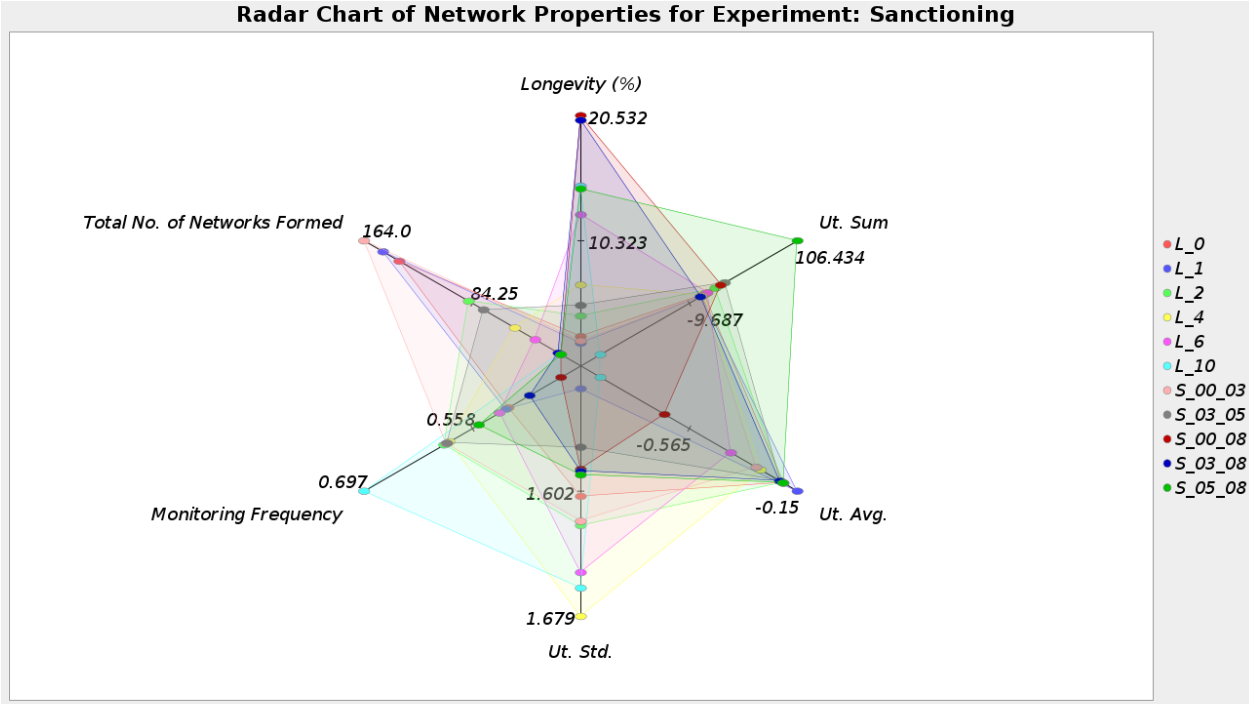


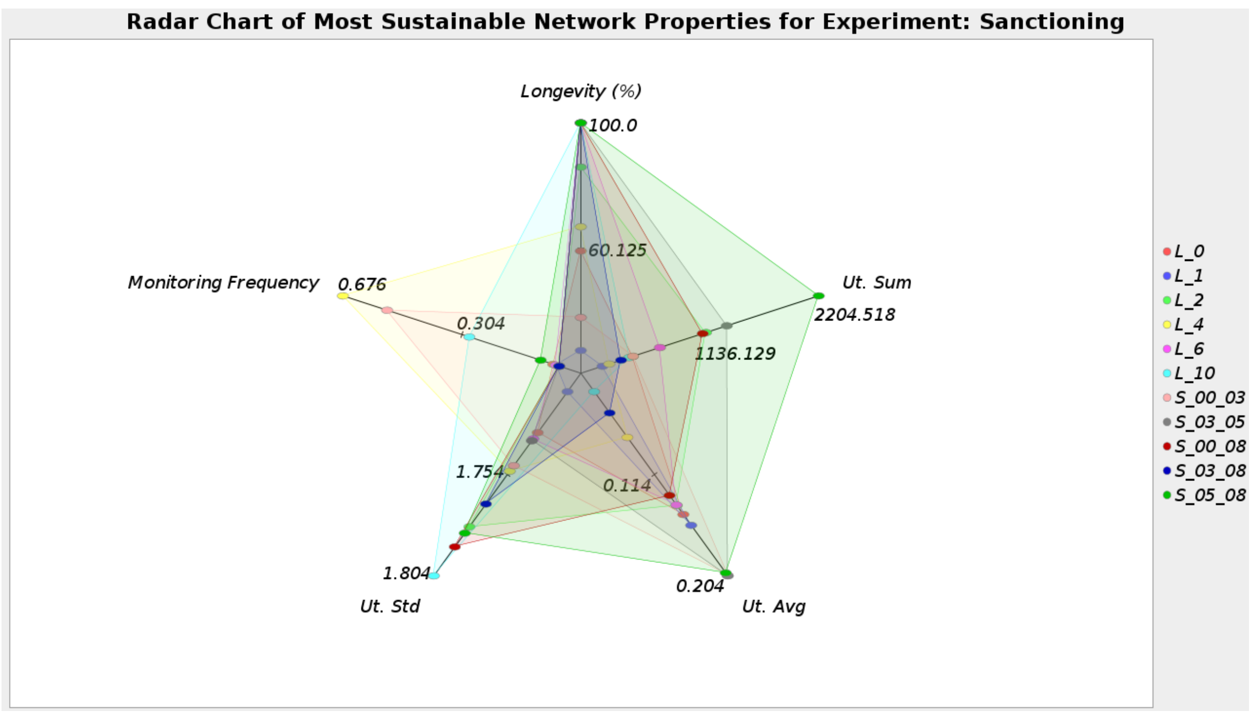

Sustainability of Networks in Experiment: Sanctioning
Avg. Longevity of Networks (% of Overall Simulation Time)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Method
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08

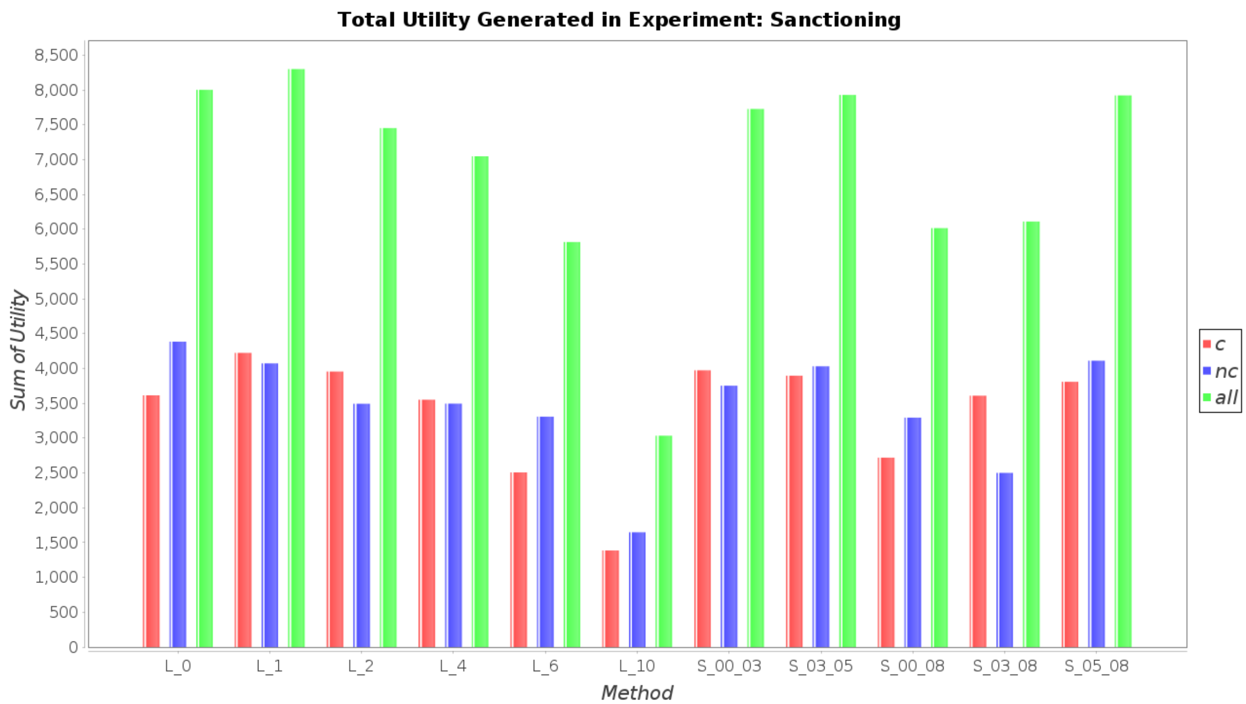

Avg. Agent Propensity to Cheat in Experiment: Sanctioning
PCheat
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1,000
Timestep
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08


Avg. Agent Perceived Risk in Experiment: Sanctioning
Risk
Timestep
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08

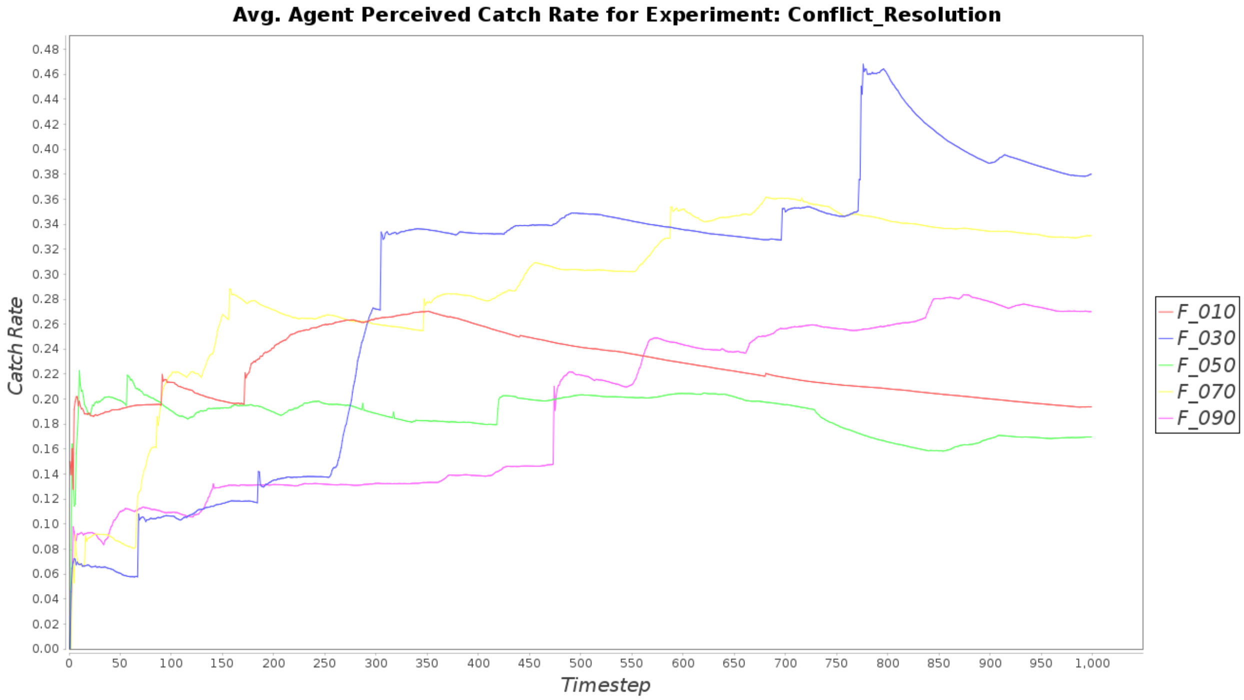

Avg. Agent Satisfaction in Experiment: Sanctioning
Satisfaction
Timestep
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08

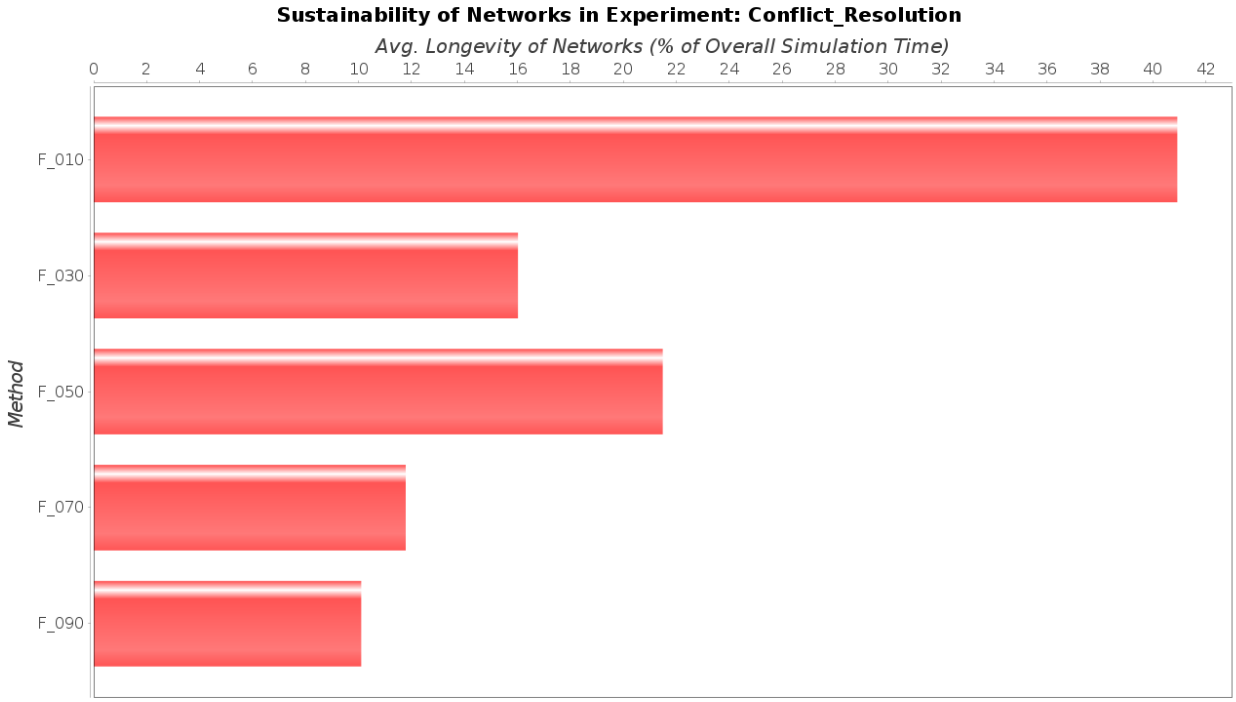

Radar Chart of Network Properties for Experiment: Sanctioning
Longevity (%)
20.532
10.323
Ut. Sum
106.434
-9.687
Ut. Avg.
-0.15
-0.565
Ut. Std.
1.679
1.602
Monitoring Frequency
0.697
0.558
Total No. of Networks Formed
164.0
84.25
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08

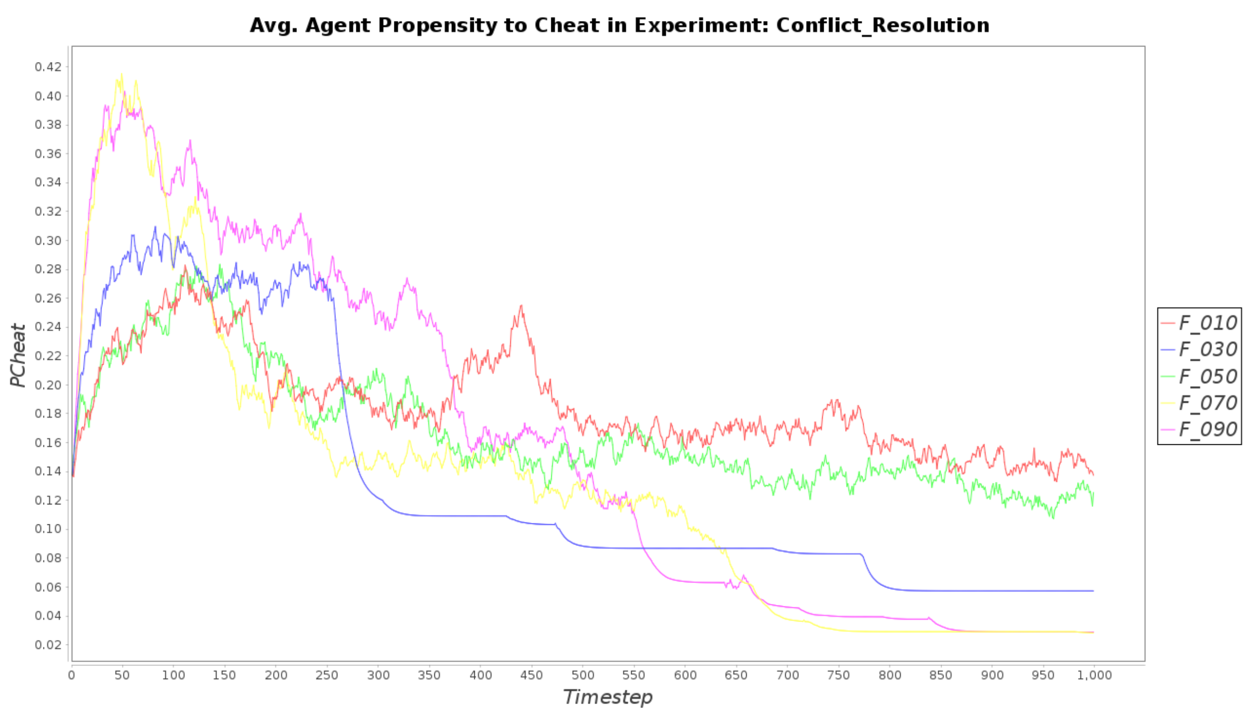

Radar Chart of Most Sustainable Network Properties for Experiment: Sanctioning
Longevity (%)
100.0
60.125
Ut. Sum
2204.518
1136.129
Ut. Avg
0.204
0.114
Ut. Std
1.804
1.754
Monitoring Frequency
0.676
0.304
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08

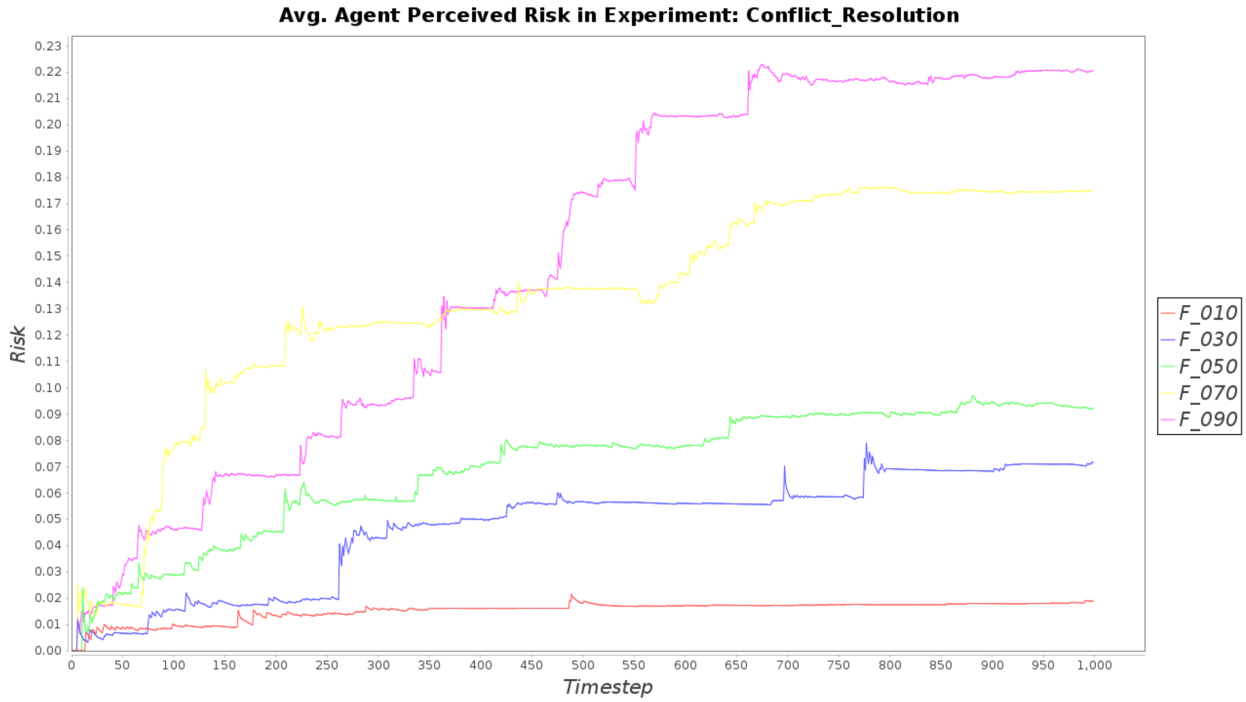

Total Utility Generated in Experiment: Sanctioning
Sum of Utility
Method
8,500
8,000
7,500
7,000
6,500
6,000
5,500
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
L_0
L_1
L_2
L_4
L_6
L_10
S_00_03
S_03_05
S_00_08
S_03_08
S_05_08
c
nc
all

All Experiment C plots:

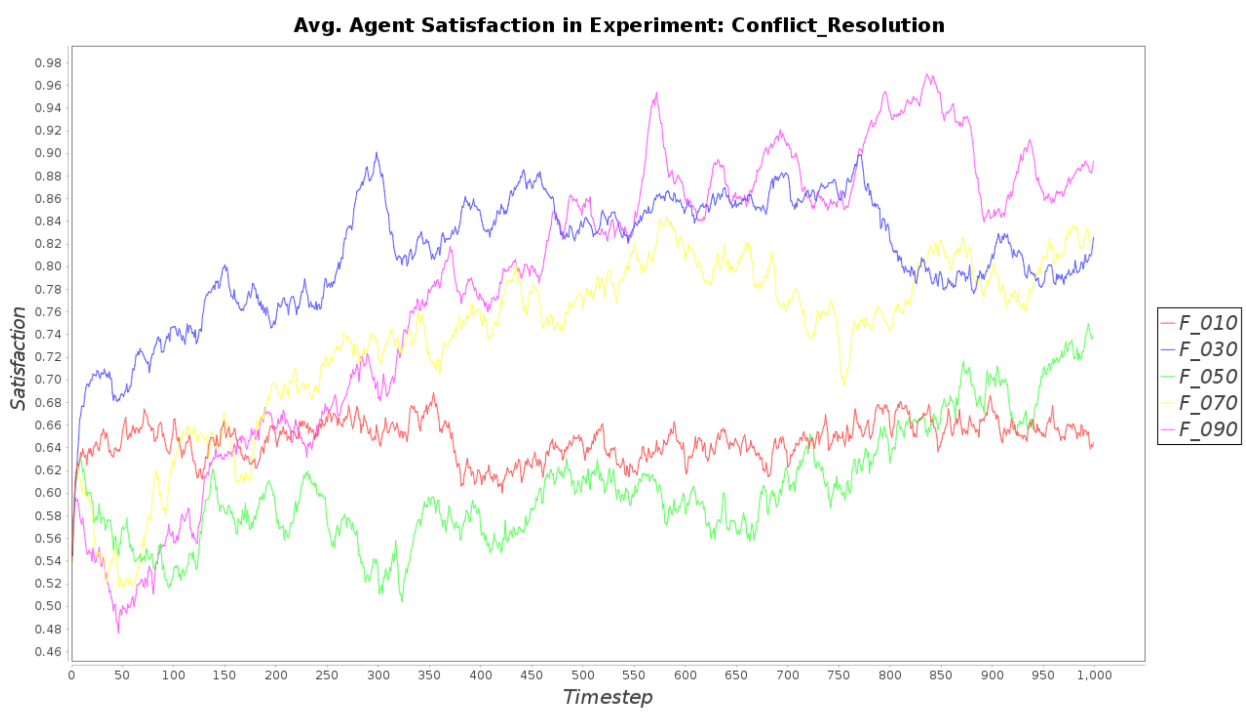



Sustainability of Networks in Experiment: Conflict_Resolution
Avg. Longevity of Networks (% of Overall Simulation Time)
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42
Method
F_010
F_030
F_050
F_070
F_090

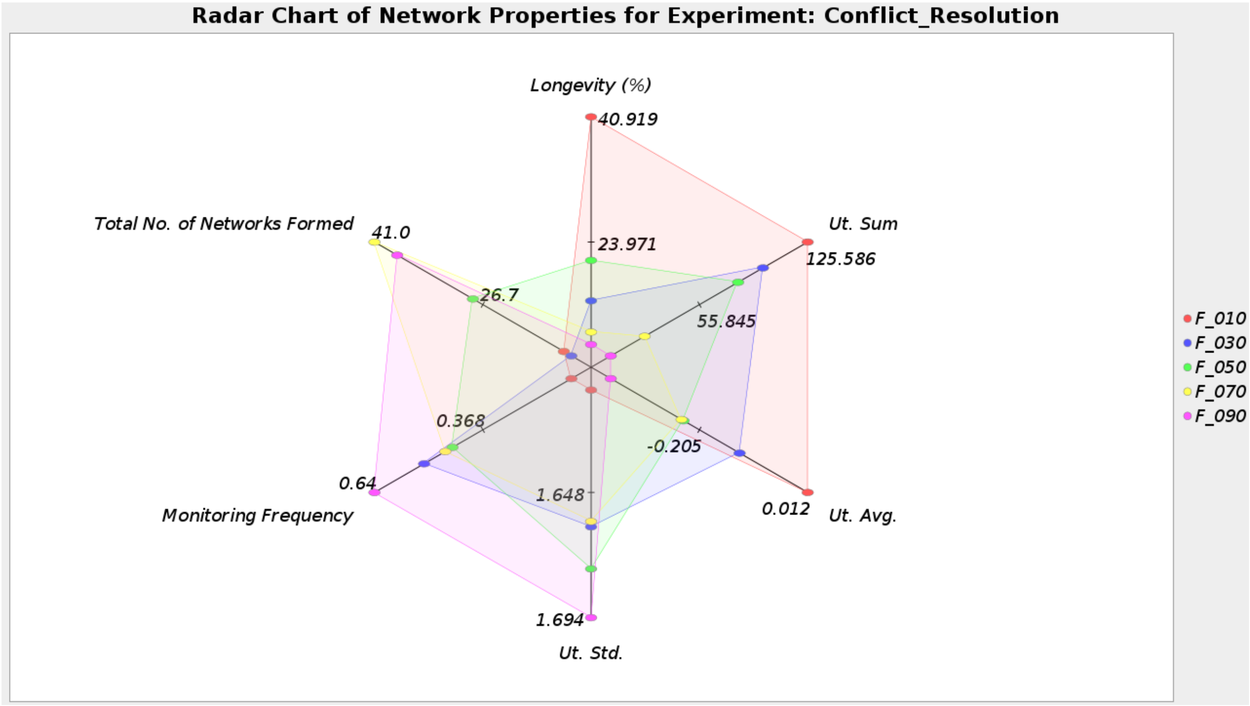

Avg. Agent Propensity to Cheat in Experiment: Conflict_Resolution
PCheat
0.42 0.40 0.38 0.36 0.34 0.32 0.30 0.28 0.26 0.24 0.22 0.20 0.18 0.16 0.14 0.12 0.10 0.08 0.06 0.04 0.02
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1,000
Timestep
F_010
F_030
F_050
F_070
F_090

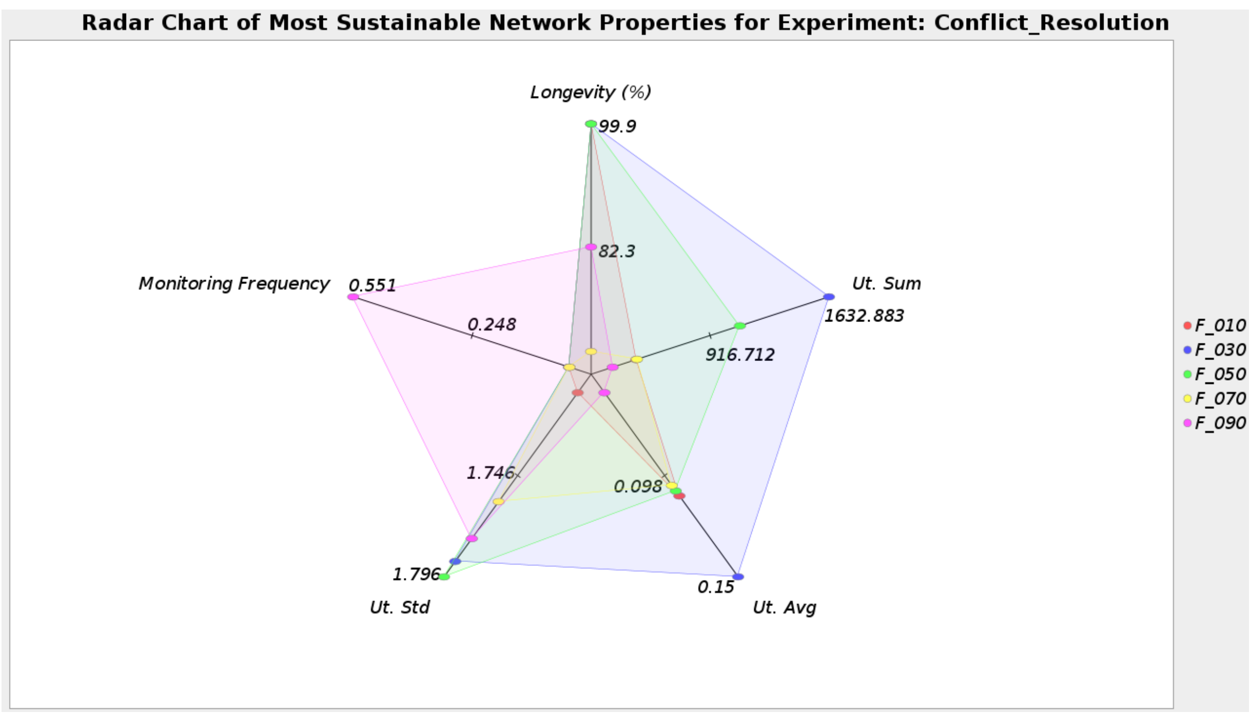


Avg. Agent Perceived Risk in Experiment: Conflict_Resolution
Risk
Timestep
F_010
F_030
F_050
F_070
F_090
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0.11
0.12
0.13
0.14
0.15
0.16
0.17
0.18
0.19
0.20
0.21
0.22
0.23
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1,000



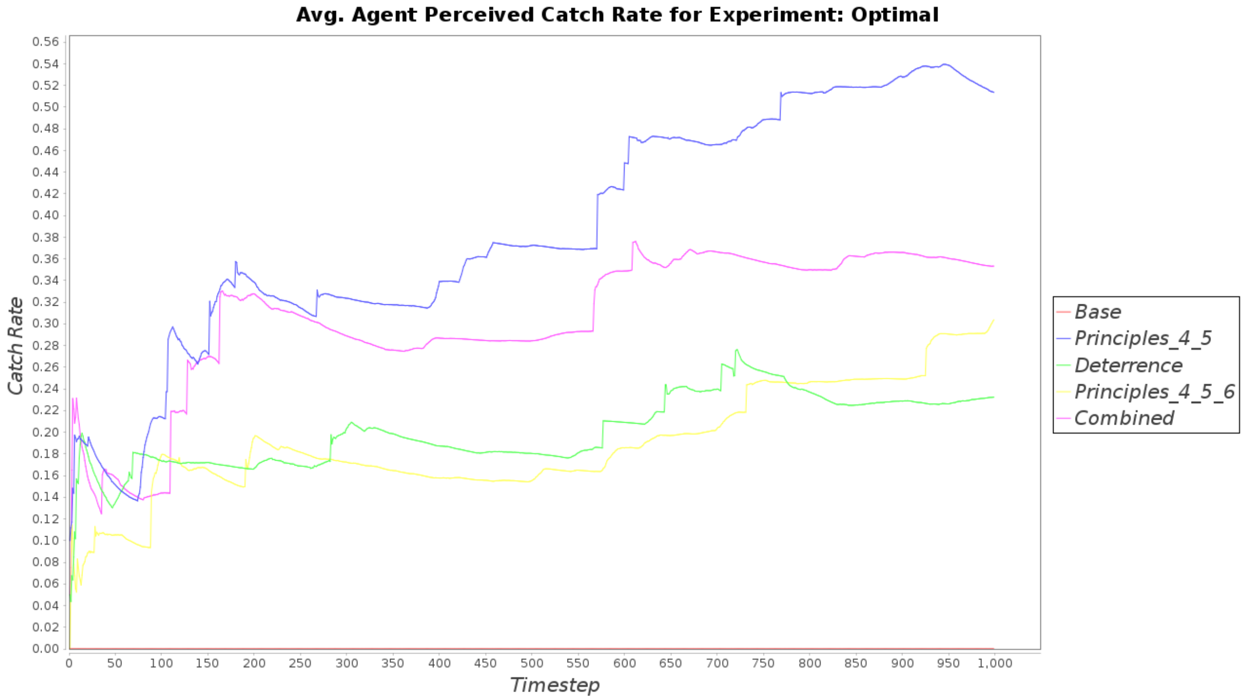
Avg. Agent Satisfaction in Experiment: Conflict_Resolution
Satisfaction
Timestep
F_010
F_030
F_050
F_070
F_090
0.46
0.48
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.68
0.70
0.72
0.74
0.76
0.78
0.80
0.82
0.84
0.86
0.88
0.90
0.92
0.94
0.96
0.98
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1,000

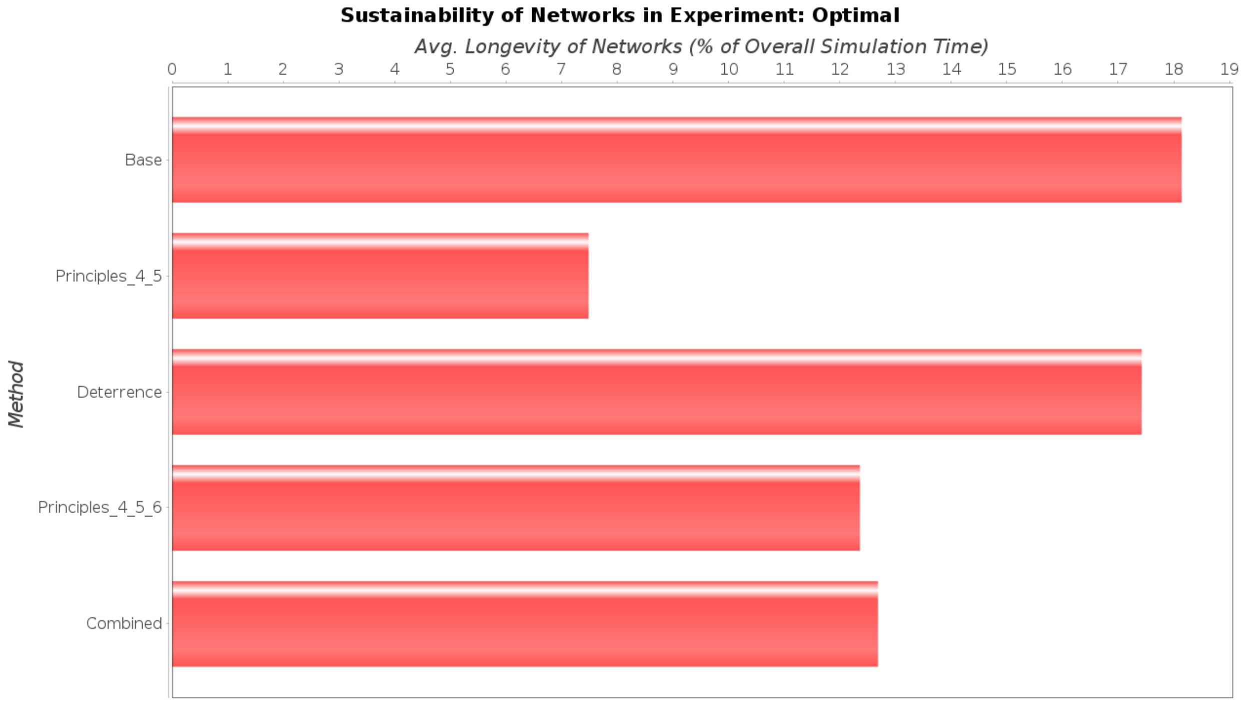

Radar Chart of Network Properties for Experiment: Conflict_Resolution
Longevity (%)
40.919
23.971
Ut. Sum
125.586
55.845
Ut. Avg.
0.012
-0.205
Ut. Std.
1.694
1.648
Monitoring Frequency
0.64
0.368
Total No. of Networks Formed
41.0
26.7
F_010
F_030
F_050
F_070
F_090

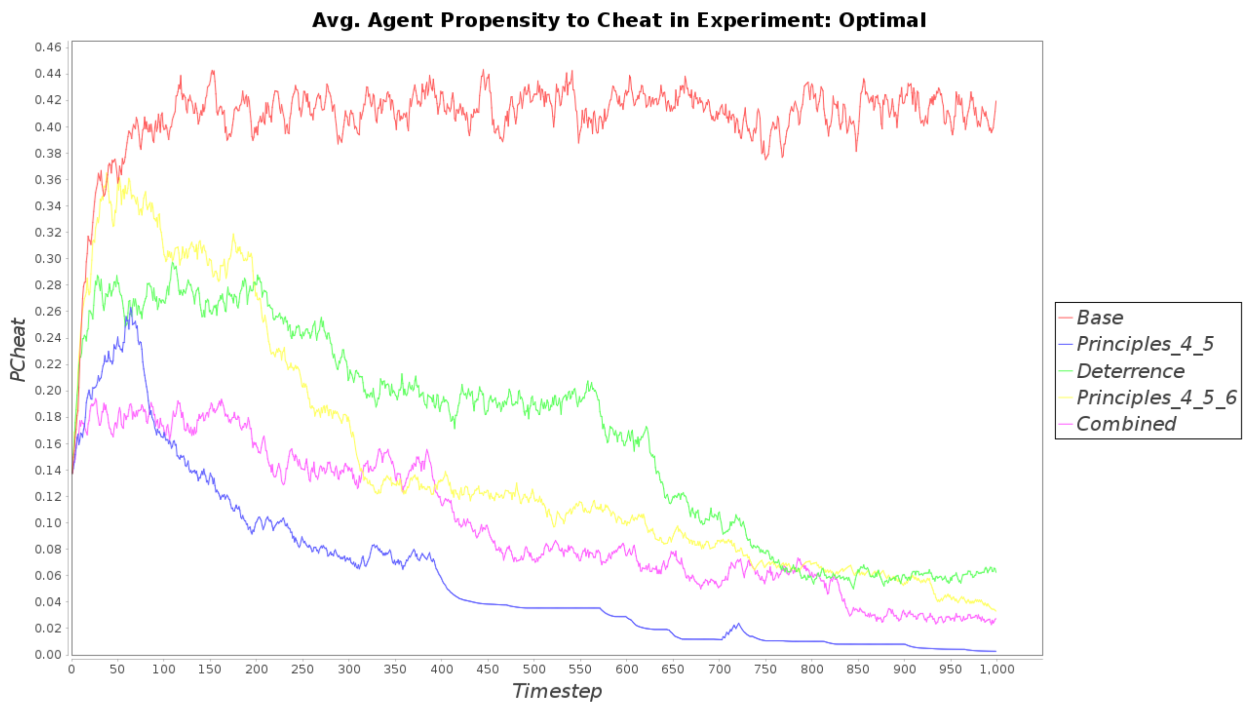

Radar Chart of Most Sustainable Network Properties for Experiment: Conflict_Resolution
Longevity (%)
99.9
82.3
Ut. Sum
1632.883
916.712
Ut. Avg
0.15
0.098
Ut. Std
1.796
1.746
Monitoring Frequency
0.551
0.248
F_010
F_030
F_050
F_070
F_090

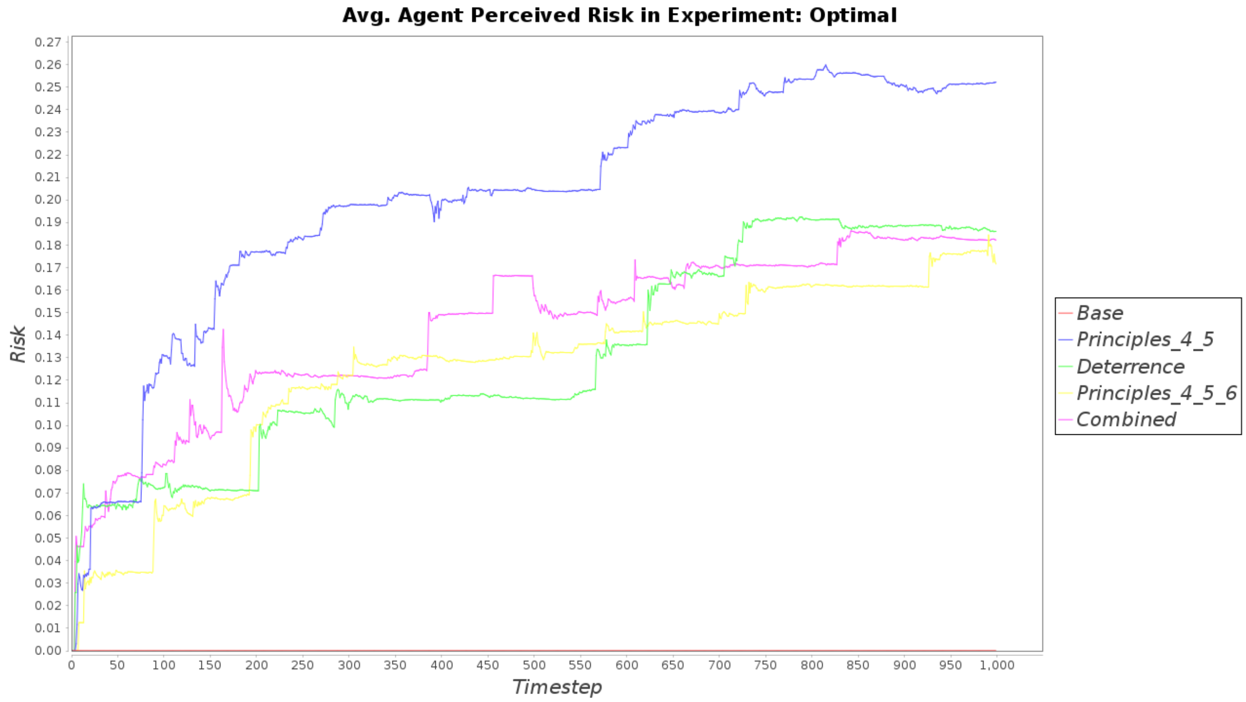

Total Utility Generated in Experiment: Conflict_Resolution
Sum of Utility
7,500
7,000
6,500
6,000
5,500
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
F_010
F_030
F_050
F_070
F_090
Method
c
nc
all

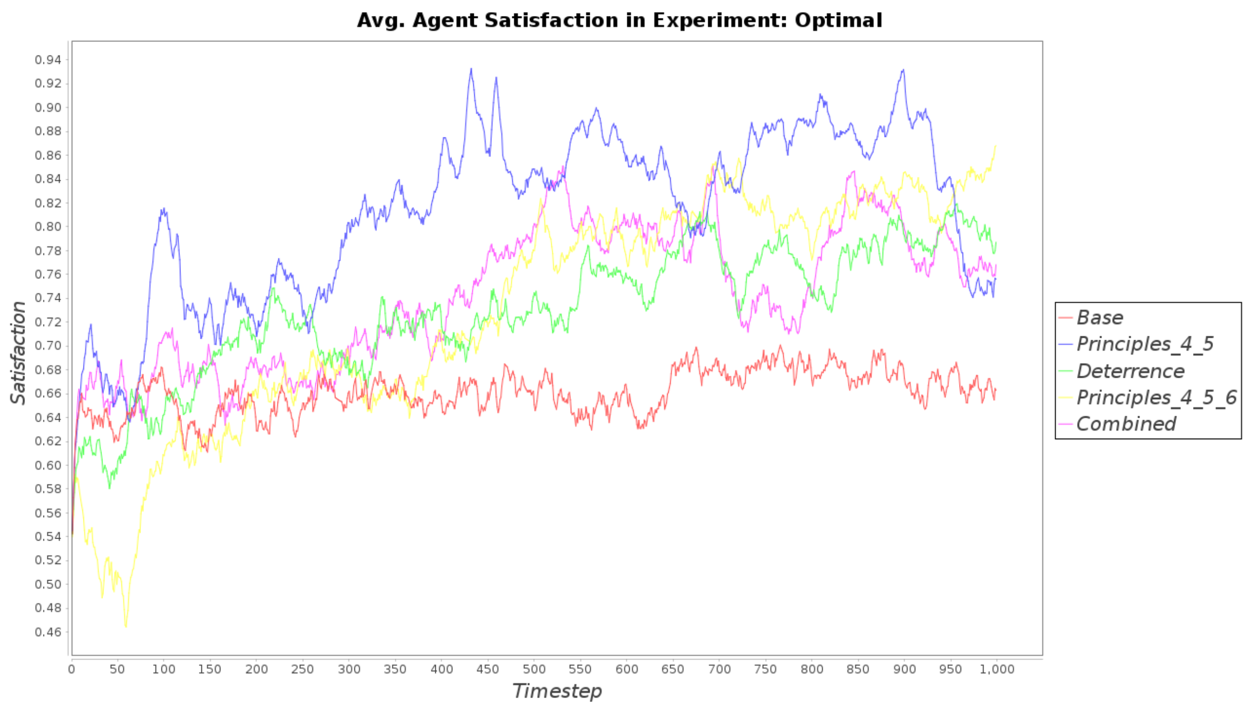
All Experiment D plots:

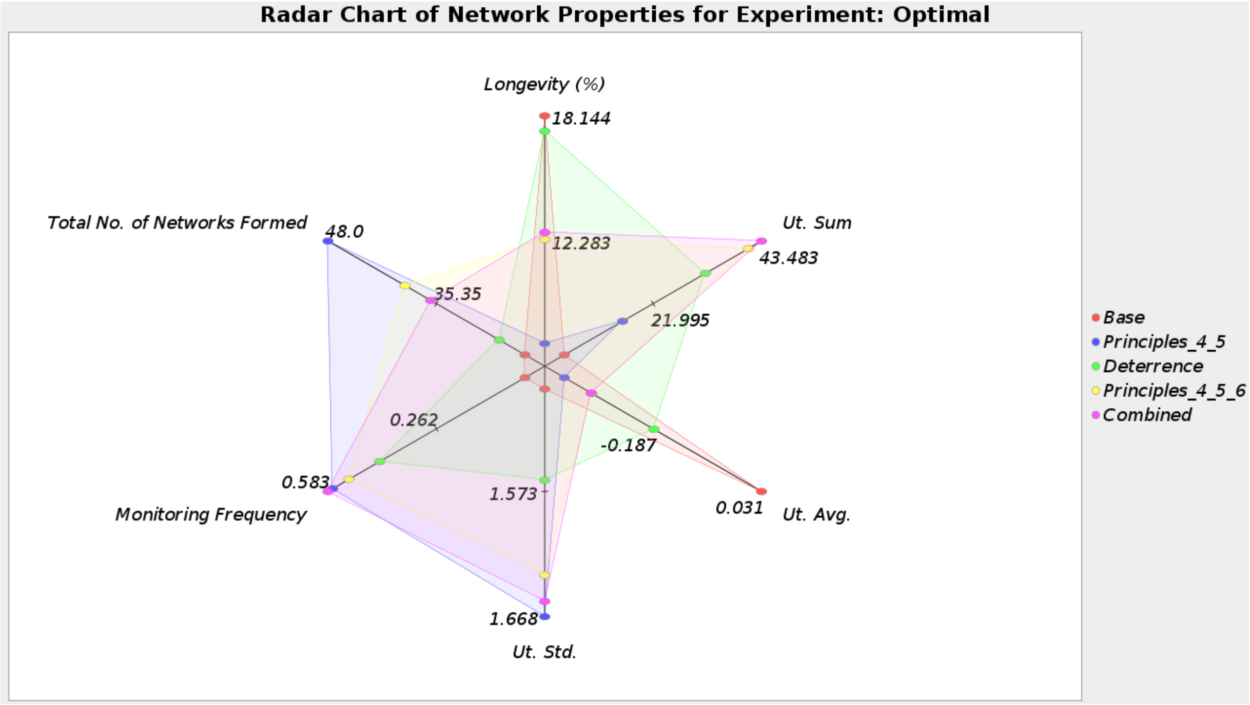


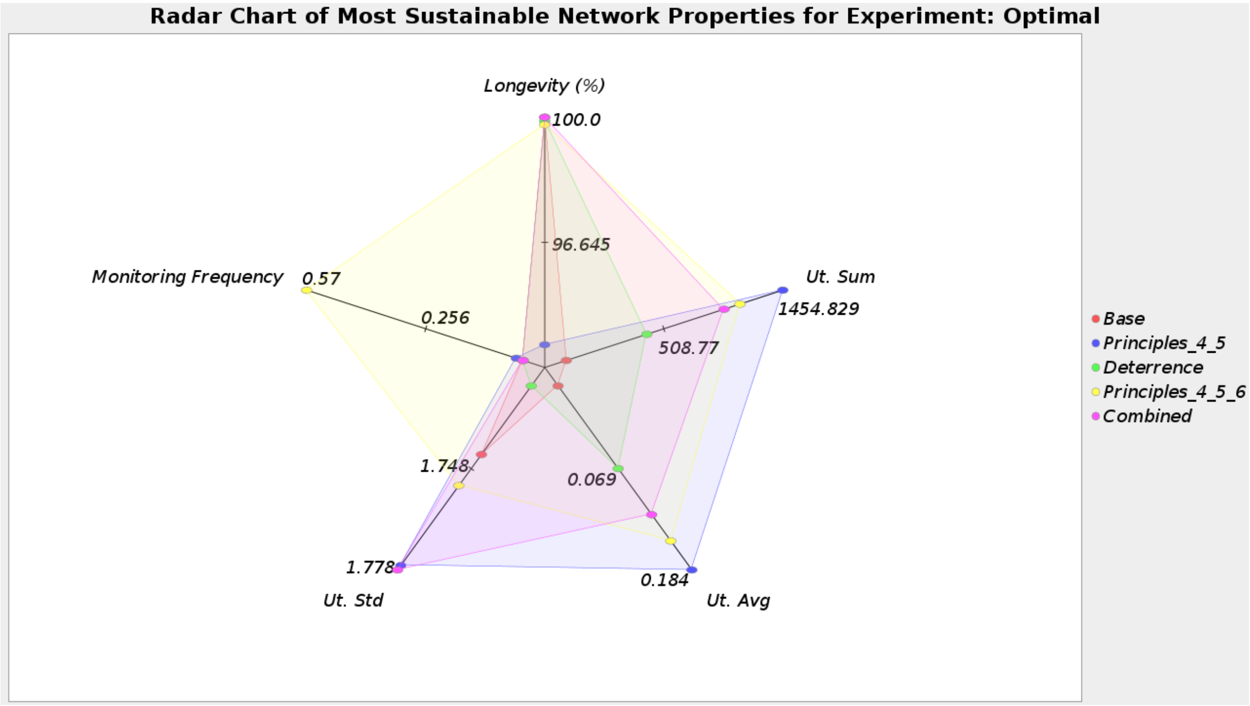

Sustainability of Networks in Experiment: Optimal
Avg. Longevity of Networks (% of Overall Simulation Time)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Method
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined

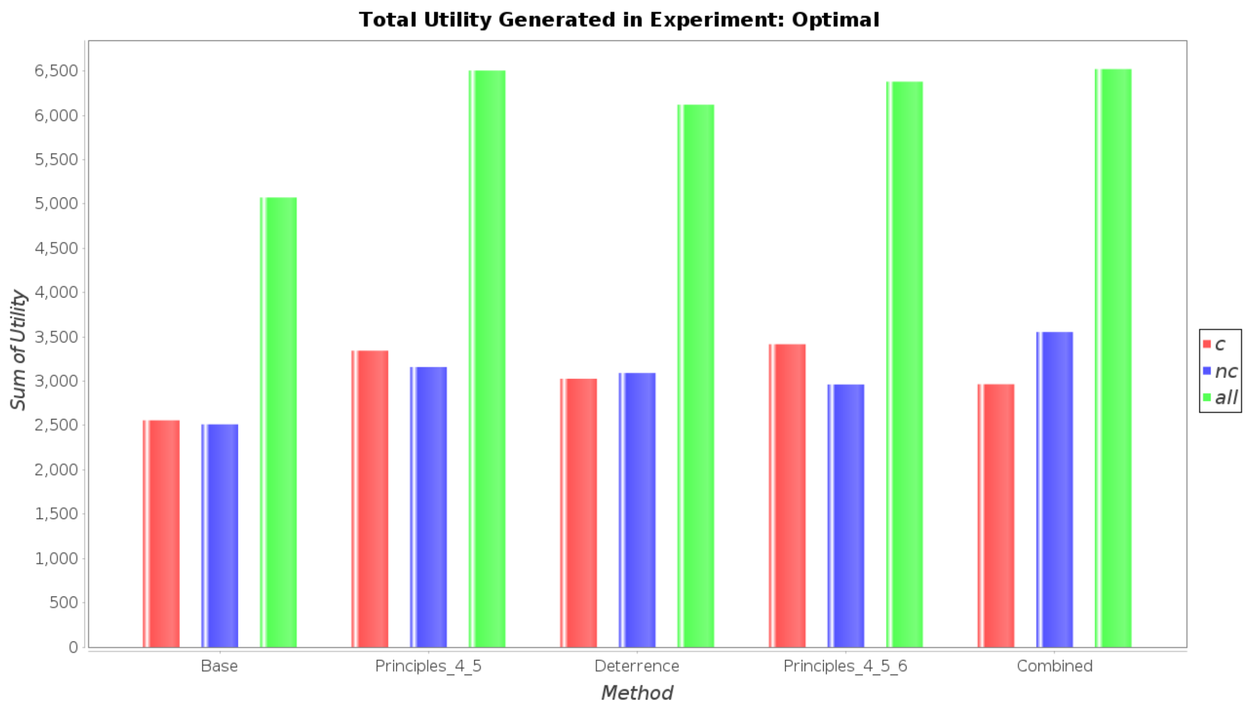

Avg. Agent Propensity to Cheat in Experiment: Optimal
PCheat
0.46 0.44 0.42 0.40 0.38 0.36 0.34 0.32 0.30 0.28 0.26 0.24 0.22 0.20 0.18 0.16 0.14 0.12 0.10 0.08 0.06 0.04 0.02 0.00
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1,000
Timestep
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined


Avg. Agent Perceived Risk in Experiment: Optimal
Risk
Timestep
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined


Avg. Agent Satisfaction in Experiment: Optimal
Satisfaction
Timestep
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined

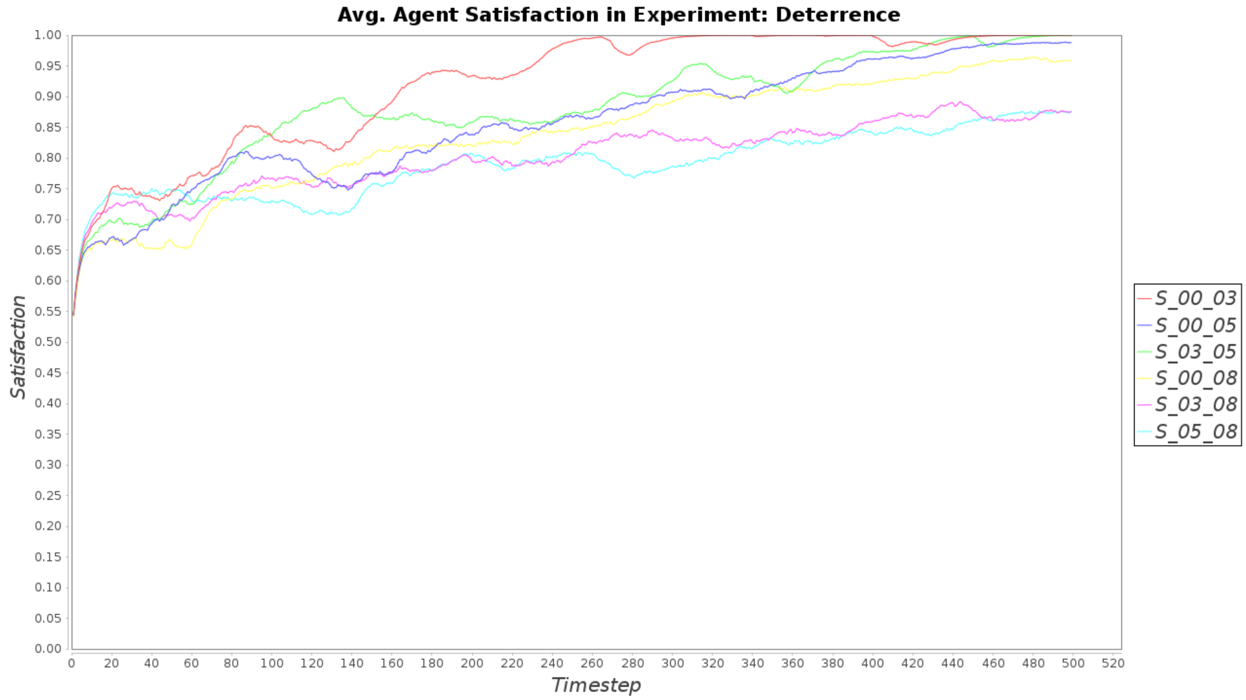

Radar Chart of Network Properties for Experiment: Optimal
Longevity (%)
18.144
12.283
Ut. Sum
43.483
21.995
Ut. Avg.
0.031
-0.187
Ut. Std.
1.668
1.573
Monitoring Frequency
0.583
0.262
Total No. of Networks Formed
48.0
35.35
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined

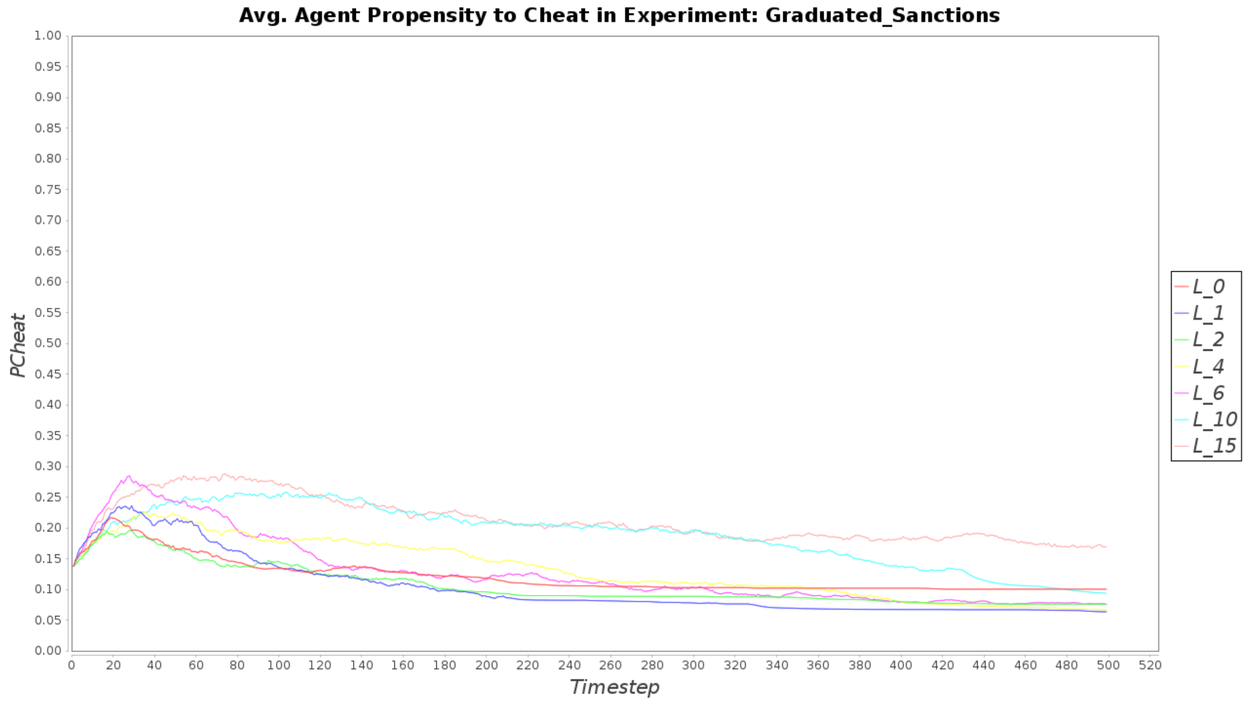

Radar Chart of Most Sustainable Network Properties for Experiment: Optimal
Longevity (%)
100.0
96.645
Ut. Sum
1454.829
508.77
Ut. Avg
0.184
0.069
Ut. Std
1.778
1.748
Monitoring Frequency
0.57
0.256
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined



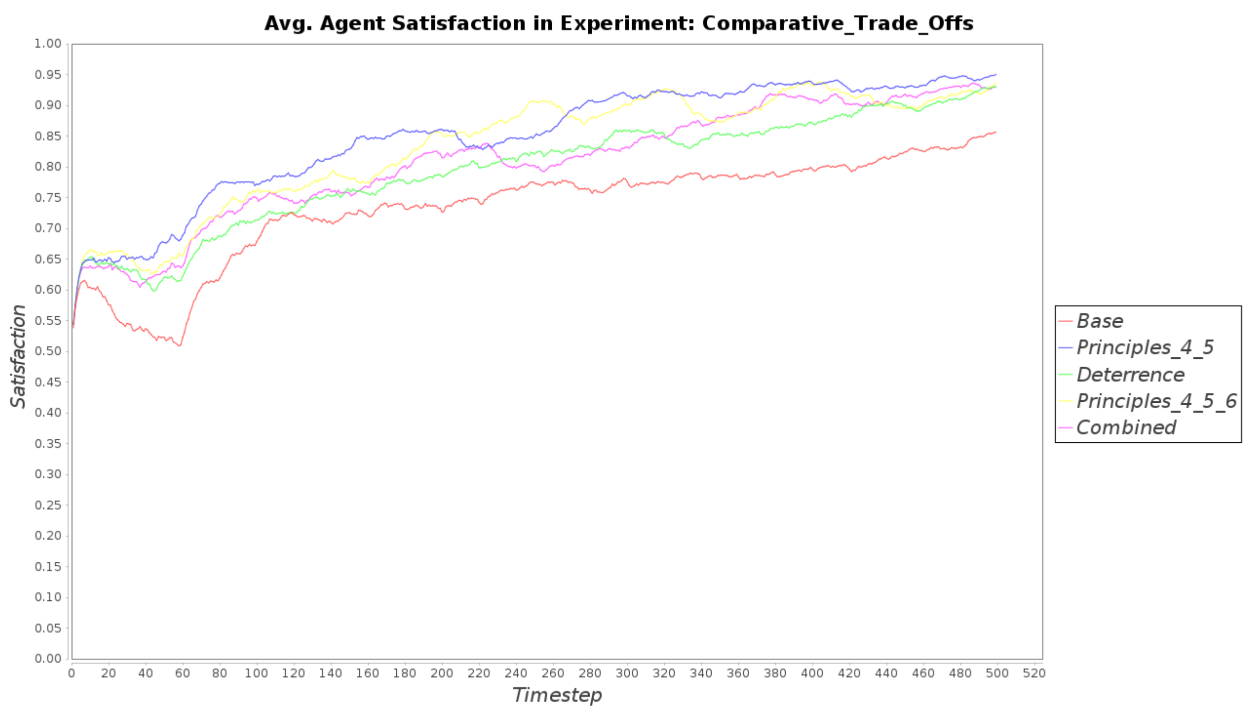
Total Utility Generated in Experiment: Optimal
6,500
6,000
5,500
5,000
4,500
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
Sum of Utility
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined
Method
c
nc
all

Averaged Simulation Plot Examples:

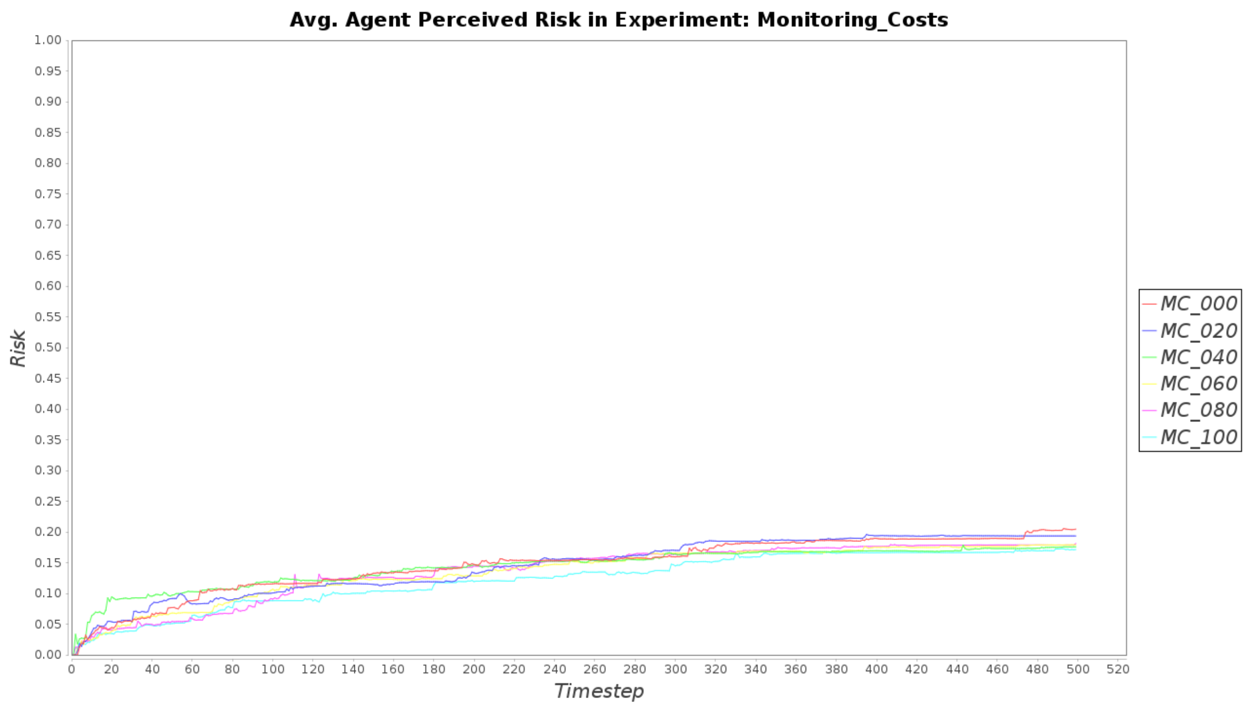




Avg. Agent Propensity to Cheat in Experiment: Graduated_Sanctions
PCheat
Timestep
L_0
L_1
L_2
L_4
L_6
L_10
L_15



Avg. Agent Satisfaction in Experiment: Comparative_Trade_Offs
Satisfaction
Timestep
Base
Principles_4_5
Deterrence
Principles_4_5_6
Combined


Avg. Agent Perceived Risk in Experiment: Monitoring_Costs
Risk
Timestep
MC_000
MC_020
MC_040
MC_060
MC_080
MC_100
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
380
400
420
440
460
480
500
520