# A DESIGN SPACE EXPLORATION METHOD FOR IDENTIFYING EMERGENT BEHAVIOR IN COMPLEX SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Vadim Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctorate of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2016

# A DESIGN SPACE EXPLORATION METHOD FOR IDENTIFYING EMERGENT BEHAVIOR IN COMPLEX SYSTEMS

Approved by:

Prof. Dimitri Mavris
Committee Chair, Advisor
School of Aerospace Engineering
*Georgia Institute of Technology*

Prof. Daniel P. Schrage
School of Aerospace Engineering
*Georgia Institute of Technology*

Prof. Graeme Kennedy
School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Kelly Griendling
School of Aerospace Engineering
*Georgia Institute of Technology*

Dr. Shuo-Ju Chou
RAND Corporation

Date Approved: August 2016

# ACKNOWLEDGEMENTS

I would like to thank everyone that I worked with during my graduate career at Aerospace Systems Design Laboratory. Everyone—from Prof. Dimitri Mavris to the research engineers to my fellow classmates—have been instrumental in my professional and personal growth. The path towards a Ph.D. is arduous but was helped by the constant support from those around me. In addition to the support, I am also grateful for the perspective that Prof. Mavris and ASDL helped to instill in me. I am confident that my new big-picture perspective on aerospace engineering—and life in general—have improved me as a person and have prepared me to take on challenges that are difficult but impactful.

I would like to thank my committee members for their feedback and guidance on my dissertation. Their feedback was valuable and helped shape the final product. In particular, I want to thank Prof. Mavris for his continued support and guidance during the Ph.D. process. I appreciate the freedom that he allowed me to pursue my research interests while challenging me to tackle difficult problems. Most importantly, Prof. Mavris helped me keep the right perspective when working on and presenting my research. His comments helped to make sure that the research stayed relevant and focused. All of my committee members provided insightful feedback that worked towards this goal and undoubtedly helped produce perspicuous research and dissertation.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS OR ABBREVIATIONS

| | |
|---|---|
| **ABM** | Agent-Based Modeling |
| **BTTCP** | Base Target Track Count Percentage |
| **CA** | Cellular Automata |
| **DOE** | Design of Experiments |
| **DSD** | Design Space Divergence |
| **KL** | Kullback-Leibler Divergence |
| **LHS** | Latin Hypercube Sampling |
| **MAS** | Multi-Agent System |
| **MASS** | Multi-Agent Surveillance System |
| **SEED** | Systematic Exploration for Emergence Detection |
| **TTTCP** | Total Target Track Count Percentage |
| **UAV** | Unmanned Aerial Vehicle |

# SUMMARY

This research seeks to gain insight into the design of distributed multi-agent systems. Distributed multi-agent systems present opportunities for accomplishing a goal using multiple simple systems rather than a more complicated monolithic system. Distributed systems, if properly designed, have the potential to exhibit self-organizing behavior which can lead to systems that require less centralized control in addition to improved robustness, reliability, scalability, and adaptability than traditional monolithic, centralized systems.

As engineered systems become more complex, their behavior is more difficult to characterize and predict. Self-organizing systems are difficult to analyze and design since the system behavior is emergent, i.e., the collective behavior only becomes apparent once the system is integrated. The collective behavior is primarily driven by the local interactions of the agents and their environment. This poses an enormous challenge for engineering these systems. The task of system design—selecting the right rules and system parameters—is difficult due to the opaque connection between inputs and responses. The goal of this research is to develop a methodology that provides a way of systematically exploring the design space in order to identify the conditions that give rise to emergent behavior. This information can be used as part of the scientific process of providing feedback through the iterative design process.

In order to address this goal, this research seeks to answer the question on how to define, measure, and use the concept of emergence in the design of a multi-agent system. Similarly, it will address the more general question about how to understand "complex systems" in order to analyze and engineer them. This will be used to guide the development of an appropriate methodology. This research develops the Systematic Exploration for Emergence Detection (SEED) methodology for evaluating computer simulations of complex systems in order to identify conditions that lead to emergent behavior. This research proposes a new quantitative measure of emergence which can identify critical transitions in macro-level performance/function of the system due to changes in system context (i.e., environmental conditions or system parameters). The methodology provides the framework for performing a design space exploration using this measure of emergence to identify critical regions in the design space. These regions help to characterize the design space and will help guide the design process by providing insight into design points where the system behavior is unexpected or changing rapidly, which are possible indicators of emergent behavior.

The SEED methodology is based on a statistical analysis approach. The design space is efficiently sampled using Design of Experiments methods. At each of these design points, the system behavior is characterized statistically using repeated runs of the simulation. The proposed measure of emergence, Design Space Divergence, is then evaluated across the design space and critical regions are identified using data visualization and clustering methods.

A case study is performed on a multi-UAV distributed surveillance problem to investigate whether this framework is capable of identifying emergent behavior. The SEED methodology is used to explore the system design space, including the number of UAVs used in the system and influential vehicle and system parameters. The results show that this methodology provides insights into the landscape of system performance across the design space. More specifically, it identifies a number of candidate designs which exhibit emergent behavior where the system performance rapidly improves as the system undergoes a transition from disorganized to organized behavior. The SEED methodology provides for a more rigorous, traceable, and thorough design process for systems which have been difficult to understand and design using traditional engineering methods.

# CHAPTER I

# INTRODUCTION

## *1.1 Introduction to Multi-Agent Systems*

Decentralized and distributed systems are of great interest to engineers. They present opportunities for accomplishing a goal using multiple simple systems rather than a more complicated monolithic system. Distributed systems, if properly designed, have the potential to exhibit self-organizing behavior which can lead to more robust systems that require less centralized control in addition to improved robustness, reliability, scalability, and adaptability than traditional monolithic, centralized systems [59, 170, 118]. On the other hand, distributed self-organizing systems also have a number of disadvantages including low predictability and understandability, difficulty of control, and difficulty of design and engineering [118]. An important question is deciding whether a distributed multi-agent makes more sense than a monolithic system. If the problem is inherently decentralized and has distributed information availability, the multi-agent approach may be better [92]. Examples of aerospace multi-agent distributed systems in the research literature include multi-static radar network using unmanned aerial vehicles (UAVs) [62], low-altitude short-endurance UAVs for tracking fires [61], a multi-UAV distributed communication system for agricultural monitoring [100], and a multi-agent autonomous system for space exploration [242]. These examples demonstrate reasons for using multi-agent systems such as the presence of a treacherous or adversarial environment where the loss of vehicles is a significant risk (e.g., wild fire tracking) or in situations where communication with a centralized controller is difficult due to bandwidth limitations, power requirements, and communication delays (e.g., deep space exploration).

### 1.1.1 Biologically-Inspired Design

The potential of improving robustness, scalability, and adaptability using multi-agent systems is clearly enticing. A significant amount of the inspiration for multi-agent systems comes from biological systems such as ants, bees, and other social insects. Through the course of evolution, many biological systems have been able to reach remarkable levels of robustness, efficiency, or adaptability. Using these types of systems as inspiration for designing evolvable or robust engineered systems allows us to use the lessons of millions of years of adaptation. Biological inspiration is particularly relevant for distributed multi-agent systems, where cooperative behavior between autonomous agents is desired [135]. One of the enticing aspects of designing systems based on collective behavior is the idea that effective and robust behavior can be achieved with simple agents following simple rules. Natural systems provide proof that systems of simple agents are capable of robustly performing tasks beyond the capability of any single agent [59].

Using social insects societies as a design metaphor, it seems possible to create highly effective systems comprised of relatively simple components [44]. Simple agents that exhibit significant interactions can lead to collective behavior that far exceeds the capability of an agent in isolation. In fact, it is the *interaction* between components that determines the system behavior. As will be discussed later, the interaction between components is what makes the analysis and design of these types of systems so difficult. Research into ant colonies has shown that ants in highly social species (i.e., strongly interacting ants) can be less complex than individual ants from less social species [8]. This suggests that there can be a tradeoff between individual agent complexity and inter-agent interactivity while maintaining system effectiveness. In other words, if we can increase the cooperation between agents then we can decrease the complexity of the individual agents (thereby making them cheaper and easier to design).

Ant-inspired behavior is a canonical example of a distributed multi-agent system and is of interest in many fields [224]. The following quotes from Maier & Rechtin and Nicolis, respectively, summarize why distributed multi-agent systems like foraging ants are so compelling to the scientist and engineer:

> “Ant colonies, for example, exhibit complex and highly organized behaviors that emerge from the interaction of behaviorally simple, nearly identical, sets of components (the ants). The behavioral programming of each individual ant, and its chaotic local interactions with other ants and the environment, is sufficient for complex high-level behaviors to emerge from the colony as a whole. There is considerable interest in using this truly distributed architecture, but traditional top-down, decomposition-oriented models and their bottom-up, integration-oriented complements do not describe it.” [177]

> “What is most striking in many insect societies is the existence of two scales: one at the level of the individual, characterized by a pronounced probabilistic behavior, and another at the level of the society as a whole, where, despite the inefficiency and unpredictability of the individuals, coherent patterns characteristic of the species develop at the scale of an entire colony.” [204, p. 232]

## *1.2 Characterizing Multi-Agent Systems*

The goal of this research is to gain insight into the design of multi-agent systems through the development of a methodology for exploring the landscape of possible behaviors in a multi-agent system (MAS). A multi-agent system consists of a group of agents, where an agent is an autonomous robot capable of performing a task (e.g., manipulating the environment or processing information). A defining feature of multi-agent systems is that the agents are spatially distributed. However, since the agents are spatially distributed, coordination of the agents becomes a fundamentally important aspect of the system design. The first major decision is whether the system has centralized or decentralized control [59].

### 1.2.1 Centralized vs. Decentralized Control

Centralized control relies on a single controller having knowledge of the states of all of the agents and directly influencing their behavior. If each agent is afforded little to no autonomy, a centrally controlled system will resemble a traditional monolithic system with spatially distributed components. Decentralized systems can be divided into either hierarchical or distributed control schemes. In hierarchical decentralized systems, specialized agents exert control influence upon other agents within their zone of influence. In a completely distributed system, the control laws are identical among all of the agents and there is no explicit hierarchy or preference built into the system. This breakdown of multi-agent systems according to control structure is shown in Figure 1.

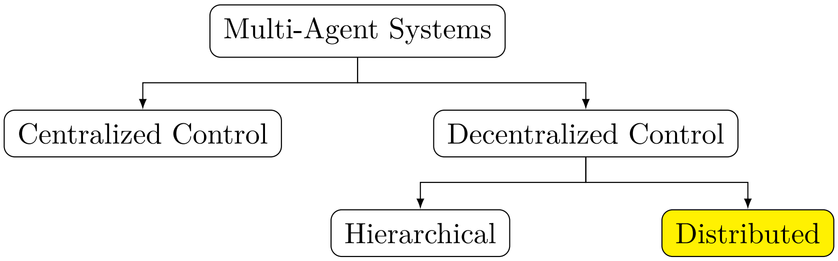

**Figure 1:** Multi-agent systems: centralized versus decentralized

Multi-agent systems with decentralized and distributed control, highlighted in Figure 1, will be the focus of this research. Although this class of problem is the most difficult to analyze and design, it also represents a revolutionary approach for engineering systems. It is clear that if we are to take advantage of multi-agent systems, we must maintain the individual agent's autonomy and leave open degrees of freedom. This autonomy (i.e., open degrees of freedom) provides both the benefits seen in multi-agent systems, but also leaves the possibility of misbehavior and other undesirable effects. However, the benefit of decentralized systems is that they are easier to scale. There is less effort required as the system grows in size compared to a monolithic system with an equal number of subsystems. The control laws are fixed at the agent level, so there is little additional effort required of a system of 100 agents compared to 10 agents.

### 1.2.2 Differentiation

Differentiation refers to the diversity of agent types within a multi-agent system. A homogeneous group is a more straightforward approach and is a common assumption within swarm engineering [59]. A heterogeneous group allows for agent specialization; however, this requires additional effort for the coordination required to take advantage of specialized agents. At the extreme end of the spectrum, a completely heterogeneous group results in a system where each agent can only complete one task, thus defeating one of the main benefits of using swarms. A large degree of agent homogeneity is required to take advantage of redundancy in multi-agent systems [221]. Within the context of this research, agent homogeneity will be assumed.

## *1.3 Self-Organization and Emergence*

The goal of swarm engineering is to achieve a self-organizing system. Self-organization is a “set of a dynamical mechanisms whereby structures appear at the global level of a system from interactions among its lower-level components” where the “rules specifying the interactions among the system’s constituent units are executed on the basis of purely local information, without reference to the global pattern, which is an emergent property of the system rather than a property imposed upon the system by an external ordering influence” [48]. While spatial structure is the easiest to observe, structure can refer to spatial, temporal, or functional structure [91]. The most important feature of self-organization is the lack of external or centralized control [15, 91, 190].

As Cao et al. explain, “the behavior of decentralized systems is often described using such terms as ‘emergence’ and ‘self-organization’ ” [59]. This naturally leads to the research questions: (1) what is emergence? and (2) what is self-organization? Many authors have used the terms emergence and self-organization interchangeably. As Di Marzo Serugendo et al. explain, “emergent properties appear transcending the properties of all the individual sub-units of the system” [97]. However, there are a number of conflicting positions in the literature. While a number of definitions of emergence include the requirement of self-organization, others argue that emergence is possible without self-organization, and vice versa [90, 91, 76]. Within the context of this research, the emergent behavior of interest is the self-organization of the system.

A swarm is a distributed multi-agent system that emphasizes using a large number of robots that are autonomous, relatively simple, and are typically homogeneous [59]. Bonabeau and Théraulaz argue "the potential of swarm intelligence is enormous. It offers an alternative way of designing systems that have traditionally required centralized control and extensive preprogramming. It instead boasts autonomy and self-sufficiency, relying on direct or indirect interactions among simple individual agents" [47]. The goal is for the desired collective behavior to emerge from these interactions between the agents and environment [221]. Self-organization is central to swarm robotics [170], which makes the field very relevant to understanding how self-organization can be engineered. Thus, the objective when designing swarm systems is to understand how to achieve self-organization.

## *1.4 Emergence in Multi-Agent Systems*

Distributed systems with strongly interacting components have the potential to exhibit emergent behavior which is difficult to predict [1, 2]. This emergent behavior can be either beneficial or detrimental to the performance of the system. Emergence can be a mechanism to achieve robustness and improve system capabilities. If subsystems were allowed to be self-organizing or adaptive, a whole new range of functionality may be uncovered. "Because systems with emergent functionality rely on self-organizing processes that require less control, they tend to be not only more adaptive and robust but also cheaper" [208]. However, emergence can also be undesirable and detrimental. In the same way that emergence can lead to new beneficial functionalities, it can also lead to emergent vulnerabilities. Predicting the conditions under which this behavior will occur is challenging. In complex systems, there is an *emergence* of behaviors at higher levels of organization which cannot be predicted at the subsystem level. These unexpected behaviors can be a result of a myriad intricate interdependencies and interactions between components, sensitivity to initial conditions or boundary conditions, enforcement of higher-level constraints on the system, or latent functions or variables in the system. These mechanisms leading to emergence will be discussed in more detail in Chapter 3.1.1. Emergence is a highly debated topic that will be discussed in detail in Chapter 2 and Appendix A. The concept of emergence is well-known and often discussed in biology, sociology, artificial life, solid state physics, and other fields. While scientists from many fields have attempted to study and understand emergence for over a hundred years, the concept of engineering for emergence has taken hold only within the last several decades. This is because it has been difficult up to this point to quantify emergence in a way which makes it a usable concept within engineering design. The fragmented concept of emergence has also resulted in long standing debates about the definition and interpretation. Some conceptions of emergence have resulted in a definition that is essentially "I'll know it when I see it." Clearly this type of definition makes the concept of emergence of little use in engineering. This work will attempt to highlight the issues of defining emergence and will propose a definition of emergence that is useful in engineering design. The goal is to address not only emergence in distributed multi-agent systems, but to propose solutions that can be generally applicable towards many complex systems relevant to engineering. As engineered systems continue to grow in complexity, techniques need to be developed to either take advantage of or inhibit emergent phenomena. But before we can *use* emergence, we have to *identify* it first. As Ryan asks, "is it possible to measure and detect emergence in simulations and in the real world?" [219, p. 177]. In order to help address this question, the first research question (**RQ1**) is posed:

**Research Question (RQ)**:

- **(RQ1)** What is emergence?
  - **(RQ1.1)** What are the characteristics of emergence and what makes it difficult to understand and predict?
  - **(RQ1.2)** How can emergence be defined?
  - **(RQ1.3)** How can emergence be detected or measured?
  - **(RQ1.4)** How can emergence be understood in the context of engineering?

**Research Objective**: The objective of this research is to develop a method for identifying emergent behavior, both beneficial and detrimental, in complex systems.

## *1.5 Multi-Agent Systems as a Complex System*

The scale of engineered systems has followed an increasing trend. Complexity, which often follows with scale, has like-wise increased. Bar-Yam claims that “all systems contain a fundamental tradeoff between complexity and scale” [26]. Initially, engineers designed mechanisms and simple, independent systems. As design, analysis, and manufacturing techniques improved, the capability of designing more complex systems improved. Systems that are more complex provide improved capabilities but are harder to design and analyze. Maier and Rechtin argue that “increasing complexity is at the heart of the most difficult problems facing today’s systems architecting and engineering” [177]. We can analyze and design simple systems (e.g., mechanical clocks) with minimal engineering effort. In simple systems, the interactions between components are minimized or well-defined so that the system can be analyzed individually. In these types of simple systems, there is minimal iteration and testing required. In complex systems (such as distributed multi-agent systems), the *interactions* of the components determine the system behavior to the same degree as (or even greater than) the individual components themselves. With increased complexity, we can still analyze and design much more complicated systems (e.g., cars, aircraft). However, these types of systems have required long design cycle times even with the use of systems engineering techniques. Increased complexity is dealt with using iterative design cycles and extensive testing. The goal of the iterative design process is to converge on a design by identifying the interaction effects caused by each design decision. However, in truly complex systems, the interactions are of the same order of importance as the primary effects of the component. This means that it may not be possible to converge to a design using an iterative approach within the typically small number of iterations during the design process.

While increasing scale (in the form of systems comprising a much greater amount of subsystems) is one source of complexity, complexity can also be a result from increased autonomy of the components of a system (as is the case for multi-agent systems). Engineering design has traditionally attempted to restrict the amount of interactions and autonomy of a subsystem in order to keep the behavior predictable. However, it is clear that as systems continue to increase in complexity, the existing paradigm of minimizing interactions is not sustainable—in fact, it is likely stifling the performance of the system and burdening the design process. Bar-Yam argues that the "fundamental reason for the difficulties with modern large engineering projects is their inherent complexity" [27]. As engineered systems become more complex, their behavior is more difficult to characterize. More critically, complex systems exhibit behaviors that are unexpected during design that only become apparent when the system is integrated and tested. Complexity is caused by the interdependence of various parts and levels within the system. This is problematic because the dominant engineering design paradigm up to this point has been *top-down* subsystem-based design, where components of a system are designed separately and later integrated to build the complete system [99, pp. 212–218]. However, complex systems often cannot be decomposed in such a way to eliminate descriptions of interdependencies [24]. In fact, we can roughly characterize complex systems as those in which the behavior of the system is dominated by interaction effects between components and the environment. There is a strong *context* dependence in which the subsystems cannot be analyzed separately from other components and the environment in which they operate. Informally, systems are cast as "complex" when traditional analysis and engineering techniques have failed to work [219]. This leads to the next research question (**RQ2**), which seeks to answer the question of what makes a system complex and how can we analyze them.

**Research Question (RQ2)**:

- **(RQ2)** How do we analyze and design complex systems?
  - **(RQ2.1)** What characterizes a complex system?
  - **(RQ2.2)** What causes a system to be complex?
  - **(RQ2.3)** How can the complexity of a system be measured?
  - **(RQ2.4)** How should a complex system be analyzed?

The purpose of studying not just distributed multi-agent systems, but the larger topic of complex systems, is to gain insight into the greater problem and to hopefully be able to utilize analysis and design techniques used across the spectrum of complex systems. A number of real world engineered systems have been identified that exhibit complex behavior. Examples include smart energy grids [67], manufacturing systems [247, 248, 95], and road networks [179]. Within the domain of aerospace engineering, air transportation systems and air traffic management have been shown to be complex systems [111, 228, 121, 113]. The dynamics of an airplane moving between airports can exhibit nonlinear, complex behavior [201]. Bouarfa et al. [53] study air transportation system safety using agent-based modeling to identify emergent behavior and lever-points in the system which can improve safety for runway crossing operations. In addition to existing complex systems, there are visions for future systems that truly take advantage of self-organization in order to achieve mission effectiveness. The goal will be to identify common threads in these types of systems that may enable us a greater understanding of how to solve these engineering problems.

### 1.5.1 Importance of Understanding Complex Systems

There are two important reasons for understanding complex systems and developing the appropriate engineering methodology. The first reason is to avoid system failures as a result of unforeseen behaviors. For critical systems, the importance of understanding the behavior and vulnerabilities of complex systems cannot be overstated. In the field of critical infrastructure protection [4], "vulnerabilities resulting from system complexity are expanding at a much faster pace than our means of understanding them" [199]. Not only are the number of possible vulnerabilities increasing, but the nature of the vulnerability is different as well. This is different from how typical vulnerability analysis is viewed—the vulnerability cannot be traced to a single system or component. The vulnerability exists because of the interactions between components. Emergent vulnerabilities are certainly one of the most important reasons for studying emergent behavior in complex systems. For applications like critical infrastructure, the amount of damage a vulnerability can inflict is staggering. Mussington argues that complex adaptive systems modeling approaches are necessary to understand interdependencies between critical infrastructure components leading to emergent vulnerabilities [199]. Although critical vulnerabilities are of course important, there are lesser degrees of undesirable behavior that cause deviation from the desired behavior or reduced effectiveness.

The second reason is that complex systems also have the potential to be more robust, flexible, and effective than monolithic systems. If complex systems are properly understood and designed, there can be improved system performance. Emergent behavior can be beneficial: “emergent complexity is often more robust, flexible and fault-tolerant than programmed, top-down organized complexity” [176]. Therefore, the engineered systems that utilize emergent behavior can benefit from better performance. As was noted earlier, traditional engineering approaches aim to isolate subsystems and minimize and control subsystem interactions. The purpose of this research is to improve system tractability and understanding. As Bar-Yam notes, “conventional engineering places [limitations] on system capabilities,” capabilities that might otherwise “be exclusive to systems that have strong emergent behaviors” [24]. Emergent behavior can “unlock” highly desirable properties and behaviors in systems.

Additionally, one of the major uses of simulation during design is to gain an understanding of the system being studied. Emergent behavior is often characterized as being “surprising.” Understanding surprising behavior is one of the best learning opportunities because it represents a condition that can yield the most amount of information. In the aerospace field (and probably others), it is commonly said that a failure during a test of a system is more useful than a success. A failure provides the opportunity to uncover flaws and remedy them. Similarly, finding the conditions for emergence would be very useful during design. It allows designers to scientifically examine the behavior and to take additional steps to either promote or inhibit its occurrence.

## *1.6 Difficulty of Analysis and Design of Multi-Agent Systems*

One of the biggest difficulties when designing multi-agent systems is understanding how individual agent rules and properties affect the system level behavior [170]. The collective behavior of the agents cannot be predicted from the individual agent’s rules [46]. The fundamental reason for this is due to various mechanisms common to complex systems. Mechanisms such as nonlinearity and chaos tend to separate the causes from the effects in such a way as to make the connection between them very difficult to predict. The lack of analytical solutions necessitates the need for simulation for analysis. In fact, as will be discussed in Chapter 2.3.1 and Appendix A.2.5.2, simulation is *the* fundamental analysis tool for the study of emergence. A bottom-up simulation approach, especially agent-based modeling, is the best way of observing the collective (i.e., emergent) behavior in the system behavior of distributed multi-agent systems [46]. Although defining the rules a priori that yield a particular behavior is difficult (i.e., inverse design), explaining the behavior of a simulation is straightforward since each agent acts according to the rules [208].

### 1.6.1 The Ad Hoc Nature of Design

As Brambilla et al. explains, "unfortunately, in swarm robotics there are still no formal or precise ways to design individual level behaviors that produce the desired collective behavior. The intuition of the human designer is still the main ingredient in the development of swarm robotics systems" [54]. Similarly, Pfeifer and Bongard [208] argue that "design for emergence" lacks a systematic framework and is an art rather than a science. Bonabeau et al. [46] provide two reasons why the design of swarms is ad hoc: (1) there is lack of mathematical detector for collective behavior, and (2) all of the possible behaviors of the system are not known ahead of time. To address the first limitation, this research will identify measures that can be used to quantify the degree of emergence (see **RQ1.3**). The use of a quantitative measure enables the use of automated tools to evaluate designs. In particular, it allows for parameter tuning to identify the conditions for cooperative behavior (i.e., emergence) to take place [59]. The second limitation will be addressed by the formulation of a methodology that enables an thorough exploration of the design space. This provides a characterization of the design space and gives insight into the range of possible behaviors the system is capable of.

**Identified Gap 1**: The design process for distributed multi-agent systems is ad hoc and heavily based on designer intuition. To overcome this problem, a methodology is needed for systematically exploring the design space in order to make the design process more thorough and traceable.

Since traditional engineering methods often fail for designing complex systems and ad hoc methods lack a scientific basis, this naturally leads to the third research question (**RQ3**).

**Research Question (RQ3)**:

- **(RQ3)** What is the appropriate methodology for engineering complex systems?

## *1.7 Developing a Framework for Engineering Emergence*

There is a need for a well-developed methodology for carrying out the exploration of complex systems. Holland explains that "complex adaptive systems are so intricate that there is little hope of a coherent theory without the controlled experiments that a massively parallel computer makes possible. At the same time, in an area this complex, experiments unguided by an appropriate theoretical framework usually amount to little more than 'watching the pot boil' " [142]. High performance computers have allowed for massive explorations of simulations; however, these simulations have to be properly designed in order to provide information effectively and efficiently. The types of uncertainties that are inherent in complex systems are statistically characterized variables (i.e., inherently stochastic) and known-unknowns (i.e., uncertainty that is understood in principle but nearly impossible to track due to combinatorial explosion, chaotic behavior, or context-dependence). Uncertainties pose risks of system failure or degradation (i.e., vulnerabilities) but also may allow for beneficial emergent behavior. The best way to understand these uncertainties is through tradespace exploration (e.g., parameter tuning) and design space characterization.

Unfortunately, the traditional paradigm of the engineering design process does not translate well to complex systems. De Wolf and Holvoet argue that a "fundamental problem is the lack of a step plan that allows to systematically specify desirable macroscopic properties, map them to the behaviour of individual agents, build the system, and verify it to guarantee the required macroscopic properties, i.e. a full life-cycle engineering process or methodology" [92]. Significant parts of the design process needs to be modified to meet the needs of complex systems. In Figure 2, De Wolf and Holvoet [92] show an attempt to modify the traditional engineering design process in order to better handle emergent behavior.

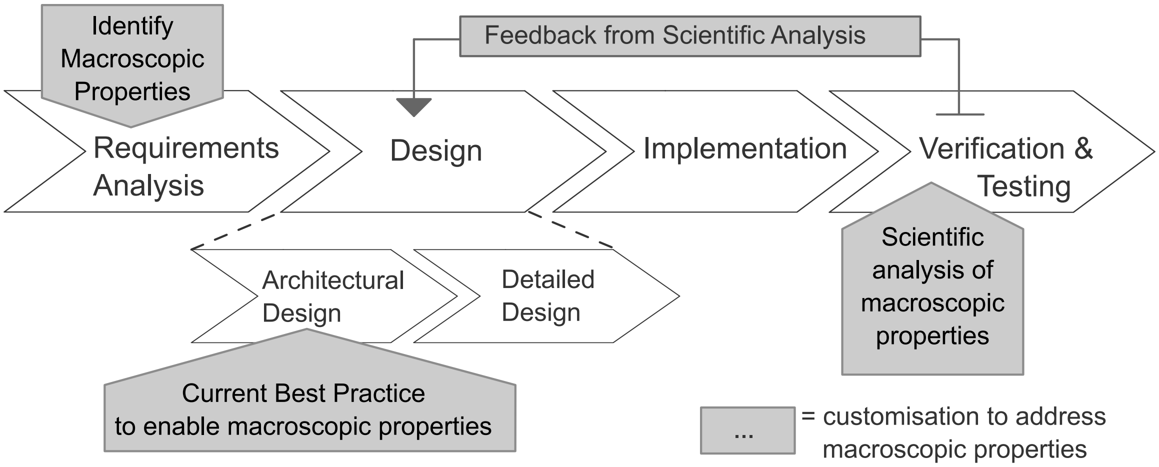
Identify Macroscopic Properties
Feedback from Scientific Analysis
Requirements Analysis
Design
Implementation
Verification & Testing
Architectural Design
Detailed Design
Scientific analysis of macroscopic properties
Current Best Practice to enable macroscopic properties
...
= customisation to address macroscopic properties

**Figure 2:** The traditional engineering design process annotated with modifications for dealing with emergent behavior (from [92])

The most important modification to the traditional engineering process is the addition of feedback loops as a way of introducing mechanisms to properly guide the emergent behavior toward the desired outcome. In the traditional engineering design process, there is exploration of the design space during iteration that is guided by optimization methods. Even though optimization is credited with helping to turn engineering design from art into a science [253, pp. 9–10], as will be discussed in Chapter 4.3.4, optimization is often inappropriate for complex systems. To replace the role that optimization held in the traditional design process, complex systems engineering needs an objective and widely-applicable measure of emergence and a way to explore the design space to identify it.

The path from design to verification and testing is long and difficult. To make matters worse, the design process will require a significant amount of iterations. In Figure 3, De Wolf and Holvoet [92] show how the emphasis on various portions of the scientific analysis feedback loop changes as a function of the iteration. Early iterations focus on setting up the elements of the system so that they are capable of performing a high-level function. Later iterations focus on tuning the system and characterizing the behavior.

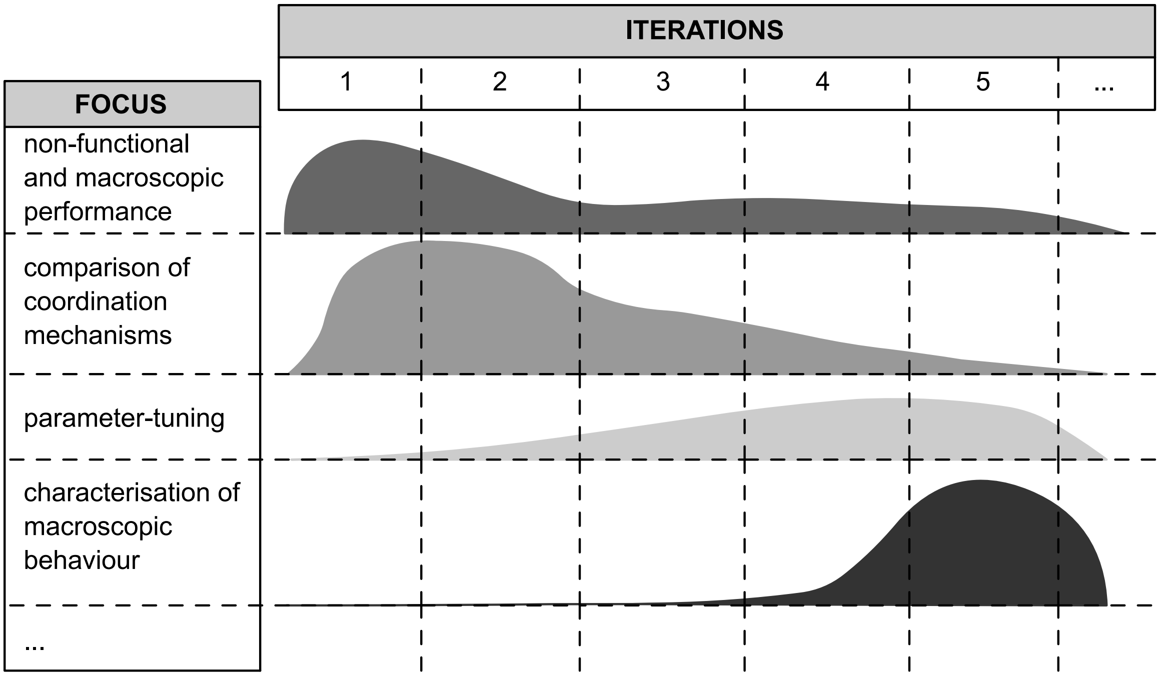
ITERATIONS
1
2
3
4
5
...
FOCUS
non-functional and macroscopic performance
comparison of coordination mechanisms
parameter-tuning
characterisation of macroscopic behaviour
...

**Figure 3:** Different focus in early iterations versus later iterations in the scientific analysis feedback loop (from [92])

To address **Identified Gap 1**, the focus of this research presented in this thesis corresponds with the parameter-tuning and design space characterization steps of the design process. The goal of this research is to provide a methodology for carrying out this portion of the scientific analysis. By improving the effectiveness of this step, the secondary goal is to help shift parameter-tuning and characterization of macroscopic behavior to earlier in the design process. Complex systems can be strongly influenced by these tuning parameters; it is clearly not beneficial to wait until late in the design process before understanding how these parameters affect the system behavior. As Cavallo explains, “design cycles that cannot adapt to rapidly changing conditions miss emergent phenomena that either need correction because they are undesirable, or need capitalization if desirable” [63]. Understanding complex systems and emergent behaviors is one of the major keys to reduce re-design work during development and system failures during deployment. Thus, the overall research goal can be stated as follows:

**Research Goal**: The proposed methodology should be able to: 1) identify design points (i.e., parameter settings) which are candidates for emergent behavior, 2) present the candidate points in a manner which allows the designer to characterize the design space and make proper inferences in coming up with strategies to exploit or avoid behavior. The success of this research will be judged based on the ability of the proposed approach to rigorously and robustly identify the design space parameters which lead to emergent effects.

The successful application of this methodology has the potential to improve system behavior assurance. Behavior assurance is becoming a more important part of engineering complex systems [217, 256]. Wulf, in a National Academy of Engineering report, wrote that "the key point is that we are increasingly building engineered systems that, because of their inherent complexity, have the potential for behaviors that are impossible to predict in advance. Let me stress what I just said. It isn't just hard to predict the behavior of these systems, it isn't just a matter of taking more into account or thinking more deeply—it is impossible to predict all of their behaviors" [261]. He goes on to ask, "How do we ethically engineer when we know this—when we know that systems will have behaviors, some with negative or even catastrophic consequences—but we just don't know what those behaviors will be?" [261]. Alderson and Doyle argue that "we are better at 'trial and error via deployment' than provable guarantees on performance, stability, etc. Moreover, it has perhaps given the false impression that the emergence of collective behavior is sufficient as a design outcome. However, as technological visions increasingly emphasize ubiquitous control, communications, and computing, with systems requiring a high degree of not only autonomy and adaptation, but also evolvability, scalability, and verifiability, a more rigorous, coherent, and reasonably complete mathematical theory underpinning . . . is needed" [4].

To be clear, the methodology proposed in this research is not designed for system certification. This methodology is intended for conceptual design where the goal is gain insight into the design space and the landscape of possible system behaviors. This will enable the downselection of appropriate concepts for further study. Validation and certification of the system will need to take place once the system is embodied. A significant portion of the research on behavior assurance is based on the use of formal methods [217]; however, formal methods are not appropriate for conceptual design since there will be crucial differences between the initial concept and the fielded system. This difference will invalidate the results of the formal methods. Instead, it is more likely that behavior assurance for complex system will be accomplished probabilistically and test via simulation [255]. However, by characterizing the design space early in the design process, there can be more confidence that the concepts chosen represent desirable solutions including initial estimates for the statistical performance of the system.

## *1.8 Research Road Map*

A mapping between the research questions introduced in this Introduction and the chapters within this dissertation that will address these questions is shown in Figure 4. Chapter 2 and Appendix A will examine the concept of emergence and attempt to address the some of the research questions posed by **RQ1**. Chapter 3 examines complex systems and the idea of complexity, including how complexity comes about and ways of measuring it. This chapter addresses **RQ2**. Chapter 4 is a discussion on the approaches taken to analyze and design complex systems. This chapter explains why traditional design approaches are not well suited for multi-agent distributed systems. Chapter 5 reviews many of the approaches that researchers have taken when trying to develop a method for measuring emergence in a system. This chapter addresses a portion of **RQ1** and contains the emergence measure proposed as part of this research. Chapter 6 introduces Distributed Multi-Agent Surveillance Simulation (Distributed-MASS), an example of a distributed multi-agent system that will be used to test the methodology developed in this research. Chapter 7 contains the development of the Systematic Exploration for Emergence Detection (SEED) methodology that is proposed as the solution for exploring the design space of a complex system in order to identify emergent behavior. This chapter addresses **RQ3**. Chapter 8 contains the results from the experiments as part of the methodology development. A case study, based on the Distributed Multi-Agent Surveillance Simulation, is shown in Chapter 9. Finally, Chapter 10 summarizes the results from this research and offers conclusions about the success of this research. Appendix B provides a basic overview of information theory. Information theory has deep and useful connections for the study of complex systems and some of the basic results from information theory are used throughout this research.

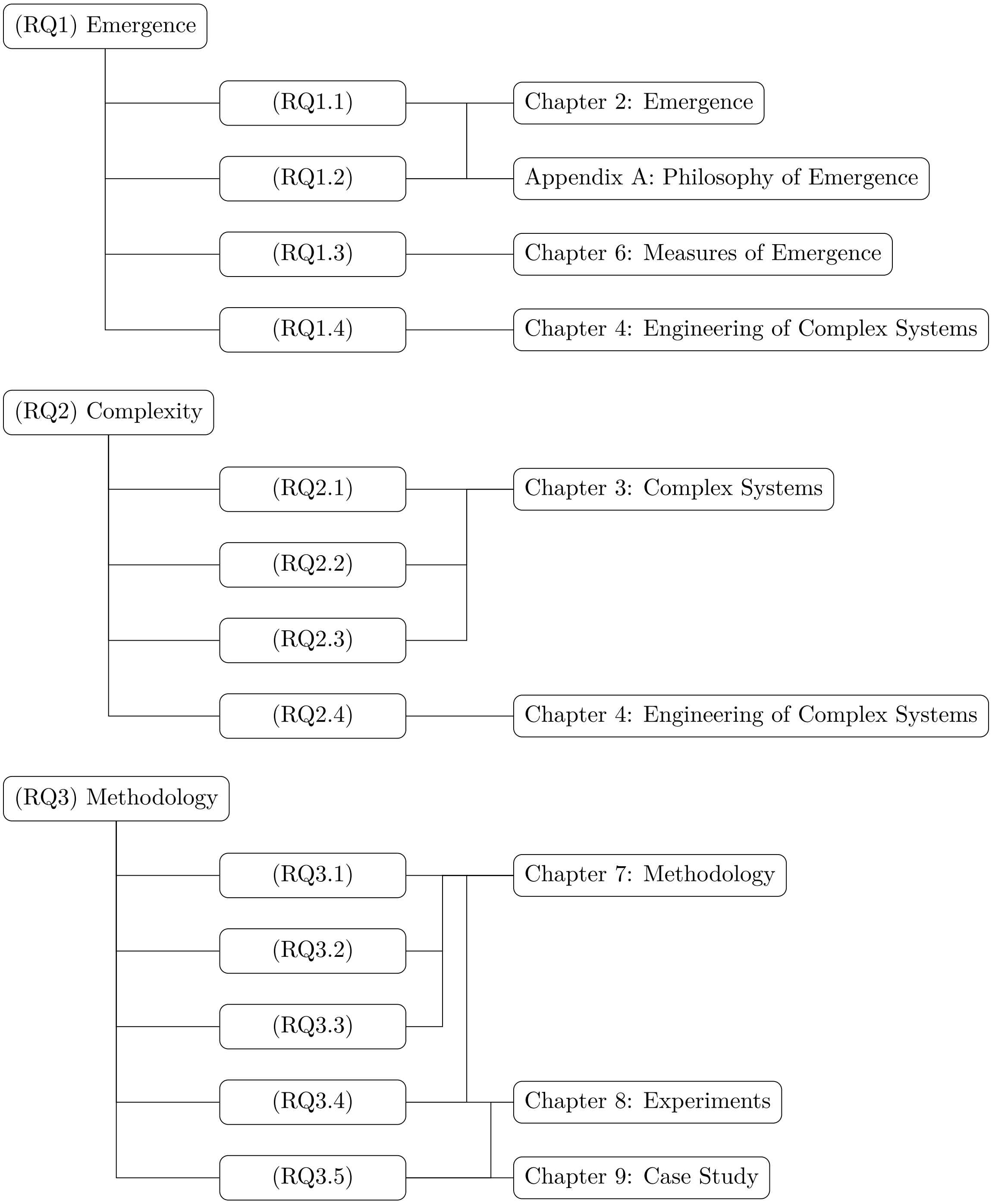

**Figure 4:** Mapping research questions to chapters

# CHAPTER II

# EMERGENCE

**Chapter Road Map**: The goal of this chapter is to define the concept of emergence and to work towards a definition that can be used in this research to identify emergent behavior in a distributed multi-agent system. This chapter addresses research question **RQ1** and related sub-questions:

- **(RQ1.1)** What are the characteristics of emergence and what makes it difficult to understand and predict?
- **(RQ1.2)** How can emergence be defined?

## *2.1 Introduction*

The concept of emergence was introduced in Chapter 1. For distributed multi-agent systems, emergence has an intuitive definition: the collective behavior "emerges" from the interactions between agents and the environment. This collective behavior far exceeds the capabilities of any single agent. As discussed earlier, social insects can be used as inspiration and provide concrete examples of how their underlying mechanisms can give rise to robust, effective, and interesting behavior. However, the concept of emergence is present in fields as diverse as solid state physics, biology, sociology, and economics. In fact, the landscape of definitions and applications of emergence is more diverse than at any time in the history of the concept. Each of these fields has their own understanding of what emergence means. Unfortunately, trying to unify these ideas into a single coherent definition is difficult.

The concept of emergence has a long and contentious history. In fact, the debate about what emergence is and its validity in scientific understanding continues to this day. To deeply understand emergence, it is best to understand the circumstances under which the term has been used, including its historical context. At its historical roots, emergence is a concept deeply rooted in philosophy of science and metaphysics. There are many concepts in philosophy of physics and metaphysics that are useful for understanding what emergence is and why it is a contentious topic. While emergence originated in the philosophy of science in the mid-nineteenth century, it has become common in scientific and engineering literature today. However, in between, emergence has a cyclical history that closely follows the successes and failures of various scientific programs. Unfortunately, some scientific and philosophical failures have left a trail of mischaracterizations that persist to this day. These issues are discussed in detail in Appendix A. Although that discussion is useful to understand the issues in defining emergence, it is not strictly necessary for understanding the results presented in this chapter and within the remainder of the research. It is therefore left for the Appendix for the interested reader.

There are two major goals when it comes to emergence: 1) defining emergence, and 2) quantifying or detecting emergence. The first goal is straight forward—provide a definition that is coherent, specific, and useful. The second goal is dependent upon meeting the first goal. Meeting the second goal of identifying emergence will give us the ability to understand, explain, and perhaps even design for emergence. Detecting emergence can take on a number of different interpretations depending on the application and goal. It could mean testing a candidate behavior to prove that it meets our definition of emergence. It could also be interpreted to mean observing a system or process and identifying whether or not emergence is part of the system's behavior. Or, as is the goal of this thesis, searching through the landscape of possible configurations and conditions of a given system to see if emergence can be identified. At its core, identifying emergence requires a definition that is objective, concrete, and actionable.

While the primary goal of this research is to develop a methodology capable of providing solutions to the distributed multi-agent problem, a secondary goal is to create a methodology (including appropriate definitions for emergence) that can be expanded to a wider range of complex systems. After all, design methods like optimization have reached wide-use because of their applicability to a wide range of problems. So too must emergence eventually reach a point of consistency and acceptance across the range of disciplines if it is to become a useful tool.

## *2.2 Defining Emergence*

Having a definition of emergence is critical for this research. Identification of emergence can only occur once a definition is made. However, the history of emergence has shown that the definition of emergence is inconsistent and sometimes controversial. Part of this is due to history. Emergence started as an empirical phenomenon. Therefore, it could not be defined axiomatically. Emergence was observed and then a definition was created that attempted to capture the characteristics of the observation. A universal definition of emergence may not be possible (although we should continue to try until we have evidence that emergence escapes a rigorous definition). A universal theory of emergence must be consistent with the underlying causality. Such a theory should be metaphysically rigorous and logically coherent—but it might be a long time before such a theory exists. In the mean time, a definition that simply captures the observed behavior might be sufficient for engineering design. Presented below is a small sampling of the definitions from the literature:

- **Assad and Packard** (1992): Coming from the perspective of artificial life, Assad and Packard's definition reflects the field's characteristic approach of using simple rules to exhibit surprisingly complex behavior. In their definition (which is based on a characterization of the body of literature in their field rather than a prescriptive definition), emergence can be described as "unexpected macroscopic behavior that is not immediately predictable upon inspection of the specification of the system" [12]. They acknowledge the subjectivity in this definition. The characterization of emergence using terms like surprise or unexpectedness is troublesome. This language implies the subjectivity of the observer or the epistemological nature of emergence. The trouble with using a subjective definition, as well as the general misunderstanding of this approach, will be discussed in detail later in this chapter. Assad and Packard also attempted to introduce a spectrum of emergence depending on the difficulty level on the deducibility of the emergent behavior. They proposed a scale from "weak" emergence that is simple to deduce in hindsight, up to “strong” emergence that is impossible to deduce from the specification of lower levels [12]. The final contribution from Assad and Packard is the distinction between three different types of emergence: *structure* (emergence of patterned structure), *computation* (emergence of novel computational processing capabilities), and *functionality* (emergence of novel functions that are used by the micro-level components) [12]. Furthermore, they hypothesize that these types of emergence are hierarchical where the first type of emergence is required before the subsequent type can take place, thus, functionality emergence requires emergence of computation which requires emergence of structure.

- **Bonabeau** (1995): “emergence is a dynamic process through which some quantity [complexity] is rapidly/dramatically varying with respect to the time constant/the spatial granularity, or more generally to the model or the level of description used by the observer.” [45].

- **Chalmers** (1996): “an interesting property that is unexpected, given the underlying principles governing the system” [64, p. 253].

- **Crutchfield** (1994): “Emergence is generally understood to be a process that leads to the appearance of structure not directly described by the defining constraints and instantaneous forces that control a system. Over time ‘something new’ appears at scales not directly specified by the equations of motion. An emergent feature also cannot be explicitly represented in the initial and boundary conditions” [82].

- **Wimsatt** (1997): “An emergent property is—roughly—a system property which is dependent upon the mode of organization of the system’s parts” [254].

- **Holland** (1998): “Emergence is above all a product of coupled, context-dependent interactions. Technically these interactions, and the resulting system, are *nonlinear*: The behavior of the overall system cannot be obtained by *summing* the behaviors of its constituent parts.” [143, p. 121–122]

- **Goldstein** (1999): Emergence is “the arising of novel and coherent structures, patterns, and properties during the process of self-organization in complex systems" [128, p. 49]. Emergence has the properties of being or having: 1) radical novelty, 2) coherence or correlation, 3) global or macro level, 4) dynamical, and are ostensive [128].

- **Kim** (1999): Emergent properties have two main characteristics: 1) "emergent properties are 'novel' and 'unpredictable' from knowledge of their lower-level bases, and that they are not 'explainable' or 'mechanistically reducible' in terms of their underlying properties;" b) emergents have "novel causal powers irreducible to the causal powers of their basal constituents." These causal powers "influence and control the direction of the lower-level processes from which they emerge." They are novel in that they "did not exist before its emergence" [158].

- **Ryan** (2007): "Emergence is the process whereby the assembly, breakdown or restructuring of a system results in one or more novel emergent properties. Assembly and breakdown are the dual processes of adding and removing interactions between system components that change the cardinality of the set of components in the system, while restructuring changes interactions between components without changing the cardinality" [220, p. 73].

There are a number of common themes in the above definitions: nonlinearity, irreducibility, novelty, and hierarchy. Each of these topics is discussed at length in Appendix A.2. These terms will be briefly summarized. Nonlinearity generally refers to small inputs to a system causing disproportionately large responses or a non-additivity when coming effects. Irreducibility is the inability to describe a property or behavior using properties or behaviors from the constituent parts. Novelty refers to the creation of new properties or behaviors that were not present in the past or under different conditions. Hierarchy refers to either the creation of physical organization and connectedness of the constituent parts or to the levels of description that describe the properties or behavior.

## *2.3 Emergence in Simulation*

In the beginning, the concept of emergence was studied in physical and biological systems. While emergence in natural systems is an active field of research, much of the current research is on the study of emergent properties within computational models. The purpose of this section is help establish a foundation of part of this research. The exploration of computer models for the purpose of understanding emergence is well established. This section will review a few concepts with respect to emergence within computational models.

### 2.3.1 Computational Emergence

Irreducibility, the idea that a property cannot be deduced from the properties of its constituent parts, takes on a more concrete notion when dealing with simulation in the form of *computational irreducibility.* A system is considered computationally irreducibility if the outcome of its evolution can only be found through direct simulation [258]. There is no shortcut in deriving the output short of carrying out the computational steps that govern the underlying dynamics. Wolfram claims that "many complex or chaotic dynamical systems are expected to be computationally irreducible, and their behavior effectively found only by explicit simulation" [258]. In the same way that irreducibility was one of the defining characteristics of emergence, we can extend that concept and say that simulations of complex systems are characterized by computational irreducibility. Darley's definition of emergence is based on this concept of computational irreducibility: "emergent phenomena are those for which the amount of computation necessary for prediction from an optimal set of rules, classifications and analysis, even derived from an idealised perfect understanding, can never improve upon the amount of computation necessary to simulate the system directly from our knowledge of the rules of its interactions" [86].

### 2.3.2 Artificial Life

The study of emergence in the computational domain is well-established, especially in the field of *artificial life.* Artificial life is particularly focused on emergence that comes about through self-generated complexity—the repeated application of a simple set of rules that leads to complex behavior which mimics behaviors observed in nature. In his well-known paper, Reynolds [214] showed how simple rules for agents can demonstrate flocking behavior. Langton is credited [31] with coining the phrase "artificial life" and setting the foundation for the field. He defined the key features of artificial life as: 1) populations of simple programs or specifications, 2) no central controller, 3) rules describing how the simple entity reacts to local situations in its environment, 4) no rules dictating global behavior, 5) behavior at levels higher than the individual programs are automatically emergent [166].

### 2.3.3 Cellular Automata

Cellular automata (CA) is a dynamic system consisting of a lattice of discrete, deterministic machines [150]. Each cell's state is governed by a rule-set which governs its transition behavior. These rules are a function of the states of the neighboring cells, resulting in macro-level behavior that is a due to micro-level rules. CA are computationally irreducible, which, along with the micro- to macro-level dynamics, makes them a candidate for exhibiting emergent behavior. Cellular automata are widely studied in the field of complexity science [33, 150]. They were introduced by von Neumann to study the behavior of extended complex systems. This approach helped to form the ideas and methods of Artificial Life. Additionally, CA can be used as discrete approximations to a set of partial differential equations, allowing them to model many physical phenomena [258]. Of particular interest to the aerospace field is the use of CA to model fluid flows in the form of lattice gas models. Ilachinksi [150, p. 18] outlines variations of CA that allow for stochasticity, non-homogeneity, mobility of cells, and even structurally dynamic lattices; however, once more than one of the key characteristics of CAs are modified or removed, it can be argued that the system more closely resembles another modeling technique, such as agent-based modeling. In fact, cellular automata can be considered as part of the roots of agent-based modeling [109, p. 306], another widely used modeling and simulation technique in the study of complex systems.

As Dogaru explains, "complex emergent behaviors in cellular automata are rare, and consequently difficult to locate" [101, p. 70]. This difficulty has lead to the large amount of research to identify the principles and rules which lead reliably lead to emergence. Kauffman found that there is an optimal (intermediate) degree of connectivity that leads to emergence and Chua found that an array of Boolean cells must be not linearly separable to lead to emergence [101].

#### *2.3.3.1 Conway's Game of Life*

Conway's Game of Life (GOL) [120] is an application of cellular automata. It is one of the most studied examples of emergence using cellular automata. The Game of Life is based on a cellular automata in which each cell follows a simple rule at each iteration: a living cell remains alive if and only if either two or three of its neighbors are alive; a dead cell becomes alive if and only if three of its neighbors were alive at the previous iteration. The resulting dynamics exhibit a wide range of complex behavior, including the generation of structures which move cohesively through the domain ("gliders"). The glider and other structures and properties in the Game of Life are recognized by many researchers as being an example of emergence [165, 33, 216, 101]. As Holland explains, "the possibility of such a spatially coherent moving pattern is not something easily determined by direct inspection of the laws of Conway's universe. The possibility only exists because of the strongly nonlinear interactions of the particles (states) in adjacent cells" [143, p. 140]. The macro-level laws "comes from our prior empirical observations of how the systems behave under different initial conditions" [33].

The Game of Life also has a number of interesting properties that make it an attractive system of study and as an archetype of emergent systems. The GOL is computationally irreducible. The R-pentomino, a five-cell edge-connected unstable pattern, has been shown to be underivable [34]. Additionally, it has been shown that the GOL can be designed to function as a Universal Turing Machine [40], and is therefore capable of computing any algorithm, at least in principle.

## *2.4 Characterizing Emergence in Complex Systems*

The discussion in the previous section provides insights but not any practical solutions for defining emergence. Although we expect our simulation to exhibit computational irreducibility, it is not a sufficient condition for emergence. As a counter example, a simple chaotic system (e.g., a double pendulum) would be computationally irreducible but would not meet our intuitive understanding of emergence. Beni [37] defines an intelligent swarm as a group of non-intelligent machines capable of universal material computation (i.e., the ability to arbitrarily transform matter, energy, or information). What Beni calls "unpredictability" is the concept of computational irreducibility, whereby there is no prediction possible that is more efficient than simulation. In addition to the already mentioned problem that computational irreducibility is not a sufficient condition for emergence, a proof of irreducibility would be very difficult to achieve. The best case scenario is a proof by counter-example; however, even a counter-example would not provide much insight into system behavior.

Other investigations into emergence in simulations like artificial life and cellular automata have focused almost exclusively on pattern formation. While pattern formation is eye-catching, it is unlikely to be a useful paradigm for engineering design. Pattern-formation and investigations of complexity using analysis of structure will be discussed in Chapter 3; however, patterns and structure are not a generally applicable and useful approach for designing engineering systems. Although there will often be some correlations between the degree of organization and system effectiveness, the connection is not clear. Using foraging ants as an example, completely disorganized and random paths by all of the agents is a clear sign that the system has not achieved self-organized behavior. However, at the other end of the spectrum, a perfectly fixed structure is not ideal either. The robustness of ant foraging allows for paths to dynamically change in response to changes in the environment and due to the natural exploration behavior of the ants. Another example is an ant mill, which is a completely degenerative behavior in which ants form a closed loop which they traverse until their exhaustion and death. It should be clear that analysis of patterns is not the right approach; instead, we need to examine the function of the system as a measure of the effectiveness.

**Claim**: Direct measures of system effectiveness are a better way of comparing system behavior rather than indirect methods such as pattern analysis.

Characterizations of complex behavior are naturally probabilistic. Despite the added complications of representing distributions, a probabilistic viewpoint is also useful and relatively well behaved in the sense that a probability distribution is bounded. Nicolis and Prigogine explain how "the problem of instability and multiplicity of solutions, familiar from the study of bifurcations in nonlinear dynamical systems, simply does not arise at the probabilistic level. ... Bifurcation is reflected by a qualitative change in the structure of the underlying probability distribution, such as the appearance of multiple humps, rather than by the multiplicity of the probability distribution itself" [204, p. 162]. Kernstine et al. found that the design spaces of complex systems are typically characterized as stochastic, heteroscedastic, and conditionally variant [157, p. 2]. Kernstine gives three characteristics of emergence: nonlinearities, changes in variance, and discrete changes in the metric [156, p. 245]. All three of these characteristics are easily generalized when dealing with probability distributions: discrete changes are manifested as either rapid shifts in the probability distribution or the existence of a multimodal distribution, where there is more than one peak in distribution. An example of a probability distribution changing as a function of a parameter (in this case time, but generalizable to any parameter space) is shown in Figure 5 (a) while a bimodal distribution is shown in Figure 5 (b). Nonlinearities are manifested as rapidly changing probability distributions for small changes in the parameter space. Kernstine writes that "neighboring points in a simulation [design space] are expected to have similar features, and transitions from one distribution or state are expected to be smooth far from emergent behaviors" [156, p. 83]. While Kernstine argues that "areas of high variance may be indicative of an emergent behavior" [156, p. 178], I do not believe that high variance is necessarily an indicator for emergence. Typically, the emergence of self-organization is manifested by *reduced* variance. Therefore, all variance shifts should be regarded as possible indicators of emergence.

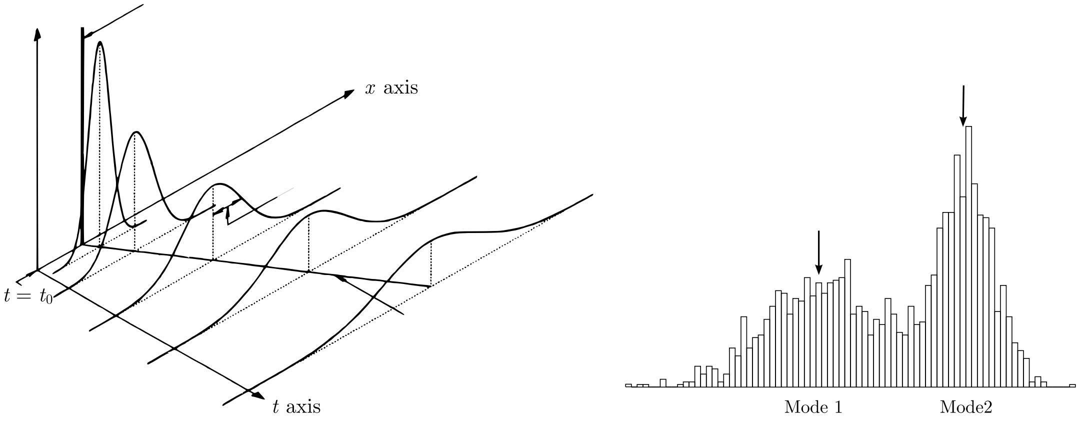

**Figure 5:** (a) Probability distribution shifts in both time and parameter space; (b) Bimodal probability distribution (both from [109])

**Claim**: Emergence is manifested by qualitatively different probability distributions compared to non-emergent design points. Both mean and variance *shifts*, relative to neighboring design points, are important in identifying emergent behavior. Emergent behavior can be identified by locating shifts in the shape of a metric's probability distribution.

## *2.5 Synthesizing a Definition of Emergence*

Part of the difficulty of defining emergence is that there are really two related tasks. The first task is to define emergence with respect to the various types of systems that exist. The second task is defining emergence for a specific system. While the former task seeks to classify the wide range of possible systems, the latter is focused on the specific states in a single given system that are associated with emergence. Since this research is looking at the design of a single system under investigation, the definition presented here is within the context of identifying system states associated with emergence. Although some parts of the definition attempt to capture entire classes of systems, other parts of the definition are specific to a system's states. This difference between the two tasks is the reason why the presented definition will have multiple components.

There is no shortage of definitions of emergence in the literature. While I disfavor an addition to the list of definitions, the lack of a clear, comprehensive, and accepted definition necessitates the creation of a refined definition. The definition of emergence I propose will be synthesized from the ideas presented in this chapter. The goal is to capture the following properties of emergence:

- *Novelty*: The instantiation of new, unexpected properties and *qualitative change* in behavior.
- *Context dependence*: The system exhibits qualitatively different behavior depending on the context (e.g., inputs, the environment, memory effects).
- *Irreducibility*: Borrowing from Bedau [34], Darley [86], and Huneman [148], emergence requires discovery through simulation due to its computational irreducibility.

I propose the following definition of emergence:

**Emergence**: Emergence is the phenomenon in a complex system that is characterized by unexpected qualitative changes in macro-level behavior due to context-dependence of the micro-level components.

However, it is not clear how this definition could be used in a quantitative and useful way. What is lacking is the engineering perspective: a definition of emergence that also includes a concept of affecting change. From an engineer's perspective, emergence is only useful if it helps us to achieve a function. Therefore, a second related definition will be proposed. As will be discussed in the Chapter 5, the idea of a critical transition is common in many measures of emergence. As Fromm explains, using "the colloquial meaning [of] 'appearance' in the broad sense, the sudden emergence of something is always possible at a clear boundary or border of something, someone, or some form of system" [118]. Therefore, I introduce a secondary definition for *engineered emergence:*

**Engineered Emergence**: Engineered emergence is the critical transition in macro-level behavior due to changes in system context (i.e., environmental conditions or system parameters).

This second definition is actionable since unexpected qualitative changes, which is a nebulous concept, is replaced with a critical transition, which can be measured directly.

## *2.6 Emergence for a Distributed Multi-Agent System*

The final section in this chapter will examine whether the proposed definition of emergence is consistent with the concept of emergence for distributed multi-agent systems. More generally, the question is: what does emergence look like for distributed multi-agent systems? Again, we can use social insects to help understand the collective behavior of multi-agent systems. Characterization of foraging ant species has shown that there is a nonlinear phase transition between organized and disorganized behavior upon reaching a critical colony size [36]. Similarly, Bonabeau et al. argue that a characteristic signature of self-organization is "the existence of bifurcations when some parameters are varied: the behavior of a self-organized system changes dramatically at bifurcations" [48]. This idea of a phase transition is consistent with the "critical transition in macro-level behavior" in the proposed definition for engineered emergence. Furthermore, it has been demonstrated in ant foraging models that the system is sensitive to fairly small changes in the properties of the individual ant [105]. This idea is consistent with the second half of the engineered emergence definition where we are searching for shifts in behavior as system parameters change. Thus, emergence in the distributed multi-agent system is going to consist in searching the design space to identify the critical transitions in behavior.

# CHAPTER III

# COMPLEX SYSTEMS

**Chapter Road Map**: The goal of this chapter is to define what it means for a system to be "complex." This chapter will also explore the mechanisms that drive complex behavior and various measures that have been proposed to measure the amount of complexity in a system. Although complexity and emergence are not equivalent, there are enough similarities and overlap in features that warrant the discussion about complex systems in this research. This chapter addresses most portions of Research Question 2 (**RQ2**):

- **(RQ2)** How do we analyze and design complex systems?
  - **(RQ2.1)** What characterizes a complex system?
  - **(RQ2.2)** What causes a system to be complex?
  - **(RQ2.3)** How can the complexity of a system be measured?

## *3.1 Characterizing Complex Systems*

It has been argued in Chapter 1 that distributed multi-agent systems are an example of a complex system. The goal of this chapter will be to understand what causes a system to be complex and to gain insight into how complex systems should be analyzed and designed. The simplest definition of a complex system is given by Crutchfield, a "highly-structured collective behavior emerges over time from the interaction of simple subsystems" [80]. It is this juxtaposition of simple components with "complex" behavior that make complex systems such a compelling topic of study. Weaver [250] proposed three classes of problems: simple, disorganized complexity, and organized complexity. Simple problems, such as in classical mechanics, have a deterministic solution. At the other end of the spectrum, is disorganized complexity; however, this class of problems has a probabilistic solution (e.g., classical thermodynamics). The hardest set of problems is organized complexity. These problems are too big to be solved using traditional analysis methods but they are too small to be solved using strictly probabilistic methods. These problems involve both randomness *and* structure. Organized complexity is observed in a range of systems. These include many physico-chemical systems that exhibit phase transitions or the formation of patterns. Moving up in scale, the Earth's weather and climate system exhibits many characteristics of complexity [204, pp. 226–232]. Biological systems, from biological subsystems like the brain [244, Ch. 5] and the immune system to social insects [204, pp. 232–238] to human systems, are the epitomic examples of organized complexity. One of the issues raised by multi-agent systems is the size of the system under investigation. There are extensive techniques available for analyzing and designing small monolithic systems. At the other end of the spectrum, statistical techniques can be used for characterizing very large systems. However, systems of intermediate size are proving to be the most difficult due to the lack of methods for analyzing and designing these systems.

One way of understanding complex systems is contrasting them with simple systems. Erdi lists three important characteristics of simple systems: 1) single cause and single effect; 2) a small change in the cause implies a small change in the effects; and 3) predictability [109, p. 6]. These are directly opposed to characteristics of complex systems: 1) circular causality, feedback loops, 2) small change in the cause implies dramatic effects, 3) emergence and unpredictability [109, p. 7]. Prokopenko et al. list other important characteristics of complex systems: 1) complex systems are *open* to the transfer of energy, information, and/or matter from the environment; 2) a large ensemble of individual components interact in a nontrivial fashion; and 3) the nontrivial interactions result in internal constraints which results in coherent global behavior [212]. Complex systems are intimately tied with emergence. In fact, many have defined a complex system as a system that exhibits emergence. Complex systems "typically have a large number of components, where the interactions lead to collective emergent behaviours that cannot, even qualitatively, be derived as a plain resultant from the individual components' behavior" [222, p. 248]. This is in contrast to *complicated* systems that "have a large number of components which behave in a well-understood way and have well-defined roles leading to the resulting effect" [222, p. 248]. Nicolis and Prigogine claim that the "fingerprint of complexity" is the "emergence of a *new level of description* brought out by the underlying dynamics" [204, p. 191, original emphasis]. In this way, the characterization of complex systems follows closely with that of emergence.

One important question, similar to the question of emergence, is whether or not complexity is a fundamental property of a system/process or whether it is a reflection of our ability to understand it. While the complexity of physical systems is distinct from computational complexity, they also share some similarities that may help to understand properties of emergence. Moore and Mertens explain that, in the field of computational complexity theory, the complexity class is a "fundamental property of a problem" and not a "subjective question about our abilities to compute" [191, p. 29]. Likewise, emergence and complexity of physical systems are intrinsic properties.

### 3.1.1 Mechanisms of Complex Behavior

We seek to understand how basic low-level interactions can give rise to complex behaviors. Huygen's pendulum clock is an interesting example of a relatively simple system that still exhibits complex behavior. Wimsatt describes Huygen's discovery in 1656 of the coupled oscillation in pendulum clocks as an example of an emergent effect: "clocks hung together on a beam became synchronized and kept better time than either did alone" [254]. This effect was due to the arrangement of the pendulum clocks with a connecting rod between them that provided the mechanism for the coupled dynamics. While this would seem to be a fairly simple system, it was profound for several reasons. The first is the unexpectedness of the result. Huygens was eventually able to determine the mechanism for the coupling effect as "the imperceptible movements" of the connecting beam [39, p. 565]. This stark difference in scale between the large motions of the pendulums and the "imperceptible" motion of the connecting beam is characteristic of many complex systems. The system is also governed by nonlinear dynamics that escape easy characterization, especially at the time of its discovery. The second reason for the importance of this observed example of emergence is the importance of the nonseparability of the system. While many current discussions of complex systems deal with large, engineered systems, Nicolis and Prigogine explained in their book Exploring Complexity: An Introduction (1989) [204] how relatively simple physico-chemical systems can also exhibit complex behavior. There are a number of characteristic mechanisms that are associated with complex behavior. The following is a list of some of the most common and their description.

#### *3.1.1.1* Nonlinearity

In a linear system, the property of superposition holds. The combined effect of two causes is the superposition of the causes taken individually. Systems that are either linear or can be approximated by a linear system are of great interest to scientists and engineers because of this simplification. A system which can be recast as a summation of a number of simpler systems can often be solved by the summation of the solutions to each of the simpler systems. In the case of a nonlinear system, this simplification is not possible. Furthermore, the effects are not proportional to the causes—a small change in the cause leads to a disproportionately large change in the effect. Amaral and Ottino [6] argue that nonlinearity is the key to novel behaviors in complex systems since nonlinear effects drive the system to qualitatively new operating regimes.

#### *3.1.1.2* Chaos

Chaos is a phenomenon where the behavior of the system exhibits extreme sensitivity to initial conditions. Two trajectories of a system will eventually diverge, even for infinitesimal differences in initial conditions. The unique property of chaotic systems is that fluctuations at the microlevel are amplified by the dynamics and affect the macrolevel behavior. This is different from most other systems, where microlevel fluctuations are local effects only (e.g., a gas system). This makes analyzing and designing systems which exhibit chaotic motion difficult, since their state at any future point in time cannot be predicted with any certainty more than an ensemble of possible trajectories. This requires that analysis and design of chaotic systems has to rely on statistical and geometric properties rather than on detailed prediction [79].

Chaos is an interesting topic to study because it is a mechanism which leads to complex behavior, but it also exhibits many of the properties of emergence by itself. Crutchfield et al. describe chaos as "the interaction of components on one scale can lead to complex global behavior on a larger scale that in general cannot be deduced from knowledge of the individual components" [79]. This is the same language of emergence. Chaos also faces the philosophical issue of whether the lack of predictability comes from an intrinsic property of the system or from limitations of the observer. Crutchfield et al. claim it is a "product of both the complicated behavior of nonlinear dynamical systems and the limitations of the observer" [79]. However, I would place more emphasis on unpredictability being a fundamental property of the system. When prediction requires an infinite amount of information (i.e., infinite precision on the state variables), the limitations on the observer are fundamental and unchangeable.

The types of systems that are considered chaotic has increased greatly. Initially, only small systems which could be analytically studied were found to be chaotic. In 1963, Lorenz discovered chaotic behavior in a low-dimensional model of fluid flow describing the weather system [204, pp. 124]. While simple systems could be analytically studied, larger systems had to be studied using computer simulations [234]. Since then, information systems [181], manufacturing, combat systems [174], and others have been shown to be chaotic.

#### *3.1.1.3* Feedback

Feedback is a common mechanism in both natural and engineered systems. Feedback occurs when the system's effect influences the action of the cause. This process leads to complexity because causation is iterative—both causes and effects are linked through a circular connectivity. Since the concept of feedback is well-known, in both regular and complex systems analysis, this mechanism will not elaborated further in this section.

#### *3.1.1.4* Symmetry-breaking and Bifurcations

Symmetry-breaking is the phenomenon where there exists an *intrinsic differentiation* between different parts of a system, or between the system and its environment where there was none before [204, p. 74]. Anderson describes symmetry-breaking as the "shift from quantitative to qualitative differentiation" [9]. Phase transitions in materials are common examples of broken symmetry phenomena. Superconductivity, antiferromagnets, ferroelectrics, liquid crystals are all material phenomena that are governed by symmetry-breaking mechanisms [9]. Nicolis considers symmetry-breaking and bifurcations to be one of the prerequisites to complex behavior [204, pp. 73–74]. Symmetry-breaking also has deep connections to the concept of *information*; symmetry-breaking is a prerequisite to information and information processing [204, p. 143]. It is the selection between possible states that is the basis for information; symmetry-breaking is the selection process.

#### *3.1.1.5* Large Differences in Spatiotemporal Scales

There is an order of magnitude (or more) difference in the spatiotemporal scale between the lowest and highest levels within the hierarchical structure of complex systems. The fundamental "unit" of a system is orders of magnitude smaller than the overall system or the highest-level at which coherent behavior occurs; however, the unit is important to the overall behavior of the system and cannot be abstracted away. This effect is often seen in chaotic systems, where small variations at the microlevel are amplified by the dynamics and bring about changes at the macrolevel. The connection between the micro and macro levels vary greatly. Some systems can be represented using continuum models where the micro level are almost completely abstracted. As Batterman explains, "continuum model equations such as the Navier-Stokes equations of hydrodynamics or the equations for elastic solids work despite the fact that they completely (actually, almost completely) ignore small scale or atomistic details of various fluids" [30, p. 256]. However, it should be noted that it is often the case that continuum models are not derived from micro-level models [30, p. 271]. Other systems exhibit some regularity or statistical similarity that allow for "coarse-graining" procedures where microlevel effects are replaced with statistical averages. "Much philosophical confusion about reduction, emergence, atomism, and antirealism follows from the absolute choice between bottom-up and top-down modeling that the tyranny of scales apparently forces upon us" [30, p. 257].

#### *3.1.1.6* Large Number of Components

Complex behavior is often attributed to systems that comprise many elements. Physicochemical systems have an astronomical number of elements. Similarly, biological systems and sociological systems also contain a large number of interacting entities. This characteristic is inherently related to systems that exhibit large differences in spatiotemporal scales; therefore, the treatment of these types of problems is often through statistical averaging [204, p. 66].

#### *3.1.1.7* Self-generated Complexity

Self-generated complexity comes about from the (infinite) iteration of a few finite rules [17, pp. 9, 249]. Phelan argues that "one of the defining characteristics of complexity research" is the search "for the simple rules that purportedly explain the behavior of complex systems" [209, p. 239]. Some examples of self-generated complexity include fractal geometry, symmetry breaking of superconductors, long-ranged correlations of phase transitions, and Conway's Game of Life. This mechanism is similar to that seen in nonlinear dynamics, where the repeated application of a simple map leads to chaotic behavior.

#### *3.1.1.8* Adaptation or Evolution

Evolution and adaptation can be considered as a special case of a feedback process [76], where the system or the components of the system adapt (i.e., change their behavior) in response to their environment or other components in the system in order to maximize their fitness. Evolution is adaptation that takes place over successive generations of agents, whose fitness improves over time.

If the dominant mechanism driving complex behavior is adaptation, the resulting system can be classified as a Complex Adaptive Systems (CAS). CAS are class of complex systems in which a large number of less complex agents work together to produce coherent high level behaviors. More importantly, these systems have three key characteristics: evolution, aggregate behavior, and anticipation (internal models). Anticipation ("basing current actions on expected outcomes" [142]) is the defining feature of adaptive systems and what separates them from other complex systems. Holland [144] gives four properties of complex adaptive systems: 1) aggregation, 2) nonlinearity, 3) flows, and 4) diversity. The nonlinearity property makes possible “lever points” where “a small input can produce major predictable, directed changes—an amplifier effect” [144, p. 5]. Additionally, there are three mechanisms which facilitate the coordinated behavior: 1) tagging, 2) internal models, and 3) building blocks [144]. The tagging mechanism “consistently facilitates the formation of aggregates” [144, p. 12]. This aggregation gives rise the hierarchical structure. The property of diversity and the internal model mechanism allows for adaptation. “The three essential functions for an adaptive mechanism are generating variety, observing feedback from interactions with the environment, and selection to reinforce some interactions and inhibit others” [212, p. 23].

#### *3.1.1.9* Large Number of Dimensions

As the number of characteristic dimensions of the system increases, there is the possibility of new behaviors that the additional degrees of freedom allow [204, pp. 82]. Any degree of freedom that is not fully constrained leaves the system open to traverse the state-space and encounter states that are not being controlled.

#### *3.1.1.10* Fluctuations

The system being open to the influx of matter or energy from the environment introduces fluctuations into the system state variables. “Physico-chemical systems are capable of exploring the phase space continuously and of performing excursions around the state predicted by the solution of the phenomenological, deterministic equations that describe the systems” [204, p. 148]. These excursions of the phase space provide the variety of the large number of trajectories of each of system components. This mechanism provides “access to novelty” [79] that enables other mechanisms like adaptation or high-dimension dynamics to take hold.

Certainly there are other mechanisms than the ones listed above which lead to complex behaviors. Complex behavior is possible in under-determined systems, where there are many important degrees of freedom that are determined by outside or context-dependent factors. Similarly, emergence is common in open-systems, where the system exchanges energy and matter with its environment. As Polanyi explains, “the structure and functioning of an organism is determined, like that of a machine, by constructional and operational principles that control boundary conditions left open by physics and chemistry" [211, p. 219].

### 3.1.2 Pattern-Formation

Patterns are often considered evidence of complex behavior. However, it would be incorrect to say that all patterns are signs of complex behaviors and are due to the same mechanism. Patterns reflect a spectrum of behaviors that range from the pattern itself being fundamental unit of the system to being a mere epiphenomenon. In the former case, the patterns seen in Conway's Game of Life represent the fundamental entities in the system and the patterns are the reason the behavior has been studied extensively. On the other extreme are patterns like rainbows; while they may be indicative of a certain arrangement and conditions in the atmosphere, they are observer-dependent and they have no meaningful feedback on the system. In the middle of this spectrum is a wide range of systems that have coherent patterns that reflect the state of the underlying system and but also have some causal effect due to feedback effects. Examples include Bénard cells in convective flows or vortices in fluid flows.

A pattern can be defined as a series where there exists "some more efficient way of describing it" [94]. A pattern is therefore a sign of structure that "relates the components" that act "like a constraint in the product space of possibilities" [220, p. 70]. What is the connection between pattern formation and complexity and emergence? Boschetti explains that "pattern formation captures the most intuitive view of emergence. The interaction of low-level simple entities, leading to symmetry breaking, generates a coordinated behaviour; this is expressed by patterns which are novel and identifiable as such by an external observer" [51]. This explanation leads to two questions: Does the pattern have causal influence in order to coordinate behavior? And is the pattern objective? To answer the first question, Crutchfield defines *intrinsic emergence* as features that are "important within the system because they confer additional functionality on the system itself" [82]. While Crutchfield acknowledges that patterns can be intrinsic emergence, they are observer-dependent in that the structure and novelty of the patterns is in the eye of the beholder and are subject to their biases [82, 80].

## *3.2 Types of Complexity: Structural and Behavioral*

While there are many examples of complexity, most can be classified as either *structural* or *behavioral* complexity. Structural complexity is the complexity of the organization of the components of a system and their connections. Structural complexity deals with the physical organization and the flow of matter, energy, and information between components and the environment. Behavioral complexity deals with how the system responds throughout the range of possible inputs. Erdi calls this “dynamical” complexity [109]. Similarly, Deshmukh et al. argue for two types of complexity measures for a complex manufacturing system: static complexity related to the structure of the system and dynamic complexity [95].

Balestrini-Robinson argues that “understanding the relationship between topology (structure) and dynamics (behavior) is of critical importance to understanding the behavior of complex systems” [20]. However, I argue that the greater and more fundamental challenge is understanding and managing the behavioral complexity—after all, the function (behavior) of the system is our end goal. What is the connection between structural and behavior complexity? Erdi argues that “there is no strict correlation between structural and dynamical complexity” [109, p. 3]. Many researchers attempt to measure structural complexity (some of which will be reviewed in this chapter), but it is not clear that there is any fundamental reason to believe that structural complexity is correlated with behavioral complexity in a context-independent way. Systems with large structural complexity can produce simple behaviors, and systems with simple structural complexity can produce complex behavior (one of the hallmarks of emergence). “The structure of a system need not be complicated for its behavior to be highly complex” [258]. The structure and organization of a system clearly influences its behavior; therefore, behavior is function of the organization. An even more interesting case is when the behavior of the system drives its organization, as in *self-organizing systems.*

Structural complexity, without behavioral complexity, can be handled by systems engineering methods in principle. Although the large number of components typical of complex systems makes systems engineering difficult, the systems engineering process can decompose the system and define interfaces. Granted, the number of interfaces can grow exponentially large if there are a large number of heterogeneous components with many possible connections. However, the greater challenge arises only when behavior complexity is added. It is here that systems engineering methods generally fail. Kitto argues that "the systems most consistently defying our techniques are those that exhibit contextual behaviour" [160, p. 1]. As Muncion et al. explains, the connections in a complex system "may be created and destroyed dynamically" and the structure, "instead of being engineered beforehand and imposing its constraints to the system, is the emergent result of the local interactions between the entities of the system" [190]. This coupling between behavior and structure of the system makes analyzing the structural complexity difficult. In that case, structural complexity has to be itself dynamic or considered as a maximum (i.e., worst-case scenario). When the behavior and the structure of the system change in various contexts, especially feedback between structure and behavior, new methods are required to study these systems.

**Claim**: Since emergent behavior is not correlated with structural complexity, its study is more fundamentally tied to behavioral complexity.

**Scope**: This work does not attempt to measure the structural complexity. It also does not assume or imply any connection between structural complexity and behavioral complexity. The goal of this research is to focus strictly on the behavioral/dynamical complexity.

## *3.3 Measures of Complexity*

One of the main goals of complexity science is to develop a measure of the amount of complexity present in a system. Complexity is inherently tied to processes that can be described on a spectrum between random or structured. Figure 6 depicts two conceptions of complexity: *deterministic* and *statistical* [82]. Deterministic complexity is a monotonically increasing function of the degree of randomness in a process, governed by the Shannon entropy rate $h_\mu$ [82]. However, statistical complexity is maximal somewhere between a process that is purely ordered (no randomness) and purely disordered (complete randomness). It reflects that our intuitive notion of complexity lies somewhere between these two extremes.

Like the distinction between Weaver's [250] simple, organized complexity, and disorganized complexity, the type of complexity we are most interested in lies between the two extremes.

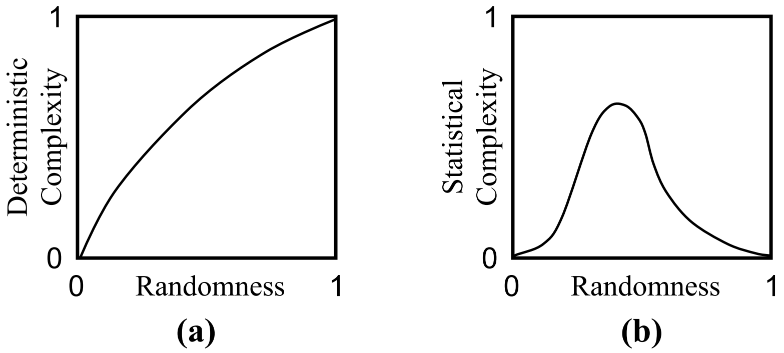

**Figure 6:** Complexity vs. randomness for deterministic (a) and statistical (b) complexity (from [82])

A number of measures exist that attempt to quantify the amount of complexity in a system. A number of complexity measures will be discussed in this chapter. Although the list of discussed measures is nowhere near exhaustive, it does represent a good cross-section of complexity measures. The complexity measures presented here are grouped into three classes: descriptive complexity, statistical complexity, and entropy-based. Descriptive complexity measures are based on the idea that complex systems require more "description" in order to reproduce their behavior. The Information Processing class of measures try to capture the idea that patterns in the system state carry transmit information that can be used to understand and predict the system.

**Table 1:** Taxonomy of complexity measures

| **Class** | **Complexity Measure** | **Key References** |
|---|---|---|
| Descriptive Complexity | Kolmogorov-Chaitin Complexity | |
| Descriptive Complexity | Logical Depth | Bennett [38] |
| Descriptive Complexity | Effective Complexity | Gell-Mann and Lloyd [123, 122] |
| Information Processing | Predictive Information / Effective measure complexity / Excess entropy | Bialek et al. [41], Grassberger [131] |
| Information Processing | Thermodynamic Depth | Lloyd and Pagels [173] |
| Statistical Structure | Multiscale Complexity | Bar-Yam [23] |

### 3.3.1 Descriptive Complexity Measures

The Descriptive Complexity class of measures is based on the idea that complexity is "the amount of information needed to describe a process, a system, or an object" [212, p. 14]. All of the complexity measures in this class use the notion of a computational algorithm as a model for describing the system.

#### *3.3.1.1 Algorithmic Complexity*

Algorithmic complexity is a measure of how complex a corresponding algorithm would be to reproduce a certain output based on an input. This type of complexity is known as Kolmogorov-Chaitin complexity or Minimal Descriptive Length (MDL). Kolmogorov-Chaitin complexity, $K(x)$, is (the number of bits of) the smallest computer program that will run on a Universal Turing Machine that outputs $x$ and then halt. There are several fundamental problems with Kolmogorov-Chaitin complexity. The first is that this measure does not match our intuitive understanding of complexity. Algorithmic complexity is maximized by complete randomness, where the shortest program is simply the data set itself. In this case, the data set is considered incompressible—there is no way to represent the data short of printing the data itself. This brings us to the intuitive mismatch in understanding complexity. Systems that are completely random do not exhibit coherent structure and are therefore not complex. On the other hand, periodic behavior would have a very low algorithmic complexity even though the underlying behavior is more representative of what we intuitively consider as complex. This difference in types of complexity is illustrated in Figure 6 (b). Therefore, algorithmic complexity is often a poor representation of complex behavior between the two extremes of interpretation [204, p. 28]. Related to the above issue is when a process has both structured and random processes occurring simultaneously. In this case, it is likely that the Kolmogorov-Chaitin complexity metric will be dominated by the random portion and will obscure any structured processes. The last problem with Kolmogorov-Chaitin, and all MDL-based approaches, is that they are generally uncomputable [82, 229]. For any practical problem, Kolmogorov-Chaitin complexity is considered biased since the shortest computer program is dependent on both the type of data and the computer language used to implement the program. Clearly, some languages represent a class of data using a shorter syntax and more efficient data structures than others.

Crutchfield et al. argue that all algorithmic complexity approaches are flawed: “Unfortunately, almost all interesting mathematical and quantitative questions about these measures of structure inherit the uncomputability associated with [Universal Turing Machines]. More fundamentally, though, the idea that everything in the world is really a discrete-state computer strikes one as inadequate; at a minimum nature is parallel, continuous, spatially extended, noisy, and quantum mechanical” [84]. Furthermore, Crutchfield argues that algorithmic complexity requires an exact replication of a string; therefore, it is dominated by the randomness in the input [81].

#### *3.3.1.2 Logical Depth*

Logical depth, developed by Bennett [38], is the run time of the Universal Turing Machine that uses the minimal algorithmic description. It is similar to the minimal descriptive length approach. Logical depth “mediates between algorithmic information, where the size of the shortest program is considered, and computational complexity, where the run time for the fastest program is investigated" [17, p. 236] However, since this approach relies on a minimal algorithmic description, it is effectively uncomputable [17, p. 237].

#### *3.3.1.3 Effective Complexity*

Gell-Mann and Lloyd's [123, 122] Effective Complexity measure is based on the concept of a minimum descriptive length, which they call the algorithmic information content. They split the algorithmic information content measure into two parts: one term (the Effective Complexity) to capture the regularities in the system and the other for random components. Thus, the Effective Complexity is a measure of the minimum descriptive length of the system's regularities [122]. This approach is similar to the idea that the "best" model of a system is one that minimizes sum of the model size and the unmodeled errors. A poor model may have a small model size but at the expense of an increased unmodeled errors. On the other hand, reducing the unmodeled errors usually requires a larger model. By finding the minimum descriptive length for the total information, the Effective Complexity is the best description of the system's regularities.

### 3.3.2 Information Processing Measures

Algorithmic complexity deals with the amount of work it takes to reproduce an output exactly. This description is not appropriate for real systems that have noise and must be treated *statistically*. To address this issue, Crutchfield and Young introduced statistical complexity, $C_\mu$ [85]. Statistical complexity is that it is the minimum amount of information about the past required to make optimal prediction of the future at the error rate $h_\mu$ [82]. It measures the degree the system departs from statistical independence [229]. In other words, statistical complexity is a measure of the average amount of historical information stored in the current state [84].

#### *3.3.2.1 Predictive Information*

A measure that tries the capture how well past information helps to predict the future states is predictive information [41]. It is the amount of information that past observations $T$

provide about the future observations $T'$, as shown in Equation 1.

$$I_{pred}\left(T,T'\right) = \left\langle \log_2 \frac{P\left(x_{future}|x_{past}\right)}{P\left(x_{future}\right)} \right\rangle = H\left(T'\right) - H\left(T'|T\right) \tag{1}$$

This concept has been defined by a number of different researchers under a number of names. *Excess entropy* is a measure of the total apparent memory or structure in a source [83, 127]. It considers how much is explained away by considering larger sets of observations (i.e., further back in time). *Effective measure complexity* is the amount of information required for optimal prediction [131, 16]. All of these measures, including others like *stored information*, are based on the same idea. These are also similar to statistical complexity, which is always at least as large as the predictive information [229].

#### *3.3.2.2 Thermodynamic Depth*

Lloyd and Pagels [173] introduce a measure called Thermodynamic Depth. As they explain, "thermodynamic depth identifies the complexity of a state of a physical system with the amount of information processed in the course of constructing that state" [173]. The thermodynamic depth is calculated as the difference in entropy of the macro and micro-level of the system.

### 3.3.3 Statistical Structure Measures

The final class of complexity measures uses statistical methods to measure the amount of structure present in a system.

#### *3.3.3.1 Multiscale Complexity*

Bar-Yam proposes a complexity measure called Multiscale Complexity [23]. Entropy-based measures are one of the most common measures of complexity. Although some of the previous measures are closely related to entropy and other information-theoretic concepts, Multiscale Complexity uses a more traditional interpretation of entropy. In its simplest interpretation, entropy is a measure of the system's state space volume [23] (i.e., the distribution of possible states). He examines how entropy changes as the system is viewed from various scales of observation, thus building a "complexity profile" of a system.

## *3.4 Conclusions*

### 3.4.1 Measures of Complexity versus Measures of Emergence

One point of confusion may be the difference between complexity measures and emergence measures. This confusion is natural since some authors use these terms interchangeably. However, in general, there is no evidence that high complexity measures necessarily leads to emergence, and vice versa. Furthermore, even the relationship between complexity and entropy-based measures is not established. Li argues that “there is no universal relationship between complexity and entropy independent of the underlying sequences” [172].

Therefore, I have attempted to keep separate strictly complexity measures (where emergence is not concerned) and measures where emergence is explicitly sought. The latter measures will be explored in more detail in the section of Measures of Emergence. Complexity measures, without necessarily emergence, are included here since many measures of emergence are related to complexity measures and use similar underlying techniques. Furthermore, a discussion on complex systems cannot be complete without the inclusion of complexity measures.

### 3.4.2 From Structure to Function

Many of the complexity approaches presented above focus on either the complexity of description or the spatiotemporal structure of the system. In particular, many complexity measures are based on entropy. Bialek et al. criticize the over-reliance on “disorder” based complexity measures: “In an attempt to create a universal measure, the constructions can be made over-universal: many proposed complexity measures depend only on the entropy density $S$ and thus are functions only of disorder—not a desired feature. In addition, many of these and other definitions are flawed because they fail to distinguish among the richness of classes beyond some very simple ones” [41]. Another criticism of entropy-based statistical structural measures is that “the relationship between complexity and entropy is not one-to-one, but rather many-to-one or one-to-many” [172]. Therefore, entropy does not provide a unique measure of complexity.

However, what is lacking from this viewpoint is a focus on system function. The shift towards a function-based complexity measure becomes much more important as we move from understanding canonical systems towards engineering realistic systems. It is not clear that there exists a logical connection between the complexity of description or structure and the system's efficacy. Hazen et al. [138] argue against an algorithmic interpretation of complex systems; instead, they claim that *function* is "the essence of complex systems." They propose a measure, degree of function ($E_x$), of a configuration's ability to perform a specific function $x$. They consider the distribution of system states, $F(E_x)$, that achieves a minimum degree of function. They define the *functional information* as shown in Equation 2.

$$I\left(E_x\right) = -\log_2\left[F\left(E_x\right)\right] \tag{2}$$

Although this approach is a good step towards the goal of using complex systems to accomplish a useful function, this formulation is not well-suited for the engineering design problem. Its first flaw is that it requires the evaluation of every system configuration. State-space explosion for realistic-sized systems would make this infeasible. This approach also does not guide us in the parameter-tuning process.

**Takeaway**: The study of complexity has focused too much on structure and representation but has neglected *function*. A more useful complexity measure must include how well the system can perform a function.

This argument is consistent with the claim made in Chapter 2.4 that direct measures of system effectiveness are a better way of evaluating system behavior rather than indirect methods such as analysis of structure or information processing.

### 3.4.3 Complexity in the Distributed Multi-Agent Problem

Returning to the distributed multi-agent problem, it is clear that it has many of the hallmarks of complexity such as nonlinearity, feedback, differences in spatiotemporal scales, large number of components, self-generated complexity, and fluctuations. By having many of the mechanisms and characteristics of complex systems, it will be easier to see that the methodology presented in this research should be generally applicable to other complex systems.

The research presented in this chapter has shown that complexity measures do not appear to be appropriate for detecting emergence. A review of various complexity measures has not shown the necessary connection between the measures of complexity and corresponding changes in system *function*. It was argued in Chapter 2 that emergence in distributed multi-agent systems should focus on measures of system effectiveness rather than structure or description.

# CHAPTER IV

# ENGINEERING OF COMPLEX SYSTEMS

**Chapter Road Map**: The goal of this chapter is to address the overall research question, "how can complex systems that exhibit emergent behavior be engineered?" Various approaches to engineering complex systems are discussed with the goal of developing a methodology that is capable of addressing the challenges posed by complexity and emergent behavior. This chapter addresses portions of Research Question 1 (**RQ1**), Research Question 2 (**RQ2**), and Research Question 3 (**RQ3**):

- **(RQ1.4)** How can emergence be understood in the context of engineering?
- **(RQ2.4)** How should a complex system be analyzed?
- **(RQ3)** What is the appropriate methodology for engineering complex systems?

## *4.1 Engineering Complex Systems*

Engineering requires the analysis and design of a system in order to meet a given set of requirements. Analysis involves the methods used to understand, explain, and predict how a system will behave under a given set of conditions. Design is the process of selecting the configuration and parameters of a system to achieve a desired outcome. This chapter will examine important issues in both the analysis and design of complex systems. In the same way that "emergence" was initially a catch-all term for behaviors that seemed unexplainable, researchers have noticed that systems are similarly cast as "complex" when traditional analysis and engineering techniques have failed to work [219, p. x]. Therefore, we need a new set of methods to study complex systems. Complex systems are often characterized by the difficulty of analysis. Complex systems require analysis and design through a different viewpoint when compared those used on simpler systems. The methods used in the analysis of complex systems come from many different fields of math and science. Important contributions have come from the fields of nonlinear dynamical systems, systems analysis, cybernetics, biology, and the social sciences.

Rouse [218] proposes four views of systems engineering that reflect how researchers have approached understanding and engineering complex systems. These four views, along with the dominant engineering approach and focus, is shown in Table 2. The first view, which he calls Hierarchical Mappings, is the traditional systems engineering approach. In this approach, the system is viewed as a hierarchical decomposition of components. The components are well-defined and the behavior of the overall system is a straightforward aggregation of the behavior of the components. This view of systems engineering is compatible with the top-down engineering approach using system decomposition followed by synthesis and verification. The second view, named State Equations, seeks to understand and control the state of the system. This view focuses understanding state transitions and on feedback mechanisms to control the system. This viewpoint can be traced back to the Systems Analysis and Cybernetics studies of complex systems. The third view, Nonlinear Mechanisms, sees complex behavior as the product of nonlinear interactions between components. The key insight is that "many apparently complex phenomena can be attributed to surprisingly simple mechanisms" [218]. This view is most often seen in the field of physics (i.e., nonlinear dynamics) where complex observed behavior, such as chaos, has a relatively simple mechanistic cause. The final view, Autonomous Agents, sees complexity as the interaction of a large number of relatively simple agents. Because of the dominance of the interactions in determining behavior, these systems cannot be decomposed. In this view, the focus is on emergent behavior.

**Table 2:** Contrasting views of systems engineering of complex systems (adapted from [218])

| **View** | **Approach** | **Focus** |
|---|---|---|
| Hierarchical Mappings | Design composition | Engineering solutions |
| State Equations | Axiomatic derivation | Control performance |
| Nonlinear Mechanisms | Behavior demonstration | Basis of complexity |
| **Autonomous Agents** | **Empirical assessment** | **Emergent behaviors** |

The Autonomous Agents view of empirical assessment using autonomous agents to study emergent behavior is consistent with the methodology presented in this thesis. It is important to understand that these views are not mutually exclusive or exhaustive. Various views can be used for different aspects of the system or phases of the design. As Rouse argues, "investments in systems engineering research should focus on elaboration of the multiple views and creation of means for translating among these views" [218]. Although the methodology proposed in this thesis will be based on the Autonomous Agents view, the other methods of understanding complex systems will be discussed in this chapter.

## *4.2 Methods for the Analysis of Complex Systems*

### 4.2.1 Nonlinear Dynamics

Nicolis and Prigogine argue that physico-chemical systems can act as archetypes for understanding other types complex systems [204, pp. 217–218]; however, the types of complex systems we are interested in are not amenable to the types of analysis used on classical dynamical systems. Typically, we have many more dimensions and number of parameters in engineered complex systems than we do in physico-chemical systems. Engineered complex systems are also heterogeneous, with many constituent components operating under vastly different governing dynamics. Because of this, we are not able to use any kind of analytic method, such as those used to study nonlinear dynamical systems. We cannot come up with a unified set of governing equations and analytically study the solutions to those equations. Statistical mechanics has dealt with the issue of large-scale, multi-scale, stochastic, dynamic, nonlinear, emergent phenomena for a long time. In essence, it is the study of aggregated effects of interactions.

### 4.2.2 Systems Analysis

Both systems analysis and complex systems approaches focus are centered on complex systems. However, there is a difference in philosophy between the two fields. As Phelan explains, "systems theory is predominantly focused on intervention, whereas complex systems is more interested in exploration and explanation" [209]. Systems theory sees complexity as a result of a large number of constituent parts and interactions. On the other hand, complex systems sees complexity as a result of simple rules of agents (i.e., self-generated complexity). Systems theory examines feedbacks [209, p. 239–240] among subsystems; complex systems looks at iterated interactions. Feedback and self-referentiality is important in both, but the emphasis is different. In complex systems, the feedback mechanisms can be context dependent and may be relayed through the environment instead of through direct measurement. As Phelan notes, these differences are not necessarily irreconcilable differences in philosophy; rather, they reflect the state of the art and overall goals at the time of their prominence. This means that as our understanding and analysis capabilities increase, we can unify the approaches of both complexity science and systems analysis. Since the approach advocated in this thesis is primarily focused on exploration and explanation, there will generally be more overlap with terminology and methods from complexity science as compared to systems engineering methods.

Another important difference between systems analysis and complex systems is the assumption of fixed structure. As Manson explains, while complex systems "concerns non-linear relationships between constantly changing entities. Systems theory, in contrast, studies static entities linked by linear relationships defined by flows and stocks (e.g., of energy, information)" [180]. In systems theory, even though the overall behavior may be nonlinear, the underlying relationships are fixed. This structure allows the use of traditional analysis methods, such as solving systems of differential equations.

Systems analysis has its roots in von Bertalanffy's General Systems Theory that used interdisciplinary principles to study open systems. However, of more relevance to the work in this thesis is the theory of cybernetics. Cybernetics is better suited to the study of behavioral complexity; therefore, it will be especially useful for developing the theoretical underpinnings of this thesis. It is important because it provides context and establishes the use of several techniques and ideas, especially the use of black-box analysis and the connection to information theory.

### 4.2.3 Cybernetics

Cybernetics was the study of how to coordinate, regulate, and control behavior of machines. It is important to the study of complex systems for several reasons. The first is that cybernetics helped to establish the connection between information theory and all systems (not just communication systems). The second is that cybernetics also helped to support the view that complex systems have to be viewed as an ensemble. Norbert Wiener and W. Ross Ashby were particularly influential in establishing the cybernetics movement in the 1950s.

Wiener was a mathematician and originally motivated by his work in computing machines used for prediction theory in anti-aircraft systems. He coined the term *cybernetics* to refer to "the entire field of control and communication theory, whether in the machine or in the animal" [253, p. 11]. The term is derived from the Greek word for "steersman" and reflects the study of mechanisms (i.e., feedback) that are used to control systems. In 1948, Wiener developed a statistical theory of information when working on the "problem of noise and message in electrical filters" [253, p. 10–11]. Many aspects of Wiener's statistical theory of information were later formalized and captured by Shannon's information theory. Wiener wrote, "Just as the amount of information in a system is a measure of its degree of organization, so the entropy of a system is a measure of its degree of disorganization; and the one is simply the negative of the other" [253, p. 11]. This connection to information theory would be an important contribution of cybernetics to the study of complex systems that continues to be used to this day. Wiener advocates for the statistical analysis of complex systems [253, p. 33, 37, 92–93]. Although Wiener did not discuss emergence in the same explicit way that Ashby would do, he did claim that the transition from classical mechanics (reversible Newtonian) to irreversible Gibbsian mechanics enabled a framework that allowed the novelty seen in complex systems to exist and parallels the divide between mechanism and emergence [253, pp. 37–38].

While Wiener focused on applications in engineered systems, Ashby was a psychiatrist and focused on studying the brain. According to Ashby, cybernetics sought to ask “what are all possible behaviors that a machine can reproduce?” [11, p. 3]. Ashby applied Shannon’s information theory to the study of systems. He recognized that a communication channel can be generalized to include any process that describes “behavioural relations between two points” [11, p. 180]. Cybernetics embraced complex systems and recognized that systems that large can only be treated statistically. Cybernetics also realized the importance of feedback in systems, which is a circularity of affect. In a way, cybernetics is the study of change in a system. It seeks to identify the transformations between the possible states of a system focusing on what happens instead of why it happens [11, p. 11]. This causation-agnostic approach allows Ashby to use a black-box approach to study systems. This is necessitated by very large systems that can only be specified incompletely (i.e., statistically) and/or direct observations of the system were not possible. In a black box analysis, once the inputs and outputs are given, no more information can be gained. There is no unique transformation between inputs and outputs. The canonical representation specifies a mechanism up to an isomorphism. Ashby conceived of emergence arising only in the case that information is incomplete. If the canonical representation of a black box and all of its arrangements is known, then the outcome is completely determinate (i.e., predictable) and emergence is not possible [11, pp. 110–111]. However, Ashby recognizes that it is often the case that complete knowledge of a black box is not possible, which means that emergence is likely in real systems. Nevertheless, emergence as a concept had a limited role in cybernetics.

Ashby proposed the Law of Requisite Variety as a useful principle within cybernetics [11]. This law stated that the variety in a controller had to be at least as large as the variety of disturbances that it encountered. If the variety in the system was greater than the variety within the controller, the system could not be effectively controlled [26]. Using this law, we can see why complex systems are so difficult to control. In a complex system, the many interacting components have the potential to create an astronomical number of possible states in the system. Although it is not necessarily the number of system states that is the issue, the variety of possible disturbances within the system is likely to increase as the number of possible states increases.

Cybernetics is particularly important because it provided a framework for analyzing the behavior of complex systems that was based on a statistical analysis of behaviors. It provided the connection to information theory, a common technique in the study of complex systems today.

## *4.3 Characterizing the Difficulty of Engineering Complex Systems*

There are a number of characteristics of complex systems that make traditional engineering analysis difficult. The failure of hierarchical decomposition severely hinders the use of traditional, top-down systems engineering methods. This failure is an impediment to the Hierarchical Mappings view presented earlier. State-space explosion occurs when the number of possible states describing the system reaches astronomical numbers due to the number of components, number of interactions, combinations of components, and unconstrained behaviors. The State-Equation view described above becomes impossible to rigorously apply in the face of state-space explosion. Another major issue is the lack of predictability and optimality for complex systems. There are a number of mechanisms—enumerated in the Section Complex Systems: Mechanisms of Complex Behavior (Chapter 3.1.1)—that can contribute to the system having a fundamental property of unpredictability. In the domain of nonlinear physics, most interesting behavior occurs “far-from-equilibrium” [75]. Both of these characteristics contribute to the notion that optimality either does not exist or, more importantly, is not the most interesting and important behavior that we are after. These issues, described in more detail below, make traditional engineering methods inappropriate for studying complex systems.

### 4.3.1 Failure of Simple Decomposition

The problem of studying and designing complex systems is difficult because the traditional engineering approach is ill-suited for studying complex systems. Traditional engineering design has relied on decomposing the system into modular components that have well-defined interfaces. As Bar-Yam explains, "modularity incorrectly assumes that complex system behavior can be reduced to the sum of its parts. As systems become more complex, the design of interfaces between parts occupies increasing attention and eventually the process breaks down" [25]. The V-Model [192] for systems engineering (typically applied to software systems), shown in Figure 7, is based on two phases of the engineering process: *decomposition and definition* followed by *integration and verification.* The failure of decomposition makes this design process fundamentally incompatible.

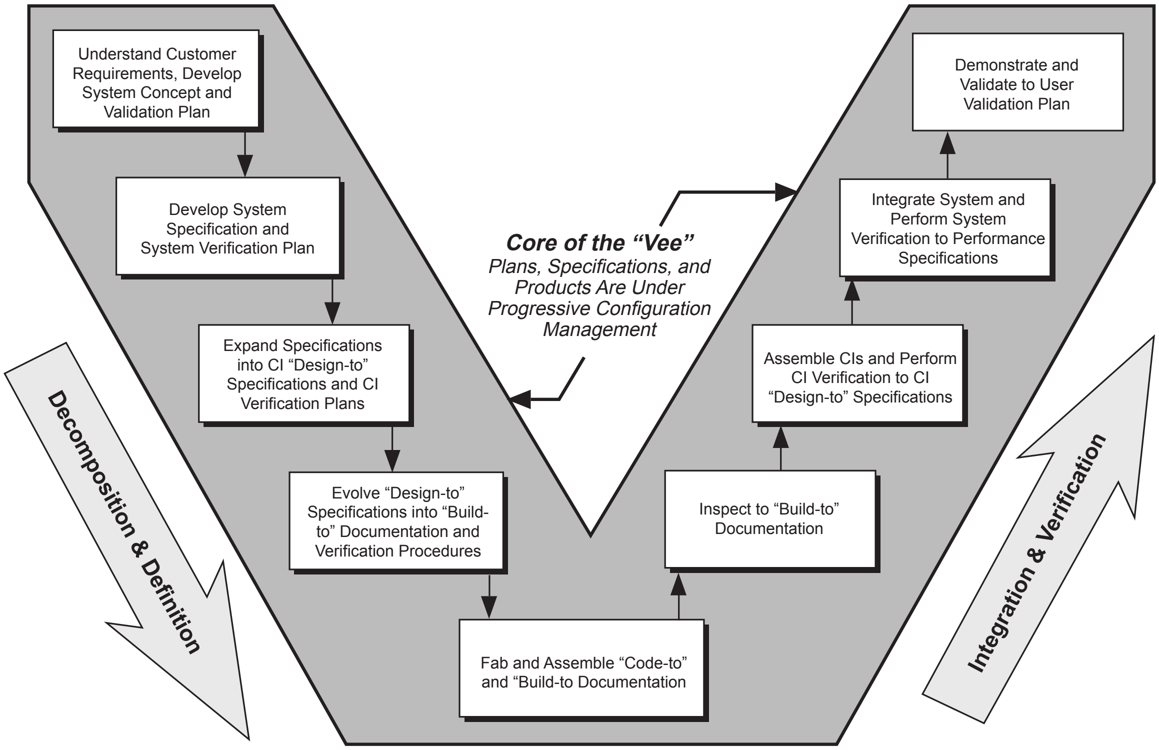
Understand Customer Requirements, Develop System Concept and Validation Plan
Develop System Specification and System Verification Plan
Expand Specifications into CI "Design-to" Specifications and CI Verification Plans
Evolve "Design-to" Specifications into "Build-to" Documentation and Verification Procedures
Fab and Assemble "Code-to" and "Build-to Documentation
Inspect to "Build-to" Documentation
Assemble CIs and Perform CI Verification to CI "Design-to" Specifications
Integrate System and Perform System Verification to Performance Specifications
Demonstrate and Validate to User Validation Plan
Core of the "Vee"
Plans, Specifications, and Products Are Under Progressive Configuration Management
Decomposition & Definition
Integration & Verification

**Figure 7:** V-model for systems engineering (from [192])

Emergence can only be discovered in a *bottom-up approach.* In a bottom-up approach, individual components are designed and then integrated to yield the complete system. However, traditional engineering design has often followed a *top-down approach* where an overall system is designed and then decomposed into subsystems to achieve the overall functionality. This mismatch between approaches is one of the reasons that emergent effects are missed when using a top-down design methodology. Haglich et al. explain how systems with "emergent behavior simply cannot be fully considered through the use of traditional system development methods, such as testing and model checking" [133]. Emergent behavior cannot be anticipated and designed for. It must be observed through inductive methods (i.e., experiments and simulations).

### 4.3.2 State-Space Explosion

Searching for emergence in the design space is difficult because of high dimensionality. Surprising results can occur from the combination of state spaces of the systems, which can become astronomically large due to the curse of dimensionality and the large number of components in a system. The number of interactions between components increases combinatorially with the number of components. The complexity of the systems, coupled with the high dimensionality and uncertainty of the scenario space make this task very difficult. Identifying emergent behavior is challenging because they are often the result of a confluence of many uncertain factors and component interactions. Many of these complex systems operate in highly uncertain environments in which uncertainty masks underlying behavior by confounding the conditions for which the behavior appears. These factors, along with the large number of components in a complex system make the problem intractable for any decision maker or designer. It becomes clear that any analysis technique that requires enumeration of the state-space or its transitions for the system is infeasible [112, 141].

While this combinatorial explosion does make the problem difficult to engineer, it also has the potential to enable new behaviors. As Minai et al. explain, complex systems "benefit from the combinatorial explosion. In combination with a mechanism for selective reinforcement, the diversity provided by exponential possibilities represents an opportunity rather than a problem. The extreme diversity of configurations makes it likelier that solutions to difficult sub-problems are present within this space, and complex systems—notably exemplified by biological evolution—have discovered ways to 'mine' it" [188].

### 4.3.3 Open Degrees of Freedom

Complex systems are characterized by having unconstrained degrees of freedom [188, 56]. This is both a blessing and a curse. The unconstrained degrees of freedom allow for the system to exhibit emergent behavior such as self-organization and adaptation; however, leaving open important degrees of freedom goes against the traditional engineering design. As Wolfram explains, "Conventional engineering requires detailed specification of the precise behaviour of each component in a system. To make use of complex systems in engineering, one must relax this constraint, and instead require only some general or approximate specification of the overall behaviour of systems" [257]. The unconstrained degrees of freedom may exist for several reasons: either they must be left open by design to allow the system to adapt to its environment, or the degrees of freedom cannot be constrained due to deep uncertainty about the environment, or because it is infeasible to apply any constraint on a particular degree of freedom. As an example of the last point, any system which has a human element will be difficult to constrain the interaction between the human and the other parts of the system. Although training and carefully designed interfaces help, making sure humans interacts with the system as designed is a herculean task.

### 4.3.4 Inappropriateness of Optimization

While many have argued towards the use of modeling and simulation for exploratory analysis, the dominant use of simulations in engineering is used for prediction and analysis. When models of systems are well-understood and well-behaved, they can be used for prediction and design. However, in complex systems, the ensemble approach must be used to account for all of the possible range of behaviors. Prediction typically implies the determination of a single outcome, which does not make sense with respect to complex systems. Much of the current design paradigm is based around the use of models for predictive analysis, which consequently allows the use of optimization methods for the purpose of *design.* Design, simply stated, is the process of making choices in order to meet some objective. Norbert Wiener, one of the founders of the Cybernetics movement, describes how "engineering design has been held to be an art rather than a science. By reducing a problem of this sort to a minimization principle, we had established the subject on a far more scientific basis" [253, pp. 9–10]. Maier and Rechtin, nevertheless, advocate for the necessity of the "art" in complex systems design, despite it being "a process of insights, vision, intuitions, judgment calls, and even 'taste' " [177]. Design is often considered an "art" rather than a science because the choices are often made based on intuition and there is little documentation along the way that describes each choice and why it was made that way. While we certainly cannot capture every decision that goes into designing and implementing a system, it is my goal for this research to add rigor to the design space exploration phase of conceptual design.

The introduction of optimization methods coupled with analytic models helped to transform engineering design from art to science. However, the scientific rigor that optimization methods lent to engineering design gets lost with complex systems. John Holland, one of the preeminent complex systems researchers as well as writing one of the seminal works on genetic algorithms, writes that "optimization in complex adaptive systems is rarely possible, and it is often not even meaningful" [143, pp. 244–245]. Complex systems are characterized by unpredictable, stochastic, and sometimes chaotic behavior—all of which make optimization around a single design point meaningless.

Davis [87] argues for "demoting the paradigm of optimization" when dealing with models that exhibit adaptive and complex behavior. He argues that we must embrace the inconvenience and messiness caused by the use of agent-based models. I would agree that complex behavior should be treated with the same rigor as traditional models. However, new techniques and viewpoints will be required to understand complex behavior. At this point, the design of complex systems is not yet a science and still lives in the "art" world. If complex systems cannot be treated as an optimization problem, then how do we design them rigorously? The method proposed in this thesis will provide traceability to the design process of complex systems by helping to eliminate the subjectivity of judgments by the designer. No longer will we rely on luck on the part of the designer to identify interesting behavior.

**Observation**: Traditional, optimization-based engineering design does not make sense for complex systems. Emergent behavior and other unexpected results yield more insight into the system than do "optimal" points. Furthermore, complex systems and modeling approaches demand a shift from simulation for *prediction* to simulation for *insight*.

**Research Objective**: The goal is to develop a method that will help to guide the design space exploration process. It will be based on an objective and quantitative approach toward identifying informative regions of the design space that will yield the most insight into the system behavior.

**Claim**: An empirical, agent-based approach is the most consistent and feasible method of understanding emergent behavior in complex systems.

## *4.4 Design Methodologies (i.e., Design for Emergence)*

Maes asks, "How can a globally desired structure or functionality be designed on the basis of interactions between many simple modules?" [176]. This is the issue facing designers of complex systems. The section above detailed a number of issues that make engineering complex systems difficult. A number of various approaches have been proposed, which will be discussed in this section. Anderson [7] lists four types of approaches that can be used to design self-organizing systems:

1. Top-down engineering
2. Bottom-up simulation
3. Design patterns (i.e., analogy, mimicry)
4. Evolutionary methods

These four approaches should not be seen as alternatives; rather, elements from each of these approaches can be used as necessitated by the problem at hand.

### 4.4.1 Top-down Engineering

The first approach, top-down engineering, is the traditional engineering process where requirements, subsystems, and interfaces are specified very early in the design process. By definition, emergent behavior is *not* specified in the system description. Since macro-level behavior (i.e., the system response) cannot be predicted from the micro-level specification, top-down engineering is precluded from being the primary design approach.

As discussed earlier, complex system behavior cannot be predicted from component specifications. Top-down engineering is a poor fit for designing complex systems. It is not difficult to construct a system which exhibits emergent behavior; however, engineering a system for a specific global behavior is difficult. The global behavior is not predictable and therefore the local behaviors cannot be defined a priori in order to meet some target global behavior. Emergent behavior can only be identified a posteriori of the integration of the system.

#### *4.4.1.1 Traditional Design Process*

Figure 8 shows the steps of the conceptual and embodiment design within the traditional engineering design process [99]. This design process is strongly linear and relies heavily on designer intuition. This design process is incompatible with complex systems due to the failure of decomposition and the inability to separate the concept evaluation, product architecture, configuration design steps, and parametric design steps. In a complex system, all four of these steps are strongly coupled and must be evaluated concurrently.

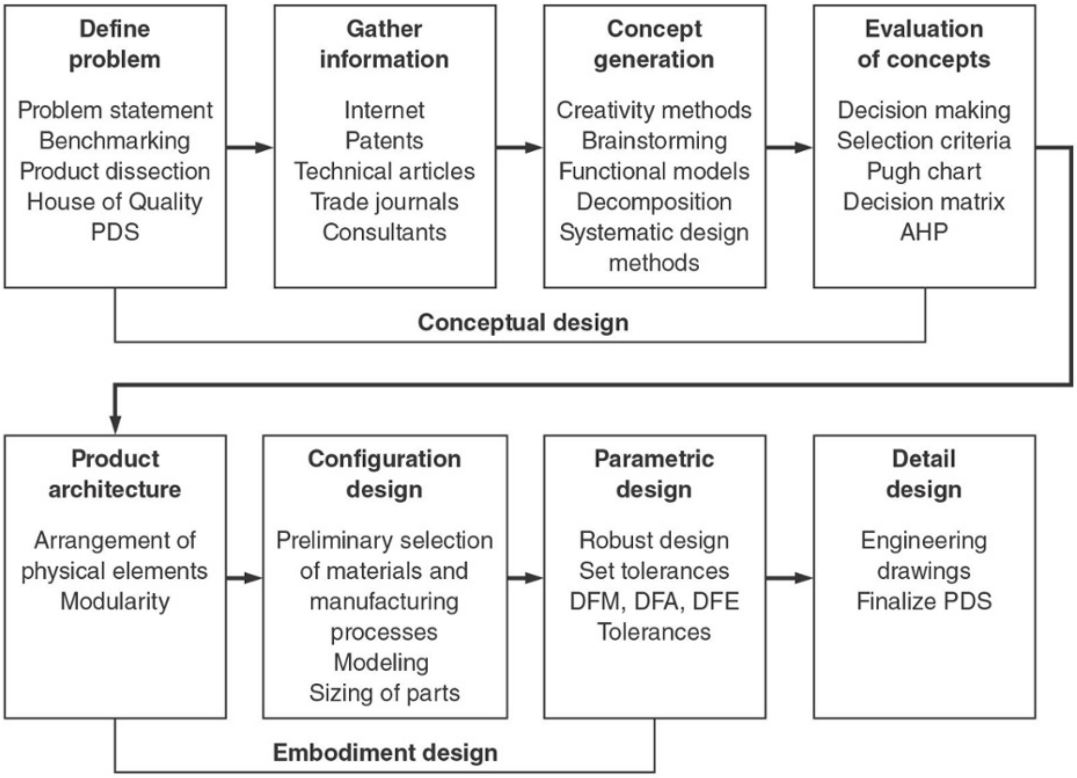

**Figure 8:** Traditional engineering early design process (from [99])

#### *4.4.1.2 Spiral Development*

The traditional engineering design process is inherently linear and makes it difficult to account for complexity within the design process. The spiral development process [177], shown in Figure 9, has built in iteration that allows for changes in the system design to account for better information about changing requirements, system performance, and other sources of uncertainty. It is clear that the spiral development process is a better fit for engineering complex systems. However, it is likely that the spiral development process for a complex system is likely to need many more iterations than is practical. Thus, although spiral development offers a better design process, the number of iterations required to converge on a design will most likely require another process better suited towards massive iteration.

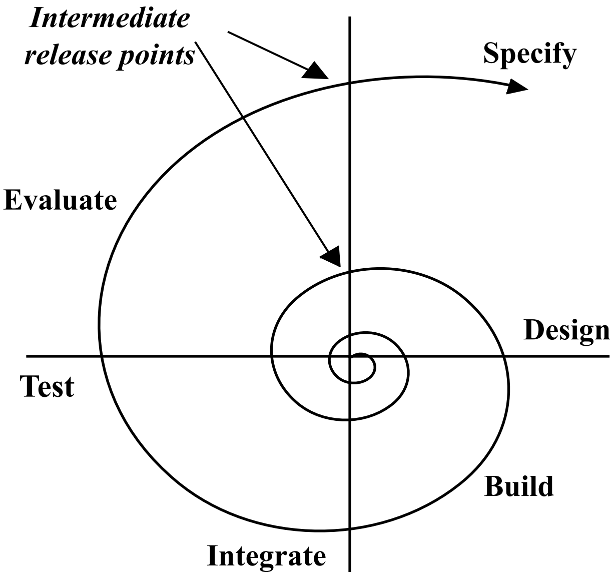

**Figure 9:** Spiral development model for systems engineering (from [177])

### 4.4.2 Bottom-up Simulation

Bottom-up simulation is based on the recognition that top-down specification is not feasible and that system behavior must be determined through building a sufficient model of the system and then simulating the behavior. Ulieru and Doursat argue that engineering complex systems needs to shift from a top-down to a bottom-up approach, which enables the system to manifest its emergent properties [243, p. 41]. However, by itself, bottom-up simulation is not a design methodology—it is just a small part of a much larger process. Although it would appear as though top-down and bottom-up methodologies are incompatible, it is likely that the correct design methodology will need to use a mixture of the two. High-level design requirements and partial specification of subsystems and interfaces may be specified in a top-down manner, while many other system parameters are left unspecified until the system can be simulated and the effects of the parameters can be evaluated.

However, bottom-up design is not a panacea for the difficulties associated with top-down engineering. As Carreras et al. [60] note, bottom-up approaches simply shift the difficulty from the initial specification to the design of the appropriate framework and components which will be capable of generating the desired behavior. The design process attempts to develop the correct set of rules, and the right mixture of components, that will reliably achieve the goal behavior. Fromm notes that a bottom-up approach faces a severe challenge as “the number of combinations and configurations grows exponentially with the number of states, elements, and rules” [117].

### 4.4.3 Design Patterns

Design patterns are reusable coordination methods which result in a desired behavior[237, 60]. For a limited set of problems, design patterns provide a short-cut to bridging the gap between micro-level coordination and macro-level behavior. There are many examples of nature generating elegant or robust systems. Analogy and mimicry are attempts to replicate natural processes in the design of engineered systems. If a system can be designed in a way that is analogous to a well-understood phenomenon, analogous mechanisms can be put in place to achieve a similar behavior observed in nature. However, it is likely that most problems that system designers are not amenable to natural systems. Additionally, many natural systems may not be as efficient as required or we may not be able to tolerate failure in individual components as nature would (e.g., the loss of a small number of ants in an ant colony is not a problem; however, if we use that analogy when designing transportation systems, we would not want to tolerate the loss of vehicles if lives are at stake).

Although not necessarily, almost all design patterns used have been taken from nature [97]. Sudeikat and Renz [237] provide a list of design patterns from nature: molding and aggregation, quorum, web weaving, morphogenesis, brood sorting, flocking, schooling and herding, nest building, and foraging. These design patterns provide a catalog of possible coordination mechanisms; however, these are only a small fraction of possible coordination mechanisms and are not general enough to be a solution to the design problem.

### 4.4.4 Evolutionary Design

Bottom-up simulation approach does not specify how to manipulate the system and the rules in order to achieve a specified behavior. Although design patterns offer a solution in cases where the analogy is appropriate, more generally, it is not clear how to modify the micro-level behavior. There are many researchers advocating a design methodology that takes advantage of evolutionary techniques [25, 178, 27, 188, 231]. Ulieru and Doursat propose a methodology called emergent engineering (EE) which uses general principles that will produce the desired global behavior without dictating them using an evolutionary meta-design process [243, 103, 104]. The meta-design process is based on "designing the components of the system and endow them with capabilities of dynamic self-assembly, disassembly, and re-assembly, in order to enable *evolve-ability*" [243, p. 44]. The goal of the meta-design process is to come up with the correct framework of components and rules that are able to evolve towards the desired functionality.

When using evolutionary techniques in the design process, there are generally two approaches. The first approach uses evolutionary algorithms, most commonly genetic algorithms, to optimize system parameters to improve the fitness of the system. The second approach, called interactive evolution [7, 46], replaces the evolutionary algorithm with a user who guides the evolutionary process by selecting the winning design at each iteration. The motivation for this approach is to take advantage of the user's insight, experience, and ability to evaluate patterns and other interesting behaviors. However, this approach may also be biased by the user—a user's "experience" may make them less likely to select unorthodox solutions. Rather, the goal should be to minimize the designer's bias towards a particular outcome [42]. One major problem with both of these approaches is the lack of traceability. Especially within interactive evolution, it is difficult to understand and document how the final solution was obtained.

Similar to the other approaches presented above, evolutionary methods alone are not sufficient as a design methodology. Edmonds [106] argues that we need an approach that combines engineering and adaptation. Evolutionary methods focus too much on adapting random designs while formal methods rely too much on strict proofs. Bottom-up simulation, combined with evolutionary techniques, offers a more complete picture about a possible methodology.

Genetic algorithms (GA) are a popular way of implementing evolutionary design [32]. However, GA suffer from a number of issues. They do not scale well to large design problems that have many design variables with many potential settings. GA also only consider the final solutions and do not take advantage of the structure of the solution space [251]; therefore, they are inefficient evaluators of models that are also plagued by stochasticity and that require long execution times. Because of these issues, it is not likely that GA will be able to handle exploring the entire design problem. Additionally, we will always have some design parameters under our control. The desired methodology should include a way to evaluate the design parameters we control along with those that are subject to evolutionary changes.

**Observation**: The desired methodology needs to be able to evaluate the influence of both design variables under our control and those under evolutionary influence.

## *4.5 Developing a Methodology*

One of the most important elements of a design for emergence is a methodology for searching the system design space to identify emergent behavior. As Dogaru explains, "it is of practical interest to develop methods to locate in this space of parameters the regions where emergent phenomena are likely to occur. This question is in fact difficult to answer since 'the surprise effect' associated with emergence does not allow us to specify in advance what we are looking for i.e. the desired emergent behavior" [102, p. 4]. The design space exploration methodology should be general enough to allow us to identify specific design points while still allowing us to evaluate the points within the greater context of the landscape of possible behaviors of the system. This leads to the general research question below:

**Research Question** (**RQ3.1**): What are the required features of a design methodology for emergence?

A partial answer has already been identified as a crucial part of the design methodology—the systematic exploration of the design space.

**Claim**: Design space exploration is required in order to identify emergent behavior in a system.

### 4.5.1 Design Space Exploration

Design space exploration is the key element in a design for emergence methodology. Design space exploration, simply stated, is the evaluation of various candidate solutions. The goal is to understand the system behavior as the various design variables are modified. As shown

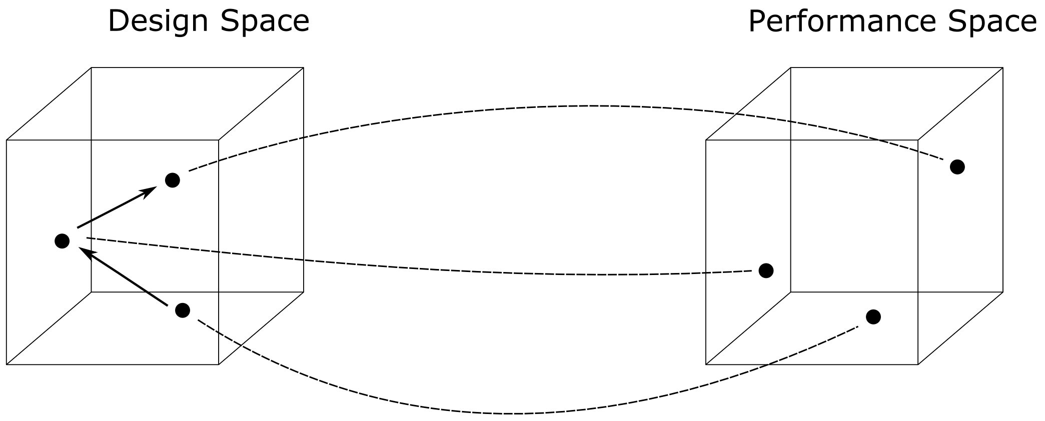

**Figure 10:** Mapping between the design space and performance space (adapted from [71])

in Figure 10, both the design and performance space can be represented as a hypercube, where each dimension in each space represents either a design variable or a performance measure, respectively.

There are two complementary goals when performing design space exploration: 1) parameter tuning, and 2) design space characterization. Parameter tuning is the selection of appropriate values of the design variables in order to achieve a particular behavior. Design space characterization is a less well-defined process; however, it is generally the identification of all possible behaviors the system is capable of exhibiting over a range of initial conditions. This step also includes features such transition points in behaviors and any other "interesting" features.

Davis [88] argues that the appropriate method of evaluating complex systems is using exploratory analysis, in which the entire domain of possible initial states is evaluated. Hastings and McManus [137] note the importance of performing this exploration in order to identify both the positive and negative outcomes so that they can either be exploited or mitigated, respectively. Dogaru [102] proposes a "design for emergence" approach that focuses on the parameter tuning aspect of the process; however, the proposed approach cannot be applied to the general design space exploration problem because of its reliance on cellular neural networks. Despite the lack of applicability of Dogaru's method, it does help to establish the body of methodologies that try to identify the parameter settings that lead to emergent behavior. Both De Wolf and Holvoet [92] and Welch et al. [252] advocate a scientific-approach to parameter tuning; however, neither suggest a specific methodology of exploring the parameter space. This leads to a sub-question of Research Question 3 (**RQ3**):

**Research Question** (**RQ3.2**): How can the design space of a complex system be systematically explored in order to identify emergent behavior?

A review of various design space exploration techniques will be explored below.

## *4.6 Design Space Exploration Methods*

The appropriate method for design space exploration depends on the characteristics of the design space and response variables. A number of strategies are shown in Figure 11 based on the characteristics of the design space $\theta$ and the response variable $f$ of interest. If the design space is discrete and the response is deterministic, an exhaustive evaluation and ranking of all design points can be performed for small design spaces while search methods will have to be used for large design spaces. If the design space is continuous and the response is deterministic and continuous, surrogate models and gradient-based optimization can be used. The goal of this research is to develop a method for design space exploration if the design space is continuous but the response variable is stochastic. This is stated as a research question:

**Research Question** (**RQ3.3**): What is the appropriate way to perform design space exploration on a complex, stochastic space?

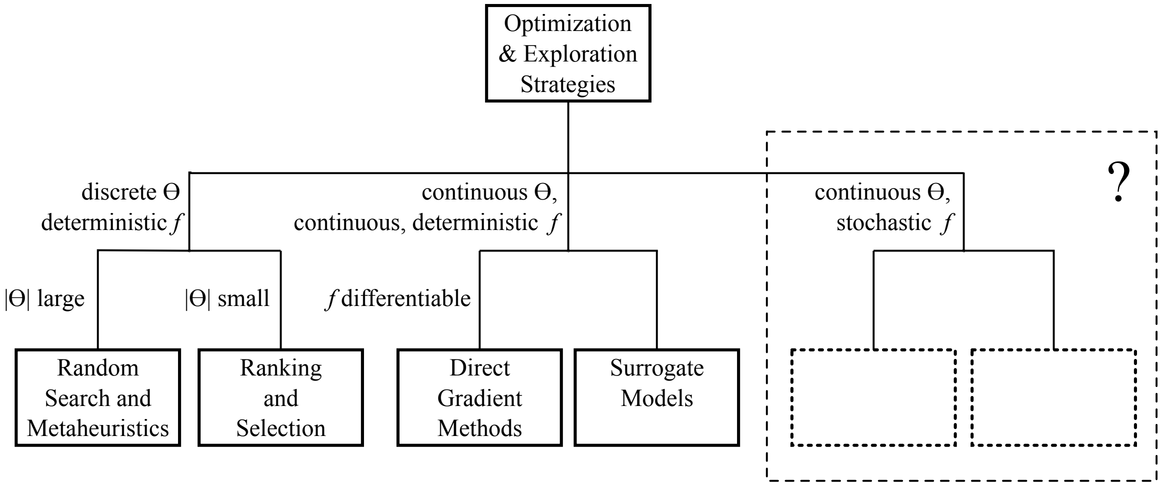
Optimization & Exploration Strategies
discrete Θ deterministic f
continuous Θ, continuous, deterministic f
continuous Θ, stochastic f
?
|Θ| large
|Θ| small
f differentiable
Random Search and Metaheuristics
Ranking and Selection
Direct Gradient Methods
Surrogate Models

**Figure 11:** Optimization strategies for various problems based on design space, $\theta$, and response, $f$, characteristics (adapted from [22])

### 4.6.1 Ad Hoc

Many steps in the design process rely on the designer's intuition, thus making them ad hoc. Although design's intuition can serve as a shortcut, it eliminates the traceability of a design. More importantly, unexpectedness is a common characteristic of emergence—intuition will no longer be valid. Instead, a systematic and thorough exploration of the options must be carried out.

### 4.6.2 Design of Experiments

Design of Experiments (DOE) is a set of methods for structuring experimental setups in order to yield the most amount of information from the data. Given the factors we wish to explore and their ranges, DOE methods can tell us the combinations of factors to run in the most effective manner. The method used depends on the goal of the experiment. A full-factorial design is the most exhaustive way of sampling the design space; however, the number of points grows exponentially with the number of dimensions. Fractional factorial designs use a fraction of the full factorial design in order to cut down on the number of points required at the expensive of confounding in determining the interactions between factors. Building a 2nd-order response surface model can be done effective using a central composite design. However, these last two methods are primarily useful for building regression models, which is not the goal within this research. The sampling used for the fractional factorial and response surface is generally fairly sparse and assumes that the underlying model or process is fairly well-behaved and follows the principle of parsimony, where the number of factors and interactions are relatively small. For the purpose of this research, the objective of DOE methods will be to thoroughly sample the design space; therefore, full-factorial designs and space-filling designs are appropriate candidate solutions.

### 4.6.3 Surrogate Modeling

Surrogate modeling is a design space exploration technique that can be used to rapidly evaluate the design space [88]. In this approach, a mapping between the design space and the performance space is created using a mathematical function, usually a statistical regression. The surrogate model is a deterministic function that is easy to evaluate. Surrogate models are limited to capturing a single valued regressed variable and not a distribution that we need to accurately capture the range of possible outcomes. In the surrogate modeling approach, $f\left(\theta\right)$ is the system response as a function of the design variables, where $\theta$ is the vector of design variables of the system. Typically, the surrogate model is the expected value of some system performance measure $Y\left(\theta\right)$ as shown in Equation 3.

$$f\left(\theta\right) = E\left(Y\left(\theta\right)\right) \tag{3}$$

Although surrogate models are purportedly developed based on stochastic response functions, the nature of the assumed uncertainty is very limited. The range of behaviors in the performance space of a complex system are not due to simple uncertainty. In traditional engineering design problems, the variation in the response can be treated as a "noise" variable where there is a relatively small variation around the mean. As an example of this assumption, in the Response Surface Methodology, the response is modeled as a deterministic regression plus an error term $\varepsilon \sim N\left(0, \sigma^2\right)$ that is assumed to be normally distributed with a constant variance (i.e., homoscedastic). Uncertainty is *always* present; however, the degree to which it influences the outcome is the difference in whether we are able to treat it as noise or as a deep and fundamental uncertainty in the range out possible outcomes. In the latter case, the range in possible behaviors may be driven by adaptive or evolutionary behavior.

**Claim**: Due to the inability of surrogate models to capture nonparametric distributions, the surrogate modeling approach is not an appropriate design space exploration method for complex systems.

This approach assumes a form (i.e., model) of the performance space. Although some surrogate model approaches (e.g., neural nets) are capable of handling nonlinear responses, they still impose a limitation on the form of the response. Kriging methods are more general than polynomial models but similarly assume a smooth response function [22]. Kriging models generally perform better when attempting to model the global response of a simulation [161].

Kernstine's research focused on creating surrogate models of complex systems. He showed the difficulty of creating surrogate models of stochastic design spaces [156, 157, 155]. Kriging is an interpolation method that is often used as a surrogate modeling technique. Its response is traditionally a single deterministic value; however, using a "nugget" parameter, a stochastic response can be attained. The "noise" in the response is assumed to be normally distributed and with constant variance (i.e., homoscedastic), which is not a good model for complex systems [156]. Kernstine et al. found that both Kriging and the Beers method performed poorly with complex systems [157]. Due to Kriging's computational expense, it is not capable of handling very large data sets. Kriging was found to be prohibitively slow for data sets above 500 points [156]. Additionally, it was found that MARS methods "may be incapable of determining interactions in the presence of noise" [156]. In summary, Kernstine writes, "it can be concluded that [system of systems (i.e., complex systems)] simulations may not follow traditional assumptions for regression methods (parametric, normally and identically distributed)" [156]. However, his work focused on capturing mean and variance, but not the actual distribution, greatly limiting the amount of information we could get from a complete distribution.

It is clear that regression methods that only capture a single value (often just the deterministic response or the mean of a stochastic response) are insufficient for adequately characterizing the behavior of a system. One proposed approach is the use of quantile regression, where regions of the probability distribution are modeled as a way of characterizing the system behavior [213]. Although this is a step in the right direction, a fully nonparametric model of the probability distribution is superior to regressions based on quantile subsets of the distribution. A data-driven approach is necessary to avoid introducing artifacts in the response that may not be actually be present in the underlying data.

### 4.6.4 Optimization

Optimization methods are well-suited for the task of automatic parameter tuning. Once an objective function has been defined, an optimization algorithm can be used to modify the design variables to improve the value of the objective function. Terano [240] uses genetic algorithms to explore the parameter space of an agent-based simulation. The parameter values which lead to the best solution at the end of the optimization are selected. This approach has the benefit of providing objective, specific solutions. The use of the objective function results in all candidates being evaluated rigorously and consistently. As mentioned earlier, the use of optimization helped to turn design from “art” to “science.”

However, optimization methods have several flaws as a design space exploration method. They focus heavily on the specific solutions at the expense of characterizing the landscape of all behaviors in the design space. The singular focus on maximizing the system performance (i.e., optimality) often leads to solutions that are not robust.

**Observation**: Optimization methods have the beneficial property of focusing on specific solutions and can be used to reduce the ad hoc nature of design. The proposed methodology should likewise be an objective measure that leads to specific candidate designs.

### 4.6.5 Inverse Design

Another common method for design space exploration is the use of inverse design. In inverse design, the design space is sampled thoroughly and the corresponding responses are stored. Afterward, the desired responses are selected based on their ability to meet the desired performance metrics and constraints. The design points which result in these selected configurations are chosen as candidate designs. This process works well if there is a one-to-one mapping between outcomes and input design points. However, this approach does

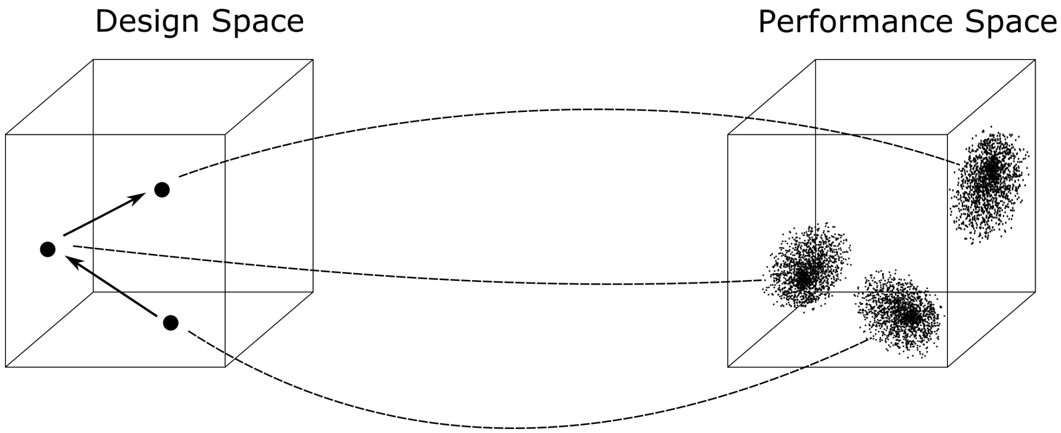

**Figure 12:** Mapping between the design space and performance space for complex systems

not work when dealing with complex systems where the metrics are ensembles of possible outcomes. This one-to-many mapping between design points and outcomes means that it is impossible to invert the problem. In a complex system, the performance space is uncertain and has to be treated as an ensemble (i.e., the set of all possible outcomes). Figure 12 shows the difficulty of mapping the performance space back to the design space. It is likely that any selection from the performance space will not lead to a unique design point.

**Observation**: Current design methods make it difficult to explore design space for complex systems. In particular, inverse design methods are difficult to interpret for complex systems.

### 4.6.6 Sensitivity Analysis

There are two types of sensitivity analyses: local and global. Local sensitivity determines the sensitivity around a specific solution; on the other hand, global sensitivity determines the sensitivity of the entire model to the inputs. Local sensitivity examines the perturbations on the response from each of the individual function dimension. Given a model $y = f(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_n)$, the local sensitivity of the solution $y^* = f(\mathbf{x}^*)$ is the partial derivative $\left({}^{\partial y}/_{\partial x_i}\right)_{x=x^*}$. Local sensitivities are typically evaluated one dimension at a time. Because of this, they may not capture the coupling between variables. Complex systems often have highly-coupled variables, so this limitation is significant. Furthermore, local sensitivity is usually formulated with a scalar response; however, complex systems are inherently stochastic, so local sensitivity needs to be re-formulated to capture more general changes in probability distributions. Despite these limitations, the idea of a local sensitivity may still be useful within the context of complex system design. Complex systems are context-dependent (i.e., their behavior is strongly influenced by the parameters of the system and the environment), so it is useful to find the design points $\mathbf{x}^*$ that have a highly sensitive response, which may indicate a system lever-point that allows the designer to affect behavior.

Global sensitivity considers the response $y$ of the model as all of the inputs are varied simultaneously over the domain of $\mathbf{x}$. The local sensitivity studies a specific solution while the global sensitivity studies the model as a whole [235]. Most sensitivity analysis techniques are local [71, p. 97], but the Monte Carlo approach described briefly in the next section is an example of a global sensitivity method. The drawback with the global sensitivity approach for complex systems is not useful for parameter value selection—it does not tell us how to achieve a particular behavior.

### 4.6.7 Monte Carlo Methods

Markov chain Monte Carlo (MCMC) techniques use repeated samples from allowed input distributions to estimate output distributions. MCMC techniques are also used to estimate numerical integrals over a large number of dimensions. They are useful because of the relative ease in which they can be implemented. Additionally, MCMC techniques are not affected by the number of dimensions—the accuracy of an estimate is not dependent on the dimensionality of the problem, although getting independent samples becomes more difficult as the dimensionality increases [229, pp. 42–43]. The Metropolis algorithm [186] is the classical implementation of the MCMC sampling method. Another use of Monte Carlo methods is for sensitivity analysis. Monte Carlo can be used to examine the global system sensitivity subject to the distributions on the inputs [235].

## *4.7 Conclusion*

The research presented in this chapter examined various types of design methodologies as well as techniques for performing design space exploration. It is clear that top-down methods are incompatible with our current understanding of emergence. On the other hand, bottom-up simulation fits perfectly with the theory and practice of complex systems.

Bottom-up simulation is an appropriate choice for designing a methodology. The final two possibilities have limitations that will eliminate their further consideration. Design patterns are great because they provide us with examples and mechanisms for building complex systems. However, as discussed earlier, design patterns have limited applicability and are unlikely to be helpful outside of very specific applications. Finally, evolutionary design can solve the problem of parameter tuning but leaves out the equally important task of design space characterization. Evolutionary design is based on optimization techniques, so they suffer from all of the issues related to optimizing complex systems. This approach would also only show beneficial emergent conditions but would completely neglect emergent vulnerabilities, which are equally important.

The final outcome from the research in this chapter is the selection of the design space exploration technique. It was clear that several approaches are inappropriate. Surrogate modeling, optimization, and inverse design techniques were argued as flawed due to several important considerations. However, Design of Experiments offers a potential solution for performing design space characterization. DOE techniques can be used to efficiently sample the design space to understand how the system changes behavior. Sensitivity analysis also offered some insights that proved to be useful. The proposed definition for engineered emergence (as a reminder, engineered emergence is the critical transition in macro-level behavior due to changes in system context (i.e., environmental conditions or system parameters)), is analogous to a measure of local sensitivity. Thus, evaluating the proposed emergence measure will be similar to evaluating local sensitivities with modifications to take into account the probabilistic nature of the problem.

# CHAPTER V

# MEASURES OF EMERGENCE

**Chapter Road Map**: For emergence to be a useful concept, there must exist a way to quantify it in an objective manner. This goal is often called quantitative emergence. This chapter will review different approaches that researchers have taken to measure the amount of emergence in a system or process. Finally, a measure of emergence for the methodology presented in this thesis will be presented. This chapter addresses portions of the Research Question 1 (**RQ1**):

- **(RQ1.3)** How can emergence be detected or measured?

## *5.1 Overview*

There are a variety of approaches used to measure or detect emergence. These methods will be organized into three groups: model-based, multi-scale, and metric-based. Model-based methods recast the system using a particular formalism that enables detection via a fundamental feature of the formalism. By using a model that naturally lends itself to an interpretation of emergence, model-based approaches offer a rigorous and self-consistent method for emergence detection. However, model-based approaches are also the most restrictive. They are limited in their ability to model systems of all types. Many model-based approaches (e.g., those based on automata theory) become unwieldy when faced with realistic systems due to state-space explosion. Like many model-based approaches, multi-scale methods corresponds nicely with a particular interpretation of emergence. While coarse-graining is good for identifying coherent structures in a multi-scale system, its usefulness for finding general forms of emergence is limited. Coarse-graining is well-suited for examining systems large-scale systems exhibiting spatial and temporal patterns. This type of emergence is common in biology, chemistry, and condensed-matter physics; however, it may not be as common in engineered systems. The last group of methods are the metric-based methods. These methods attempt to detect emergence by measuring some aspect of the system state or a measure of system performance. This approach provides the most generality since it can be applied to almost any system or model. The main drawback of this approach is that the connection to emergence (especially any of the formal definitions of emergence) is more tenuous. All three groups of methods will be discussed in more detail below.

## *5.2 Model-Based Methods*

Model-based methods attempt to understand and classify the underlying dynamics of the system rather than measuring the state of the system directly. They can be thought of as a meta-model approach: build a model to capture the observed dynamics and then study the behavior or complexity of the meta-model in order to understand the behavior of the original system. The goal is to select a model which both captures the structure and dynamics of the system while also having a correspondence with a definition of emergence.

Many of the model-based approaches have used automata theory to model the behavior of the system. This approach was convenient because the study of the formal languages of the automata systems allowed for a rigorous comparison between the language of the individual components and the whole system. Other research has focused on the computational complexity of the resulting automata machines. The machine complexity, rather than the state or output, is examined. A number of methods using this approach are described below.

### 5.2.1 Automata Theory

A number of techniques used to model and detect emergence are based on automata theory and the resulting symbolic dynamics of the system. An automaton is a discrete finite model that process sets of symbols (i.e., strings that map to discrete states). The automaton operates under a set of rewriting rules (i.e., grammar system) on the input, and outputs a formal language, $\mathscr{L}$, that is the set of all finite strings that it can produce.

Kubik [165] uses grammar systems to study multi-agent systems. In this language-theoretic approach, he studies the difference in language that can be generated by the whole grammar system compared with the languages generated by individual grammars of individual agents. The summation of agents' languages is considered to be the overall system behavior. Kubik calls this the "sum of conditions the agents can bring about in the environment if they act individually in the environment" [165, p. 51]. A language (i.e., behavior) is considered emergent if it can be generated in the whole system but cannot be generated in the individual agents.

Some limitations of this approach include how the formalism restricts it to fairly simple idealized systems. Since the automaton is discrete and must operate on a finite alphabet, continuous systems and time-series data must be partitioned, where an interval in the continuous domain corresponds with a symbol in the discrete domain. However, the dynamics of the system are quite sensitive to the choice of partitions when transforming continuous systems into discrete [229]. As Bollt explains, "the consequence of a misplaced partition can be severe, including significantly reduced topological entropies and a high degree of non-uniqueness" [43]. Similarly, if the time-series is constructed from a measuring instrument, noise can lead to measurement-induced complexity [82]. The other serious issue is that this approach suffers from state-space explosion due to the combinatorial nature of the sets of strings that comprise the language observed in the system [239].

#### *5.2.1.1 Computational Mechanics*

In Crutchfield's intrinsic emergence, emergent structures "confer additional functionality which supports global information processing" which leads to "an increase in intrinsic computational capability" [80]. Emergence, in this computation-theoretic approach, is defined if "a process undergoes emergence if at some time the architecture of information processing has changed in such a way that a distinct and more powerful level of intrinsic computation has appeared that was not present in earlier conditions" [80]. A computational model is constructed that attempts to predict a discrete series of measurements from a process. The architecture and complexity of the resulting machines therefore reflects the information processing capabilities of the system it models. The computational machines are classified according to their computational resources: amount of historical data, available memory, and time available for estimation [82]. This approach is known as *computational* *mechanics.*

However, this focuses on information processing *within* a system. While the mechanism that drives emergence is interesting, it is the *outcome* we are most interested in. As Auer et al. explain, "Computational mechanics discusses how a [system] computes but not what it computes" [13].

#### *5.2.1.2 Hierarchical $\epsilon$-Machine Reconstruction*

The process of using computational mechanics involves the discovery of new model classes that better describe the underlying process it attempts to model [80]. "The key step in the emergence of complexity is the 'innovation' of new model classes from old. This occurs when resource limits can no longer support the large models—often patchworks of special cases—forced by a lower-level model class" [82]. Hierarchical ε-machine reconstruction is an approach to model the appearance of novelty through the generation of increasing computational capabilities of the computational machines. An ε-machine is a minimal model at the least computationally powerful class yielding an optimal finite description [82]. The statistical complexity, $C_\mu$, is the size of the reconstructed ε-machine and the entropy rate, $h_\mu$, is the rate at which information is produced. The entropy rate (also known as the entropy density or metric entropy) is shown in Equation 4, where $\Pr(s^L)$ is the marginal distribution over the set of length $L$ sequences $s^L$ and $H$ is the average of the self-information.

$$h_\mu = \lim_{L \to \infty} \frac{H\left(\Pr(s^L)\right)}{L} \tag{4}$$

Hierarchical ε-machine reconstruction is based on *causal states*, which are the set of subsequences that render the future conditionally independent of the past [85]. If two configurations lead to an identical outcome, they can be represented by the same state. Once the causal states, $S$, are found, the symbolic dynamics capture the transitions, $T$, (i.e., the map) from state to state. The ε-machine is the set $M = (S, T)$ and the process of generating the machines is known as hierarchical ε-machine reconstruction. In practical situations, ε-machines can be generated using the CSSR algorithm [230]. Vrabič and Butala apply the ε-machine approach to studying complexity in manufacturing systems [247, 248].

This approach captures the information flow in the process since the information can be determined from the transition probabilities between causal states. The complexity of the underlying process can be inferred from the ε-machine themselves. In this way, the complexity of the language is determined by looking at the minimal automaton capable of recognizing the language. The Chomsky hierarchy is often used to delineate increasing computational complexity in formal language systems. It is essentially based on the memory requirements for each automaton. However, the automata classes are fairly coarse and changes between classes may not actually correspond to any meaningful changes in the actual system. As Badii writes, "even apparently simple physical systems may turn out to correspond to high computational classes" [17]. It is quite possible that simple physical systems can only be characterized using a Universal Turing Machine, leaving us with little to no room to differentiate between observed behaviors.

### 5.2.2 Regression-Based Methods

Seth [226, 227] propose a method of detecting emergence using the Granger causality formalism. This is a model-based technique because it models the system behavior in terms of a statistical regression. A number of concepts are defined within this framework. A variable is *autonomous* if its own past history better predicts future states than the history of other system variables. A variable $X$ *causes* another variable $Y$ if the inclusion of the $X$ reduces the variance of the prediction of future states of $Y$. In his framework, a variable is emergent with respect to another variable if it is both autonomous from and caused by that variable. The fundamental flaw with this approach is that it relies too heavily on the ability to model the system with a given regression. Although the regression technique can be improved to include nonlinear terms and more generalized regression models; it is still fundamentally tied to the fit between the real system and the regression model. In the vast majority of cases, I believe that the regression model would not be an accurate representation of a system in order to enable claims of causality.

### 5.2.3 Disadvantage of Model-Based Methods

As alluded to earlier, the fundamental flaw with all of the model-based approaches is that they assume that the model fundamentally captures the behavior of the physical process. For example, it is not clear that computational models using automata theory is justified as an accurate representation of the mechanisms at play in the real world. As Polani explains, model-based approaches "make it difficult to allow for a 'natural' concept of emergence to arise from the intrinsic structure of a system" [210, p. 282]. Wolpert and Macready similarly argue that "before a model-driven approach can be used to assign a complexity to a system, one must already fully understand that system (to the point that the system is formally encapsulated in terms of one's model class). So only once most of the work in analyzing the system has already been done can one investigate that system using these proposed measures of complexity. Another major problem with model driven approaches is that they are prone to degeneration into theorizing and simulating, in isolation from the real world" [259]. Instead, they argue that a data-driven approach in general is more likely to be successful.

**Claim**: A model-free approach, being the most generally applicable, is ideal for a design for emergence methodology.

## *5.3 Multi-Scale Methods*

Coarse-graining methods attempt to take a system comprised of many individual elements and to represent the structure or behavior of the system using less detail than afforded by the lowest-level elements. This approach is a natural extension of the successful use of a coarse-graining approach in condensed-matter physics, chemistry, biology, and other natural systems. For example, the cell is often a convenient level that offers more explanatory power than any of its individual components when describing the structure and function of a living organism. In fact, this idea of improved explanatory power when defining levels of abstraction is one of the most fundamental aspects in the concept of emergence. For this reason, coarse-graining methods are theoretically well-positioned to address the question of emergence. However, in practical terms, this formalism is suited for systems with a very large number of components that exhibit spatial or temporal patterns across a wide range of scales. For example, computational biology [223] uses coarse-graining methods to bridge the gap between the atomistic and mesoscopic scales. In the natural sciences, there are several orders of magnitude difference in the number of atomistic and mesoscale elements, as well as in the time scales of the governing dynamics. For the foreseeable future, engineered systems will not approach these scales, which makes coarse-graining methods a poor fit. This approach makes sense for systems that exhibit spatial and temporal patterns and that have many coupled components; however, coarse-graining will not be the correct approach for most classes of complex systems. As another significant hurdle when applying coarse-graining techniques, Weeks [251] demonstrates the explosion in the number of possible coarse-grainings even for a simple system like a cellular automata and the sensitivity of an emergence measure to various coarse-grainings.

Balduzzi [19] applies a coarse-graining technique to cellular automata systems. Emergence is defined as a process that is "best expressed at coarse granularities" [19]. Two information-theory based measures are defined. Effective information captures how selective the system output is given an input. Excess information is extra information generated by the system compared to the sum of the subsystems. A coarse-grained unit is emergent if it has excess information relative to its sub-units and more excess information than its neighboring grains. As the Balduzzi notes, even for a system as simple as a 2D cellular automata, the space of all possible coarse-grainings is vast [19].

Chen et al. [69, 70] use a method of defining emergence in terms of its lower-level constituent components. They propose defining "complex event types" that are composed of simple events types. A complex event type is an emergent property that defines a set of relationships between lower-level components. During a simulation, the occurrence of complex events (i.e., those events that match the set of relationships defined by the complex event type) is used to detect emergence. The fundamental flaw with this approach is that it requires the a priori specification of the emergent behavior. As such, it is only potentially useful for a small subset of complex systems in which the emergent behavior is well-understood. Even with hind-sight, it is difficult to come up with specific relationships which uniquely define the emergent condition. Overall, this approach suffers from a severe lack of generalizability.

Wolpert and Macready [259] argue that the best way to detect emergence is to measure the degree of self-dissimilarity between scales. The dissimilarity between scales is related to the amount of information present between levels. They propose using an information-theoretic approach to measure the dissimilarity of the probabilistic description of the structure of each level.

O'Toole et al. [207] attempt to detect emergence in a distributed manner at the micro-scale. In their framework, each agent observes the correlation between variables describing their state and their observations of the environment. They argue that the presence of emergence can be detected if these correlations come into existence after a period of no correlation. Their approach has each agent tracking the correlation between certain sets of variables using a sliding window time series history. This approaches introduces a number of issues, including the necessity to include some type of consensus algorithm and additional criteria to determine if enough agents are experiencing correlations sufficient to declare emergence for the system. Additionally, this approach focuses on micro-level variables which are generally of less interest to a system designer; rather, the focus should be on the macro-level variables that describe the system behavior. As described by Anderson [7, p. 112], individual agents with myopic views can enter into pathological system wide behavior. Anderson describes the formation of circular mills by army ants. Each individual ant may experience local correlations as they circle endlessly, but the global behavior has entered a fatal scenario.

Dogaru [101] proposes a "local activity theory" for reaction-diffusion cellular [automata] systems where knowledge of the local cell structure improves the likelihood of prediction of emergent behaviors. However, the applicability of this method to general complex systems is not clear and would need to be demonstrated on a case-by-case basis.

## *5.4 Metric-Based Methods*

These attempt to measure the “amount” of emergence present using a quantitative metric based on an observation or measurement of the system. These are of great interest because they provide the potential for us to measure emergence and complexity using a single scalar value and possibly allow for comparisons between different systems. The collapsing of a great number of ideas about what emergence and complexity entails down to a single scalar value brings with it the obvious discussion about whether or not that single value is representative of emergent behavior in general.

These are often “global measures” and can allow us to detect whether emergence may be occurring or not. The downside of a global measure is that it does not give us insight into how or why emergence is occurring. However, during complex system exploratory design, our first objective should be to learn if emergence may happen or not.

### 5.4.1 Complexity-Based Approaches

A number of approaches have focused on using complexity measures (discussed in the chapter on Complex Systems) as a measure of emergence. These measures are often closely related to the complexity measures such as algorithmic complexity, logical depth, statistical complexity, or predictive information.

Hovda [146] tries to determine the amount of simulation needed to derive a fact about a system (whether or not a feature exists). His premise is that “the more simulation required to derive something, the more emergent it is” [146]. Hovda tries to develop a “canonical description” that uniquely and minimally describes the state of the system and the rules for evolution. The amount of simulation is equal to the amount of times of the application of the rules to the micro-elements of the system to reach the desired time step or to reach a state at which there is no further evolution possible. However, this approach is troublesome. Like other approaches based on algorithmic complexity, this is not a scheme that can actually be computed for most real systems. Furthermore, if something takes longer to develop (more iterations), there is no reason to believe that it is more emergent or more interesting. In fact, emergence is most interesting in the intermediate evolution of the system. While something that only takes one time step to emerge might be considered somewhat trivial, on the other hand, something that takes an excessively long time to develop also starts to lose its interestingness and usefulness.

Fuentes [119] uses Gell-Mann and Lloyd's [123] *effective complexity* to define a criterion for detecting emergence. He argues that a property is emergent if the effective complexity measure has a discontinuity at some location in the parameter-space describing the system. However, the biggest problem with this approach is that it relies heavily on Kolmogorov complexity to define the algorithmic complexity of the system. As discussed in the Complex Systems chapter, Kolmogorov complexity is generally uncomputable [82, 229]; therefore, it is difficult to apply to realistic complex systems.

Wright et al. [260] propose a method to use where the complexity of the system is measured using entropy. Emergence is defined as sudden transitions in the system complexity measure relative to smooth transitions in system parameter changes.

The trouble with many of these approaches is that there is not a clear link between high complexity measures and emergence, as was clearly seen with algorithmic complexity. These complexity measures only measure how much work it takes to describe or explain a process but does not actually capture the aspect we are most interested in: the system behavior.

### 5.4.2 Interaction Metrics

Chan [66] uses the metric of interaction counts to detect emergent behavior. However, there are a number of things that interaction counting gets wrong. There's no connection between number of interactions and the system actually becoming more effective. Also, "interactions" are very implementation dependent constructs. Whether something interacts with another entity or not depends on how it is modeled: some interactions are indeed direct connections/interfaces, other interactions can be modeled through interactions with the environment only (e.g., ant foraging), while other interactions are interactions at a distance (e.g., gravity). This approach might work for some systems, but it does not seem general enough for the vast majority of complex systems. In a traffic jam, the number of interactions is simple and scales identically with the number of cars on the road. While it is possible to re-define “interaction” to be when two adjacent cars dip below a critical distance, this shows how this approach is not general and does not capture the subtleties of the interactions.

### 5.4.3 Information-Theoretic Approaches

#### *5.4.3.1 Information Flow*

A number of approaches look at the information flow through the system (or available to an agent) as an indicator for emergence. Chibaya and Bangay [73] investigate the ant foraging system and look at the amount of information an agent uses when making path selection decisions while moving in the environment. They argue that “the amount of information that is made available to agents provides a measure of the degree of emergence to be accomplished” [73]. Johnson et al. [154] define emergence based on the flow of information between various levels in a system. In their formulation, a reduction in the information flow between levels results in less organization which can result in unexpected system states.

#### *5.4.3.2 Discrete Entropy Difference*

Mnif and Müller-Schloer [189] define emergence as the difference between an entropy at the beginning of a process and at the end. A self-organizing process is emergent if $\Delta\mathrm{H} > 0$, where $H(x)$ is the entropy with respect to the observed attributes of a system. This formulation restricts measuring emergence to just between two points in a single dimension: time.

$$\Delta H = H_{start}(x) - H_{end}(x)$$

#### *5.4.3.3 Information Transformation*

Gershenson and Fernandez define emergence as “information at a higher scale that is not present at a lower scale” [124]. Emergence, $E$, is simply defined as the information that a system produced relative to the information received: $E = \frac{I_{out}}{I_{in}}$. The transformation between the information out and the information in can be dynamic evolution of a process, or a change in scale. They define self-organization, $S$, as as the difference between the information the system received and the information produced: $S = I_{in} - I_{out}$. Complexity, $C$, is the product of emergence and self-organization: $C = E \times S$. Assuming random inputs, they claim that emergence is the opposite of self-information [124, p. 37]. I do not feel that these definitions accurately capture emergence and self-organization. These oversimplified definitions are misleading and do not reflect complex behavior.

#### *5.4.3.4 Persistent Mutual Information (PMI)*

Ball et al. [21] use predictive information to detect emergence in a time series. They use the mutual information between the past and future history which persists across an interval of time $\tau$ . Persistent Mutual Information (PMI), $I(\tau)$, is the relative entropy of the joint history compared to that of past and future taken independently, as shown in Equation 5, where $x_{-0}$ are the observations up to the present time 0, $x_{\tau+}$ are the observations from time $\tau$ onward, $p\left(x_{-0}, x_{\tau+}\right)$ is the joint probability density of the past and future, and $p\left(x_{-0}\right)$ and $p\left(x_{\tau+}\right)$ are the marginal distributions of the past and future, respectively.

$$I(\tau) = \int \log\left(\frac{p\left(x_{-0}, x_{\tau+}\right)}{p\left(x_{-0}\right) p\left(x_{\tau+}\right)}\right) p\left(x_{-0}, x_{\tau+}\right) dx_{-0} dx_{\tau+} \tag{5}$$

This is a measure of how much of the future can be predicted from the past, and thus measures the difficulty in predicting the future based on the past. It is important to note the inclusion of the $\tau$ parameter, which captures only correlations which exist across time scales greater than $\tau$. One important limitation of this approach is the difficulty of choosing the value of this parameter *a priori* and without biasing the outcome. At the extreme values, $I(0)$ is equivalent to Bialek's predictive information [41]. Ball et al. also define $I(\infty)$ as Permanently Persistent Mutual Information (PPMI). They claim that this is the best measure of emergence, and reflects the "the degree of permanent choice spontaneously made by the system" [21]. This approach is limited to a single dimension, either in time, space, or another causal dimension. Ball et al. use the k-Nearest Neighbor (k-NN) approach of Kraskov et al. [162] to estimate the entropy.

#### *5.4.3.5 Divergence-based Measures*

Fisch et al. [114] define emergence as "an unexpected or unpredictable change of the distribution underlying the observed samples." Although the Kullback-Leibler (KL) divergence is generally the most commonly used divergence measure in information-theoretic formulations, they propose to use the Hellinger distance (Hel) due to its easier implementation. The

Hellinger distance is shown below in Equation 24 where BC is the Bhattacharyya coefficient, $\mathrm{BC}\left(p,q\right) = \sum_{x\in\mathbf{X}} \sqrt{p\left(x\right)q\left(x\right)}$ .

$$\mathrm{Hel}\left(p,q\right) = \sqrt{1 - \mathrm{BC}\left(p,q\right)} \tag{6}$$

They find that the Hellinger distance is superior to the discrete entropy difference of Mnif and Müller-Schloer [189] for detecting emergence. They used a Parzen window kernel density estimator to estimate the probability density functions. They estimate $p$ and $q$, two distributions of samples from a dynamic process, in a sliding data window with a fixed length in time.

MacKay introduces the concept of a space-time phase, which is the probability distributions for state of a system as a function of space and time which evolved from an initial distribution [98]. Emergence is defined as the distance of the space-time phase to the set of products for the probability distributions of the independent units. He uses the Dobrushin metric to measure the distance between probability distributions. In this approach, the measure is indicative of how the behavior deviates from the mean-field approximations. This approach is an improvement over previous divergence-based approaches because it includes the additional spatial dimensions.

The approach of Fisch et al. represents the best candidate for developing a measure of emergence for use in the methodology in this thesis. Most importantly, it is general enough to be applied to any complex system. It is also well suited towards design space exploration; although the original formulation looked at a stochastic time-series, it is easily adapted towards evaluation of stochastic responses at various design points.

### 5.4.4 Measures for Complex Networks

Complex networks are also a significant area of study. Boschetti et al. [52] argue for using information-theoretic measures for measuring the correlations between components in a complex network as a means of detecting emergence. Moncion et al. [190] use a graph-theoretic approach to identify nodes that form clusters of strongly interacting components within a self-organizing complex network. For them, the dynamic formation of these clusters constitutes emergence. Both of these approaches analyze the topology of the complex network and are therefore only concerned with the structure of the system. Neither of these approaches are capable of measuring the global behavior or performance of the system.

## *5.5 Other Approaches*

There are several approaches that do not fit neatly into the above categories. Ronald et al. [215] approach the problem from the perspective of Artificial Life, where emergence has often been seen in a subjective way and open to interpretation. Their criteria for emergence is based upon the degree of surprise that an observer experiences based on knowledge of the system's design. They define a language of design, $L1$, and a language of observation, $L2$. The degree of surprise in observing $L2$ given $L1$ becomes the criteria for emergence. This approach has two issues. The first is the subjective view of emergence, which I've argued against in the chapter on Emergence. The second is the difficulty in actually applying this test to realistic systems. Artificial Life has typically dealt with toy problems which may be amenable to this type of analysis; however, this approach would not be feasible for engineered complex systems.

Moshirpour et al. [197, 195, 196] propose a method of detecting emergence by searching the system definition for *implied scenarios*. Implied scenarios occur due to indeterminism, where the rules governing the behavior are not specified for certain scenarios. This method is basically a requirements analysis to make sure that no ambiguities are specified in the system definition. However, implied scenarios are just a small subset of the causes of emergent behavior. Additionally, searching the scenario space for any practical system is infeasible. Fully defining a system for every scenario becomes an impossible task. Bar-Yam argues that the "explosion of interface specification" [25] becomes overwhelming in a complex system. This method also goes against the goal of self-organizing systems, which are purposefully left under-determined so that there are additional degrees of freedom left for adaptation. The authors' goals were to eliminate the surprise of emergent behavior during system development, so their work approaches the problem from a different perspective than many of the other approaches presented here.

Reconstructability Analysis (RA) is an analysis method based on information theory that attempts to identify the relations between variables (input and/or output) in a system. In the analysis of emergent behavior, the goal of the analysis is to identify the microlevel variables that are responsible for the generation of the macro effect [238]. The simulation is executed and the values of the micro and macro-level variables are recorded. Reconstructability Analysis is essentially a model building procedure between inputs and outputs. However, RA is not an emergent behavior identification approach. It is an analysis method to help identify the important relationships once a candidate emergent behavior has been identified. It is much more appropriate as a verification method, where a candidate emergence condition is tested to substantiate emergent behavior. This approach attempts to identify causality and explanation for the underlying behavior.

Gore et al. advance an approach called Explanation Exploration (EE) which attempts to understand unexpected model behavior [130]. Explanation Exploration allows for both structural and parametric explorations of the system models. However, like reconstructability analysis, it is up to the subject matter expert to identify candidate emergent behavior. Therefore, EE is an automated "observed simulation behavior hypothesis testing that allows users to validate or reject emergent model behaviors efficiently" [129]. Gore's approach uses directed acyclic graphs and probability theory to infer causality. Part of the method uses *program slicing* to decompose the simulation source code to identify which statements affect the target variable [129]. However, I would argue that this is not a useful approach. This approach ignores self-generated complexity where simple rules lead to complex behaviors. As in chaotic maps in nonlinear dynamics, some simple mappings can lead to very intricate/chaotic behavior. Analysis of the map does not make this complexity clear. Furthermore, causality cannot be assigned to a small set of elements or features or line of codes—the true causality is due to the myriad individual contributions and interactions. Furthermore, these dependencies and sensitivities are context dependent and are only true for the particular conditions observed, not for the system in general.

Boschetti [49, 50] takes an anomaly detection approach toward detecting novel behaviors. In his approach, a time series data set is embedded in a feature space, a similarity measure is defined and then various behaviors are identified using classification/clustering techniques.

Data points which fall outside of the clusters identified represent possibly qualitatively different behaviors. The fundamental aspect of all anomaly detection methods is the definition of typical behaviors and then classifying whether candidate behaviors are typical or not.

### *5.6 A Common Theme: Critical Transition*

One of the common themes in a number of the measures of emergence presented above, and within complexity science, is the idea of a critical transition as an indicator of emergence. As discussed in the chapter on Emergence (Chapter 2), an informal definition of emergence is the appearance of *qualitatively* different behavior. This implies that there is a transition between regions of *similar* behavior and *different* behavior (or properties). Batterman has argued that phase transitions in condensed matter physics represent emergent behavior [29]. When studying the behavior of cellular automata (CA), Langton proposed the idea of "edge of chaos" [167]. He argued that "CAs exhibiting the most complex behavior—both qualitatively and quantitatively—are found generically in the vicinity of this phase transition. . . . Most importantly, we observe that CAs in the transition region have the greatest potential for the support of information storage, transmission, and modification, and therefore for the emergence of computation" [167]. Although the specifics of his results have been criticized, the underlying concept of a transition region as a useful indicator of emergence remains. Vicsek similarly argued that systems operating on the edge of chaos may "be a general property of systems that are capable of producing interesting (complex) behaviour" [246]. The field of catastrophe theory attempted to identify qualitative shifts in systems due to (small) changes in parameters. Although catastrophe theory did not prove to be a useful and long-lived field of study, it did provide a useful metaphor for understanding the behavior of complex systems [234]. Di Marzo Serugendo et al. [97] similarly argue that self-organizing systems are observed after a critical transition. Grossman et al. [132] define emergent behavior as a changing point in a time-series and can be found using detection algorithms. Wright et al. define emergence as "sudden transitions in the [system complexity] measure, relative to smooth changes in system parameters. Emergent behaviours are those whose parameter values are associated with these abrupt transitions in the [system complexity]" [260].

**Claim**: The proposed measure of emergence should enable the detection of critical transitions in behavior.

Another aspect of the critical transition is that it provides the potential for a larger set of possible outcomes. Minai et al. argue that the one of the fundamental issues in the complex systems engineering is the search for "solution-rich configuration spaces" [188]. Similarly, Kernstine writes that "these transition regions and areas of rapid change are of specific interest when investigating [system of systems] simulations" [156]. Finding emergence requires identifying critical conditions for its manifestation. The key idea is that, at the critical conditions for emergence, changes in the system configuration or parameters will produce a corresponding shift in behavior. Therefore, searching for emergence can be reduced to comparing behavior at neighboring points in the design space.

## *5.7 Proposed Quantitative Emergence Measure*

**Identified Gap 2**: Few of the measures of emergence would be appropriate for use in a design space exploration methodology. Divergence-measures have been applied to identify emergence in time-series data but there exists a need to extend the approach to design space exploration.

Based on a review from the literature about different methods to identify emergence in a complex system, divergence-based measures are most appropriate to meet the needs of this work. The lack of connection between algorithmic complexity and emergence as well as its difficulty to apply to general systems means that algorithmic complexity based measures are not useful. Approaches based on symbolic dynamics and computational mechanics have a plausible relation to emergent behavior; however, it is not clear that studying the discrete automata representation of a process actually represents a meaningful distinction when designing a complex system. A metric-based approach is most general and practical approach for a design space exploration. A divergence-measure based on information theory is general enough to be applied to a wide range of problems and retains the deep connection to complex systems and emergence. However, almost all applications of divergence measures in complex systems have focused on time-series observations and have not been applied to design space exploration.

The definition of engineered emergence from the chapter on Emergence (Chapter 2) is reviewed below. This has similarities to the definition is given by Fisch et al., "emergence is regarded as an unexpected or unpredictable change of the distribution underlying certain observations of the system" [115]. The problem with this definition, at least in terms of being a quantitative measure, is the ability to address the unexpectedness or unpredictability of the change in the distribution. Applying this definition as stated would require a way to capture the expectedness or predictability of an observation. Although this may be possible through the creation of a model that captures the *expected* behavior, doing so would be fraught with issues. This model would necessarily be incorrect; therefore, any conclusions drawn from it would not be justified. Like the concept of observer-dependent emergence (see Section A.2.6 in the chapter on Emergence), this approach would make the emergence relative to a model and would not capture the intrinsic emergent functionality. To get around this issue in the emergence measure used within this research, there will be no explicit model to establish the expectedness of any particular observation; rather, the change of the distribution will be measured relative to the *current* observation. Therefore, the *expected* model is implicitly assumed to be the same as the current observation in a local neighborhood.

**Engineered Emergence**: Engineered emergence is the critical transition in macro-level behavior due to changes in system context (i.e., environmental conditions or system parameters).

The "transition in macro-level behavior" in the definition of engineered emergence is captured by the change of the underlying probability distribution from the current observation to the next, as measured by the statistical distance between them. In order to make the quantification of engineered emergence complete, the engineered emergence measure needs to capture the second part of the definition (i.e., "due to changes in system context") as well as the idea of a critical transition. Therefore, the proposed measure considers not only the statistical distance between two probability distributions but the distance between the system parameters corresponding to each observation.

The combined terms are shown in Equation 27. This expression will be called *design space divergence* (DSD) and is the proposed measure of *engineered emergence*. As discussed in the above section, an emergent design point represents a transition of a system-level measure relative to (small) changes in the system parameters. For small changes in system parameters, the system behavior is not expected to change in non-emergent regions of the design space. The corollary, and more salient idea, is that emergent behavior is expected to be manifested as a large change in the probability distribution due to small changes in the system parameters.

**Hypothesis (HYP1)**: Assuming a desired range of system behavior has been identified, large divergence measures of the distributions in the response variables reflects possible emergent conditions near critical conditions while low divergence measures reflect robust design points.

$$\text{Engineered Emergence} \equiv \text{Design Space Divergence} = \frac{\text{Statistical Distance}(\mathrm{P}, \mathrm{Q})}{\text{Design Space Distance}(\mathbf{x}_\mathrm{p}, \mathbf{x}_\mathrm{q})} \tag{7}$$

It is obvious that this proposed measure for emergence does not capture every definition of emergence, such as the property of irreducibility or of various levels of representation. However, from a practical standpoint, this measure becomes much more useful in for engineering, where system function is the primary goal. As Kernstine argues, “the only emergent phenomena that are important are those that influence the metrics of interest in the specified amount of time” [156, p. 62]. Like Müller-Schloer and Sick [198], I argue that a loss in philosophical rigor in the definition of an emergence measure is more than made up for by the increase in generality and practicality. The designs that this measure will identify can be the starting point for a more rigorous evaluation to determine the emergent nature of the behavior. Thus, Design Space Divergence is appropriate for use in the methodology developed in this research.

# CHAPTER VI

# MULTI-AGENT SYSTEMS

**Chapter Road Map**: The goal of this chapter is to introduce the distributed multi-agent system being studied within this research. The multi-agent surveillance will serve as an example of the type of complex systems that motivate this research. The goal of this chapter is to demonstrate how to perform a design space exploration on a multi-agent complex system with the goal of characterizing the design space in order to identify how the system changes behavior as the design variables change. The methodology developed in this thesis will be demonstrated on this problem.

## *6.1 Agent-Based Modeling*

Simulation offers an approach to address posed by multi-agent systems. Simulating these systems under a large number of conditions will help to characterize the behavior of the system and identify critical design variables. Agent-based modeling (ABM) is a commonly used approach to study complex systems that can be represented as the interaction of agents with each other and the environment. Like cellular automata (discussed in Chapter 2.3.3), agent-based models are predicated on the assumption that complex behavior arises from the simple rules applied at the micro-level [150]. However, unlike cellular automata, where cells are static and have fixed relationships with their neighbors, agents are often mobile and whose interactions with other agents and the environment are much more dynamic. This additional freedom makes agent-based models much more usable than cellular automata and applicable to a much wider range of problems.

One of the reasons that ABM is popular, especially in models dealing with social sciences, is that this modeling approach allows an ontological correspondence between the agents and the real world elements that they model [125]. This allows the agents and their interactions to be modeled directly, rather than relying on having to construct a system of equations to describe the dynamics of the system. This is especially important in the case of heterogeneous agents, where each agent has its own different properties and operating rules. However, although there is an ontological correspondence between the model and the real world, agent-based modeling is rarely an accurate representation of the real world capable of making predictions [14]. Because of this inability to make predictions, validation of agent-based models is a particularly difficult problem. Because of this difficulty (and the general inapplicability of predictive methods to complex systems), agent-based models are used as an exploratory tool. The goal is to understand the rules and conditions at the agent level that yield complex behavior at the system level [150, p. 566].

## *6.2 Multi-Agent Surveillance Simulation*

The simulation used in this research is inspired by the Multi-Agent Surveillance Simulation (MASS) studied by Aksaray [1]. In this simulation, multiple unmanned aerial vehicles (UAVs) patrol a set of waypoints and relay information back to the base using a distributed peer-to-peer communication network. Information about the targets located at the waypoints is sent back to base via multi-hop communications as long as a connected network can be established between the transmitting UAV and the base. The aircraft have limited endurance and must return to base for refueling which leads to reduced coverage of the patrol area and possible disruptions to the communication network if critical communication links are removed. This problem is an example of a Mobile Ad-Hoc Networks (MANET).

The original purpose of the simulation was to examine the effects of vehicle design parameters and control strategies on being able to maintain surveillance coverage of the targets. The simulation was part of an overall methodology examining the effects of both vehicle design variables and various control strategies during the design process [1]. The simulation varied vehicle design variables such as the maximum velocity, fuel capacity, communication range, fuel consumption ratio of low altitude to high altitude loitering, and a time required to detect a threat (a measure of sensor capabilities). The simulation also examined control strategies for varying UAV replacement strategies in order to maintain the communication network when disturbed as well as the UAV refueling strategies for determining when a vehicle would choose to return to base to refuel. The simulation used identical UAVs and a single control base. The UAVs were tasked with monitoring a set of waypoints, represented by a graph of nodes which the UAVs moved among. The system effectiveness measure was based on a the amount of time that each target was monitored.

### 6.2.1 Limitations of MASS

While the Multi-Agent Surveillance Simulation provided a good starting point for demonstrating the design of a complex system and testing the methodology presented in this thesis, it lacked several key characteristics. Most importantly, while the original simulation had some aspects of complexity, the model needed more agents, more interactions between agents, and more stochasticity to be representative of the types of problems this thesis is motivated by. The original Multi-Agent Surveillance Simulation also focused heavily on examining the effects of various control strategies. These schemes are inherently binary in nature (i.e., they are either on or off), while the problems motivating this research are typically governed by a number of parameters. Therefore, the simulation used for this research will use a single strategy. The focus will be on the parametric design variables rather than control strategies. The control strategies used in MASS looked at UAV replacement in order to maintain the communication network and UAV refueling strategies for determining when a vehicle would choose to return to base to refuel. The results from the simulation showed that the randomized UAV refuel policy was not a significant factor in determining the overall system performance [2]. Therefore, only the deterministic refuel policy will be kept for the simulation used in this thesis. The other strategy MASS considered was how to treat UAVs that were removed from the system in order to refuel. The replacement strategy used a local neighbor message passing system to shift assigned waypoints in order to maintain the communication network. However, a “no replacement” strategy will be used for the UAV replacement policy in this research. The simulation will be modified to be more dynamic and stochastic, making a replacement strategy much less useful. The goal is to design a system which is more inherently robust to local disruptions and can maintain connectivity without explicit replacements taking place. As will be detailed later in this chapter, the agent rules implicitly take care of any agents which need replacement.

### 6.2.2 Emergent Behavior by Analogy

A short aside is necessary to avoid getting caught in a philosophical debate. Defining what constitutes an emergent behavior is a controversial and context-dependent task. The difficulty in defining and identifying emergence was discussed in depth in the chapter on emergence. But the philosophical challenges do not have to be a toll gate towards making progress in understanding complex systems. There are a number of natural systems where emergent behavior has been studied and there exists a body of literature supporting the claims of emergence. In particular, ant foraging has been widely studied and is often accepted to be a natural complex system with emergent behavior [135, 177, 204, 203]. This design problem seeks to identify the set of parameters (e.g., number of ants, pheromone evaporation rate, pheromone diffusion rate) for which the ant foraging system exhibits emergent behavior. The emergent behavior, in the context of ant systems, is the set of conditions which leads to the formation of stable pheromone trails which results in an efficient collection of food from the environment. For many other distributed systems, what constitutes emergent behavior is less clear. However, explicitly defining the emergent behavior is not necessary. By generalizing the type of behavior seen in the ant foraging problem, we can utilize the techniques presented in this thesis for a wide class of problems that share similar modeling and behavioral characteristics. By avoiding the philosophical debate about whether a behavior is truly emergent or not, the proposed methodology will be useful regardless of the status of emergence in any specific complex system. Rather, it is useful to show that the multi-agent surveillance problem has the structure of a problem in which emergence may take place. It has the defining feature of self-organizing systems that agent rules are defined at the local level only. The problem also has micro- and macro-level states where the micro-level consists of the agents and their individual states while the macro-level contains the connected network. There are feedback loops between the micro- and macro-levels. As Ilachinski explains, “in order to properly understand complex systems, such systems must be viewed as coherent wholes whose open-ended evolution is continually fueled by nonlinear feedback between their macroscopic states and microscopic constituents" [150, p. 558]. The upward causality exists since the macro-level network consists of micro-level agents. The downward causality is the influence of the network on the agents at the micro-level. This demonstrates that the multi-agent surveillance problem has the characteristics of a complex system that is likely to exhibit self-organizing, emergent behavior and is thus a good test problem for this research.

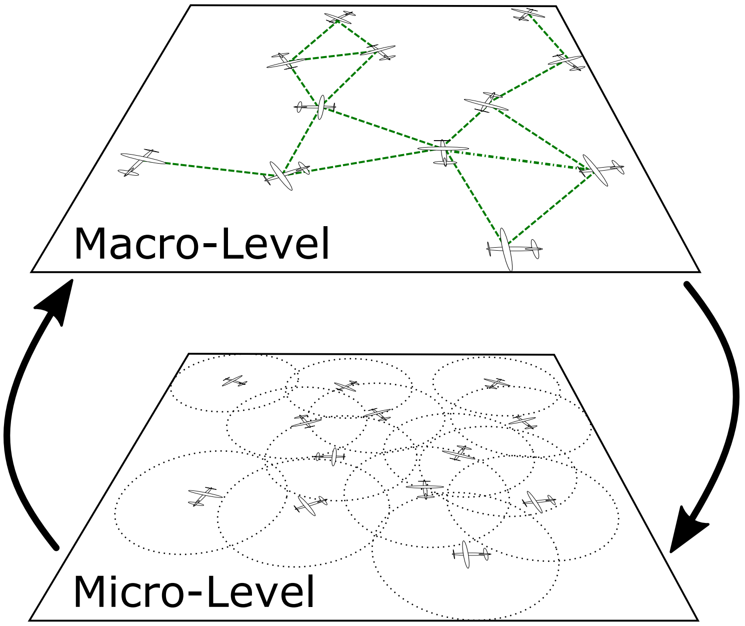

**Figure 13:** Multiple levels in multi-agent surveillance system problem

Using the analogy of emergence and the similarity to the ant foraging problem, a working definition for emergent behavior can be defined for the multi-agent surveillance system. Emergence will be defined as the conditions which lead to the creation of a self-organized and robust communication network that results in high system performance. Of particular interest are systems that can yield high performance for relatively low capability UAVs (i.e., improved situational awareness with decreased technological capability). Like the ant foraging problem, the goal is to find simpler systems (with fewer rules) that lead to high overall system capability. In essence, the goal of this case study will be to determine if a distributed surveillance system can be designed in which the network effects of a large number of UAVs create additional capability that would not be expected otherwise.

We can generally expect monotonically increasing system performance metrics as we increase the number of vehicles, increase vehicle performance, and increase sensor and communication capabilities. Therefore, we're not interested in the design points which are globally optimal, since these points are often uninteresting. More importantly, we're interested in the design points which lead to the most significant shift in behavior. Critical points are common in complex systems: phase transitions [29, p. 214], bifurcations and chaotic behavior [79], edge of chaos [167], self-organized criticality [18], etc. All of these capture the idea that most complex systems have critical points where we have qualitatively different behavior. The goal is to identify the location of the critical points. Likewise, our goal for identifying emergent behavior in the multi-agent simulation is to find the design points which are the most critical in determining system behavior.

## *6.3 Distributed-MASS*

Much of the research regarding complex systems is motivated by the complex, emergent behavior seen when agents are allowed to evolve according to a relatively small set of simple rules. Following this guiding principle of keeping the simulation as simple as possible results in a number of benefits. It improves the ability to re-create these results, which is a keystone of the scientific process and repeatable experiments. It also simplifies the model building process and allows for rapid execution of the simulation. And lastly, by keeping the governing rules as simple as possible, it makes the results more compelling and improves confidence that the observed behaviors are a consequence of the governing dynamics and not simply a fluke of the implementation or one of a myriad possible lines of logic in the code. This is similar to Fromm's argument, "the phenomenon of emergence is better comprehensible and understandable for simple agents, and in fact it is most useful for a large number of small and stupid elements. The context-dependent influence is certainly stronger for simple, stupid and purely reactive agents, and weaker for more complex, intelligent, proactive and goal-directed agents" [118].

With this set of considerations in mind, the Multi-Agent Surveillance Simulation was re-formulated into the Distributed Multi-Agent Surveillance Simulation (Distributed-MASS). The purpose of the simulation is to measure how system design variables affect overall system performance. The simulation has four main classes of components: the environment, the base, the targets, and the agents. Each will be described in the subsections below. The system design variables as well as the metrics used to evaluate the system performance will be discussed below.

### 6.3.1 Environment

The environment is the simplest component of the simulation. The environment consists of a two-dimensional plane in which the targets and agents are free to move. The size of the environment is defined by the user. For all of the simulations performed in this thesis, the size of the environment is assumed to be a $100 \times 100$ unit square. The bounds of the environment are treated as fixed walls. Collisions between the targets or agents and the walls are modeled as elastic collisions.

Time is treated at discrete steps in order to make the problem tractable. However, space is a continuous variable with the agents and targets able to move to any location within the environment. Space could have been treated discretely to create a Cellular Automata (CA) implementation. CA is a popular modeling technique popular for studying complex systems; however, a continuous space treatment is simply a generalization of the CA modeling approach [150]. Therefore, this choice should not be regarded as a fundamental distinction, but rather as a matter of convenience.

### 6.3.2 Base

The base is a single entity located at a specified point in the environment. For all of the simulations performed in this thesis, the location of the base is assumed to be at the coordinate position $(10, 10)$. Agents start the simulation at this location and return when they need to refuel. The agents communicate with the base the location of all of the targets that they are tracking. The goal of the system is to relay real-time information about all of the targets back to the base. Therefore, the system performance is measured by the amount of real-time information that the base receives. Like the agents, the base has its own limited range in order to establish a communication link. This communication range is a design variable; however, for the simulation results shown, the base's communication range is assumed to be the same value as the agent's.

### 6.3.3 Targets

The targets are the system components that the agents are trying to locate within the environment. The targets' behavior is generally quite simple. They are initially at random locations within the environment. At every time step, the target takes a random step within the environment. The target has a specified maximum velocity, which limits the maximum distance traveled between time steps. The targets do not have knowledge of other targets or the agents that are trying to follow them, so their trajectory is simply a random walk through the environment. Each target keeps track of its own position and a direction vector, which is a unit vector that represents the direction the step will be taken. To make step random, the direction vector is rotated at every time step using a random variable. This rotation angle is a normally distributed random variable with a mean of zero and a user-defined standard deviation. The user-defined standard deviation controls the erraticness of the target's movement—a small standard deviation leads to relatively straight trajectories while a larger standard deviation leads to more dramatic direction changes. A representative trajectory is shown in Figure 14.

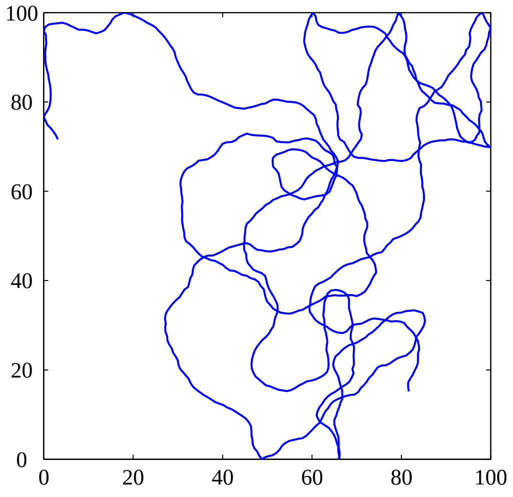

**Figure 14:** A representative target trajectory

### 6.3.4 Agents

The agents have the most complicated dynamics in this simulation. Each agent represents an Unmanned Aerial Vehicle (UAV). The agents have a number of design parameters that can be set by the user. All of the agents are assumed to have the same design parameter values (i.e., they agents are assumed to be identical). A list of agent design parameters is shown in Table 3.

**Table 3:** Agent design variables

| Agent Design Variable | Baseline Value | Units |
|---|---|---|
| Maximum Fuel Capacity | 4500 | fuel-unit |
| Maximum Velocity | 3 | distance-unit per time step |
| Target Detect Radius | 20 | distance-unit |
| Communication Radius | 35 | distance-unit |
| Fuel Consumption Rate (Cruise) | 30 | fuel-units per time step |
| Fuel Consumption Rate (Tracking/Loitering) | 30 | fuel-units per time step |
| Fuel Consumption Rate (Relay) | 15 | fuel-units per time step |

At the start of the simulation, the agents start at the base. Each agent proceeds to search the environment until a target is located within its detection range. If an agent locates a target, it will proceed to follow the target for as long as its endurance allows. Once the agent reaches a critical fuel level, it changes objectives and returns to the base location to refuel. When it is returning to base, it can communicate with other nearby agents and with the base; however, it cannot detect or track any targets. When it is done refueling and is ready to return to searching, it is assigned a new waypoint based on the last tracked location of an unassigned target. When the agent is traveling to this waypoint, it is set to a cruise mode where it can communicate with other agents but it cannot track new targets. Once it reaches its assigned waypoint, it will begin tracking. The agent can be in one of five operating modes, described below. These modes control the behavior of the agent throughout the course of the simulation.

**Refuel Mode (Mode 0)** At every time step, the agent evaluates how much fuel it would take to return to base. Once a critical fuel level is reached, the agent switches to refuel mode. During this mode, it moves in a straight line path towards the base. During this return trip, the agent can communicate with other agents and the base itself, but it cannot track new targets. Once it returns to base, it is immediately returned to full fuel capacity and it is able to return to the field at the next time step.

**Cruise Mode (Mode 1)** Cruise mode is used when the agent is given an initial waypoint assignment after refueling. The agent will proceed to the assigned waypoint in a straight line path. It can communicate with other agents but it cannot acquire or follow new targets until it reaches the waypoint.

**Tracking Mode (Mode 2)** When the agent detects one or more targets, it will switch in tracking mode. In this mode, it will move towards the nearest target. It will communicate with other agents and the base if they are within communication range. Information is exchanged between all connected agents on the location of targets within their target detect radius. Connected agents will attempt to deconflict among themselves the tracked targets. An agent will not track the same target that another connected agent is tracking.

**Search Mode (Mode 3)** When the agent does not detect any targets and there are no untracked targets among any connected agents, the agent switches to search mode. In this mode, the agent takes a random step in the environment at each time step. It uses the same random step method as described above for the targets. If the agent detects a target, it will switch to tracking mode. If no targets are detected, it will proceed in search mode until it reaches a critical fuel level and switches to refuel mode.

**Communication Relay Mode (Mode 4)** If the agent is in search mode and becomes a critical communication node between the base and other agents, it will enter into communication relay mode. In this mode, the agent holds position until it is no longer a critical communication node or until it detects an untracked target in its detection range. Since the communication network is an important part of the simulation, more details regarding this mode will be discussed later in this section.

A representative time history of a set of agent's modes is shown in Figure 15. The fuel consumption rate for an agent in communication relay mode is set to be lower than an agent searching for or tracking a target since an agent in relay mode should be able to loiter at an energy efficient speed. Because of this reduced fuel consumption, agents acting as communication relays have longer endurance. As shown in the Figure 15, the agents which spent the majority of their time as communication relays only had to refuel twice in this time window while the agent which was actively tracking a target had to refuel three times. This figure also shows the coupling of modes between agents. As some agents leave the search area to refuel, other nearby agents have to switch modes to take over tracking and relay duties. As discussed earlier, the Distributed-MASS model does not include an explicit agent replacement policy; however, agent replacement is implicitly handled by agents switching operating modes based on their knowledge of target locations and network connectivity. The rules prioritize tracking a target over maintaining a communication network. If the sensing range is small and search time is a significant factor, the best approach is to exploit any knowledge of the target distribution [108, p. 2093] and to prioritize following a target once it is found. Since finding the target is difficult, preference is given to tracking it at the expense of network connectivity.

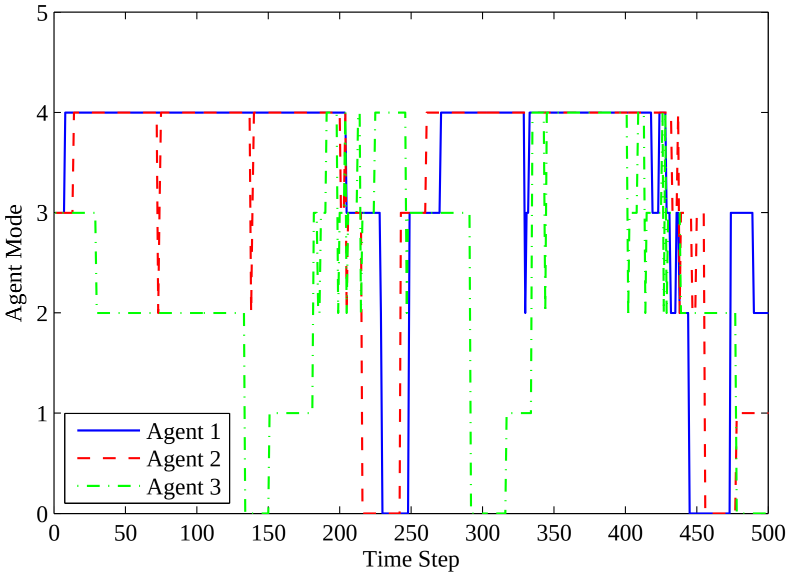

**Figure 15:** Agent mode history

#### *6.3.4.1 Agent Search Strategy*

As explained earlier, the agents use a random search strategy for locating the targets in the search space. Various patrolling strategies exist that attempt to optimize the agent's trajectories in order to search the space most effectively. The strategies generally fall into three categories: cyclic-based, randomized, and partition-based strategies [1, 72, p. 11]. The randomized search strategy was used because it is most consistent with a distributed and decentralized system. It is also the most uncertain and the most straightforward to implement. A cyclic-based approach would likely not be appropriate since defining a search graph would eliminate the strongly decentralized and autonomous nature of the system. A search graph would have to be defined a priori and it would not correspond well to the system being studied in which there are no well-defined waypoints or paths to take (e.g., roads). Partition-based approaches require centralized control since no decentralized algorithms exist for dynamic partitions [108]. Like a cyclic-based approach, a partition-based search created by a central controller would overly constrain the behavior of each agent to the point where self-organizing behavior becomes unlikely.

### 6.3.5 Communication Network

The communication network formed among the agents and base is the mechanism which governs the global behavior of the system. Since the communication network is a critical component of the simulation, it will be described in detail in this section. Every agent has a communication radius. If another agent is within that communication radius, the two agents can communicate with each other. As shown in Figure 16, Agent A can communicate with Agent B but not with Agent C. There is no limit to the number of connected agents. Agents can form a communication chain; if Agent C can communicate with Agent B, and Agent B can communicate with Agent A, then Agent C and Agent A can communicate with each other. A single value for the communication range is used for all agents. This simplifies the analysis so that every communication link is bi-directional (and the resulting graph is undirected). In deployed systems, communication systems are often subject to communication issues such as obstruction, interference, jamming, and limited range. However, in this research, it is assumed that as long as two agents are within the critical communication range, any communication between them is perfect.

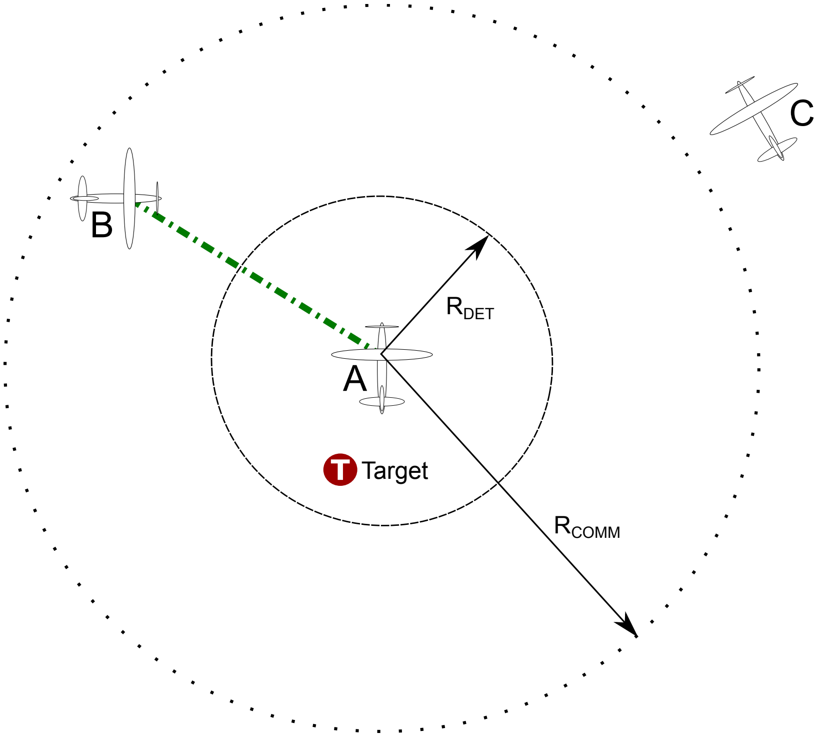

**Figure 16:** Communication among agents

The content of the communication between agents is the list of targets that are within each agent's detection range. By sharing target information, every connected agent knows the location of every target that is within the detection range of any of the connected agents. In Figure 16, a target is within the detection radius ($R_{DET}$) of Agent A. Therefore, both Agent A and Agent B know the location of the target.

However, the ultimate goal of the system is to relay the target information back to the base. In order for this to happen, the base must be within the communication range of an agent that has real-time target information. Since agents have limited communication range, the most effective system is one in which the agents form a communication chain to link agents which are actively tracking targets to the base. However, forming a multi-hop network in a dynamic environment without centralized control is a challenge. Each individual agent has limited information from which to decide which course of action to take. Specifically, an agent has to decide whether it should search for new targets or whether it should stay in place in order to form a communication chain. The agent performs a relatively simply check to determine whether it is a critical node in the communication chain (i.e., if it is removed from the network, a path between another connected agent and the base does not exist). If the agent is determined to be a critical node and it is not currently tracking a target, then it will switch to communication relay mode (described in the section above). In this mode, the agent will stay in place as long as it remains a critical node and as long as it does not detect a target. In order to determine if the agent is a critical node, it checks to see if two conditions are met: 1) the base is one of the connected nodes, and 2) whether the agent is part of the shortest path between any pair of 1-hop neighbors. If the agent itself is part of the shortest path, then it is a critical node between the neighbors. This check would be simple to implement in a real system and would not violate any assumptions of a completely distributed system. An agent will query each connected neighbor to return the list of each of neighbor's list of neighbors. For example, examining the system shown in Figure 17, Agent A has neighbors B, D, and E. Examining the neighbor pair $(D, E)$, Agent D can communicate with Agent A and Agent E can communicate with Agent A, but Agent D cannot communicate directly with Agent E. This configuration leads to Agent D being able to communicate with Agent E only through Agent A. This means that Agent A is part of the shortest path between Agent D and Agent E and is therefore a critical node. Therefore, if Agent A is not tracking a target, it will enter into communication relay mode. For the system represented in Figure 17, Agents A, B, and D are all critical nodes.

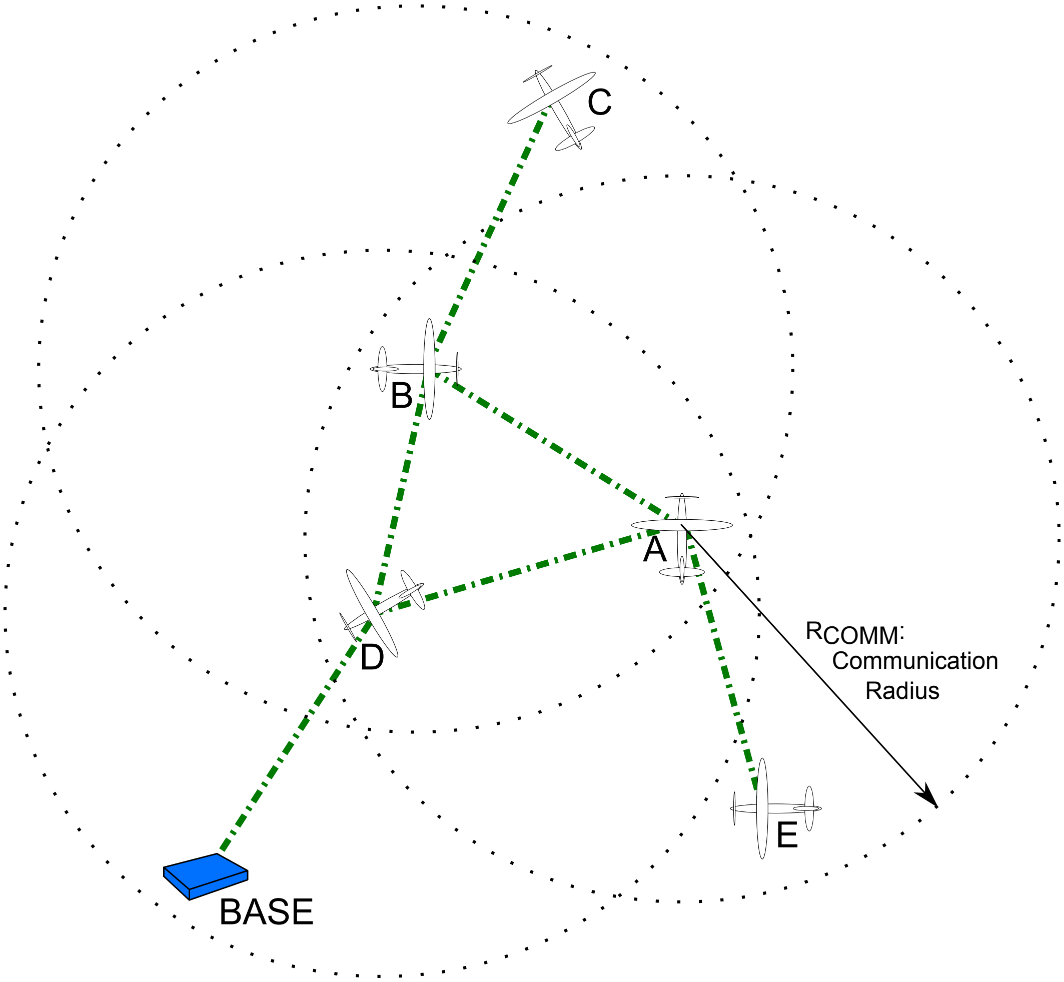

**Figure 17:** Multi-hop communication network

### 6.3.6 System Measures of Performance

In order to evaluate the performance of the system, several high-level measures are introduced. One of the measures of system performance is the Total Target Track Count Percentage (TTTCP) . This is the percentage of time that the targets were tracked during the course of the simulation. A low TTTCP value indicates that the agents were not successful in finding and tracking the targets while a high value indicates that the agents were able to track the targets during a large portion of the simulation. This metric only considers whether or not the targets were tracked by the agents and does not consider if the information was relayed back to the base. Therefore, it can be considered a metric of the aggregate vehicle performance while ignoring the overall system performance. It is calculated as shown below in Equation 8.

$$\text{TTTCP} = \frac{1}{t_{end}} \frac{1}{n_{target}} \sum_{t=1}^{t_{end}} \sum_{i=1}^{n_{target}} \tau_i\left(t\right) \tag{8}$$

Where,

$$\tau_i\left(t\right) = \begin{cases} 0 & \text{if target } i \text{ is not tracked at time } t \\ \\ 1 & \text{if target } i \text{ is tracked at time } t \end{cases} \tag{9}$$

However, for the overall system to be effective, it must convey target tracking information back to the base in real time. Therefore, the TTTCP metric is modified to take into account the target information available at the base. This new metric is called Base Target Track Count Percentage (BTTCP), shown in Equation 10.

$$\text{BTTCP} = \frac{1}{t_{end}} \frac{1}{n_{target}} \sum_{t=1}^{t_{end}} \sum_{i=1}^{n_{target}} \beta_i\left(t\right) \tag{10}$$

Where,

$$\beta_i\left(t\right) = \begin{cases} 0 & \text{if base has knowledge of target } i \text{ at time } t \\ \\ 1 & \text{if base has knowledge of target } i \text{ at time } t \end{cases} \tag{11}$$

The BTTCP and TTTCP are measures of the system effectiveness. These metrics can be combined to evaluate another measure of system performance: efficiency. Since TTTCP sets the maximum amount of information about the targets that was acquired during the simulation, the ratio of BTTCP to TTTCP is a measure of the amount of information that the base received compared to the amount of information known to the agents. The network efficiency, $\eta_{\text{network}}$, is shown in Equation 12. These three system level metrics are measured for each run of the simulation.

$$\eta_{\text{network}} = \frac{\text{BTTCP}}{\text{TTTCP}} \tag{12}$$

### *6.4 Implementing the Simulation*

Distributed-MASS was implemented using MathWorks MATLAB. MATLAB was selected due to its general purpose computing capabilities (including data storage, processing, and visualization) and its wide-spread use within the research community. Another major benefit of using MATLAB is the ability to use built-in parallel processing to take advantage of the computational resources available. Although specialized programs exist for building agent-based models, as will be demonstrated in this section, it is relatively straight forward to implement an agent-based model in a general purpose computing language.

As was discussed earlier, an advantage of agent-based modeling is the straight-forward correspondence between the system itself and the model used to simulate it. The main procedure of Distributed-MASS is shown below using pseudo-code. The simulation updates the states of the targets and agents at each time step until the simulation terminates. Since the states of all of the agents cannot be updated instantaneously and must be calculated serially, an intermediate subroutine is used to update the network connectivity and propagate shared information so that all agents are using consistent information when determining their next course of action.

**Algorithm 1** Main Procedure

```
procedure RunSimulation(Parameters)
    $SimVariables \leftarrow Parameters$                    ▷ Initialize simulation variables
    for $t \leftarrow 1, t_{max}$ do                        ▷ Iterate through time steps
        for $iTarget \leftarrow 1, numTargets$ do  ▷ Take step for each target and update state
            $Target(iTarget)$ = TargetStep($Target(iTarget)$)
        end for
        PropagateNetwork()                 ▷ Updates network and shares targeting data
        for $iAgent \leftarrow 1, numAgent$ do      ▷ Take step for each agent and update state
            $Agent(iAgent)$ = AgentStep($Agent(iAgent)$)
        end for
        CalculateNetworkProperties()          ▷ Network properties for post-processing
    end for
    ProcessResults()                                                ▷ Process results
end procedure
```

The logic for updating the target is straight forward (i.e., a random walk) and will not need detailed explanation. The logic for the agents, excluding the logic for refueling and dispatching, is shown below in Figure 18. This decision tree matches the description of agent behavior described in Section 6.3.4; however, this way of representing the logic should make it clear how the operating modes are related and are continuously changing based on the conditions encountered during the course of the simulation. The decision tree for the refueling and dispatch portions are shown in Figure 19. This portion was separated from the main portion of the logic tree in order to simplify the representation. Unlike the main logic tree where the agents transition between modes as shown in Figure 18, agents can immediately transition to the refuel and dispatch logic tree once the critical fuel level is reached. The critical fuel for each agent is calculated at every time step, so the possibility of transitioning from the main logic to the refuel logic can occur at any time. At the end of the refuel and dispatch logic tree, the agent will resume the main logic tree starting at search mode (mode 3).

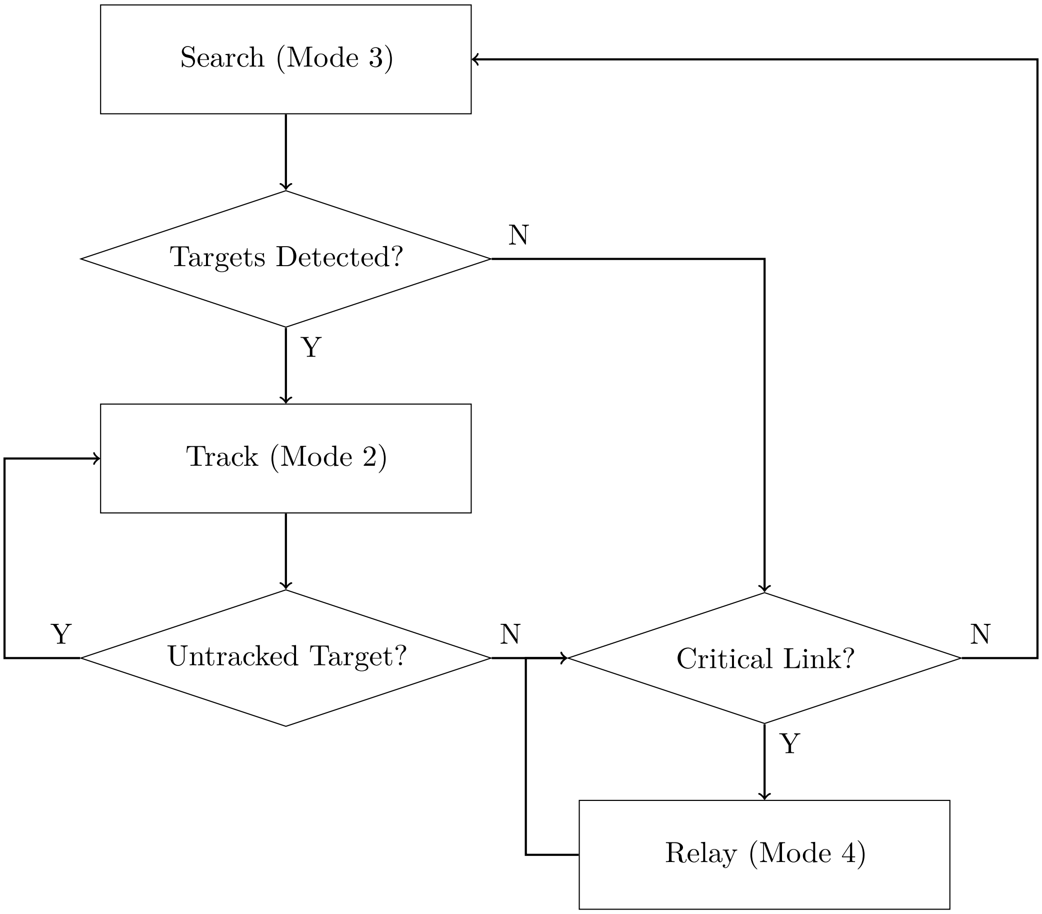

**Figure 18:** Decision tree for agent operating modes

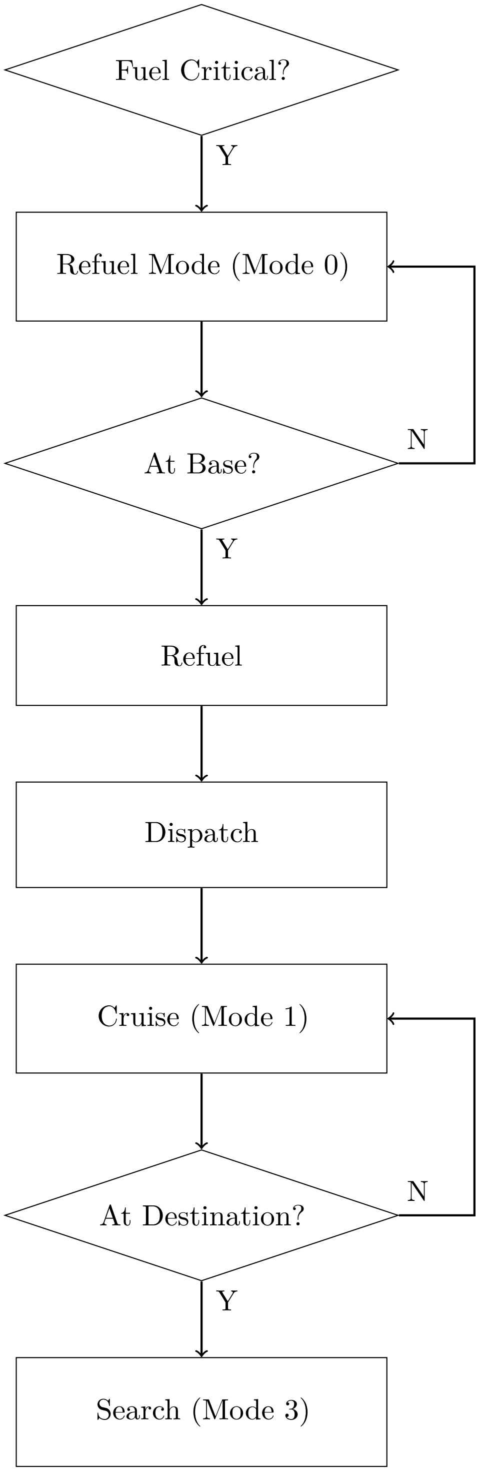

**Figure 19:** Decision tree for agent refuel and dispatch

The pseudo-code for the agent logic is shown below. In order to simplify the pseudo-code and make it easier to parse, the logic shown in Figures 18 and 19 is not shown in this pseudo-code. Instead, the decision trees should be substituted at the indicated locations within the pseudo-code.

**Algorithm 2** Agent Logic

```
procedure AgentStep
    if $AgentPosition \neq Destination$ then
        $NextPoint \leftarrow$FindNextPoint($AgentPosition, Destination$)
    else
        Start Logic Tree
    end if
    $FuelRequired \leftarrow$CalcFuelRequired($AgentPosition, NextPoint, Mode$)
    $FuelAvailable \leftarrow FuelAvailable - FuelRequired$
    $AgentPosition \leftarrow NextPoint$                    ▷ Agent takes step to new point
    Evaluate Logic Tree
    if $Mode = 2$ then
        $Destination \leftarrow TargetPosition$                        ▷ Pursue target
    else if $Mode = 3$ then
        $Destination \leftarrow$PerformRandomWalk($AgentPosition$)
    else if $Mode = 4$ then
        $Destination \leftarrow AgentPosition$                         ▷ Hold position
    end if
    $FuelRequired \leftarrow$CalcFuelRequired($AgentPosition, Base$)
    if $FuelRequired \geq FuelAvailable * FuelSafetyFactor$ then
        $Destination \leftarrow Base$
        $Mode \leftarrow 0$
    end if
end procedure
```

The pseudo-code for the PropagateNetwork is shown below. This algorithm computes the network connectedness and shares targeting information among connected neighbors.

## *6.5 Characterizing the Simulation Results*

The next step is to execute the simulation in order to understand the system behavior. Initially, the system behavior at a single design point will be examined. After that, a design space exploration will be performed in order to investigate how the system behavior changes throughout the design space.

**Algorithm 3** Propagate Network and Information Sharing Procedure

```
procedure PropagateNetwork
    for $iAgent \leftarrow 1, numAgents$ do
        $Agent(iAgent) \leftarrow$ FindTargetsInRange($Agent(iAgent)$)    ▷ Find all in range targets
    end for
    for $iTarget \leftarrow 1, numTargets$ do
        $Target(iTarget) \leftarrow$ UpdateTrackedTargets($Target(iTarget)$)    ▷ Update targets that are actively tracked
    end for
    $NetworkAdjacency \leftarrow$ ComputeAdjacency($AgentPositions$)    ▷ Calculate adjacency matrix
    for $iAgent \leftarrow 1, numAgents$ do
        $ConnectedNodes \leftarrow$ GetConnectedNodes($NetworkAdjacency, iAgent$)    ▷ Get all connected agents
        if $ConnectedToBase$ then
            for all $NeighborPair \in AllNeighbors$ do
                $ShortestPath \leftarrow$ FindShortestPath($NeighborPair$)    ▷ Find shortest path between neighbors
                if $iAgent \in ShortestPath$ then
                    $CriticalLink \leftarrow True$
                end if
            end for
        end if
        for all $Agent \in ConnectedNodes$ do
            $KnownTargets \leftarrow \cup(KnownTargets$, GetKnownTargets($Agent$))    ▷ Share targeting information with connected agents
        end for
    end for
    $ConnectedNodesBase \leftarrow$ GetConnectedNodes($NetworkAdjacency, Base$)    ▷ Get all agents connected to Base
    for all $Agent \in ConnectedNodesBase$ do
        $BaseKnownTargets \leftarrow \cup(BaseKnownTargets$, GetKnownTargets($Agent$))    ▷ Get all targets known to Base
    end for
end procedure
```

### 6.5.1 Simulation Design Variables

The design variables for the individual component classes used in the simulation were described above in Section 6.3. However, there are additional design variables and simulation parameters that define the operation of the system. These design variables and simulation parameters are shown in Table 4. While the number of agents and targets are obvious system level parameters, the number of time steps is also an important simulation parameter. Since the agents have limited endurance, the number of time steps for the simulation is important to evaluate how the periodic refueling affects the results. The number of time steps should be chosen to reflect the appropriate time scale for the system.

**Table 4:** Simulation design variables

| Design Variable | Value |
|---|---|
| Number of Agents | 10 |
| Number of Targets | 3 |
| Number of Time Steps | 2500 |

### 6.5.2 Characterizing a Single Design Point

The goal of this section is to characterize the system-level behavior shown in the simulation. The Distributed-MASS simulation is an example of a self-organizing complex system, so it is expected to exhibit several key characteristics of complex systems, especially nonlinearity and stochasticity.

Since the problem formulation is inherently probabilistic, the behavior of the system can only be seen through many replications of the simulation. The simulation was executed using the design variable values specified in Tables 3 and 4. The simulation was replicated 250 times using the same set of design parameter values. The time history of the cumulative BTTCP metric is shown for each replication in Figure 20. This plot is indicative of the range of behavior the system can exhibit at this design point.

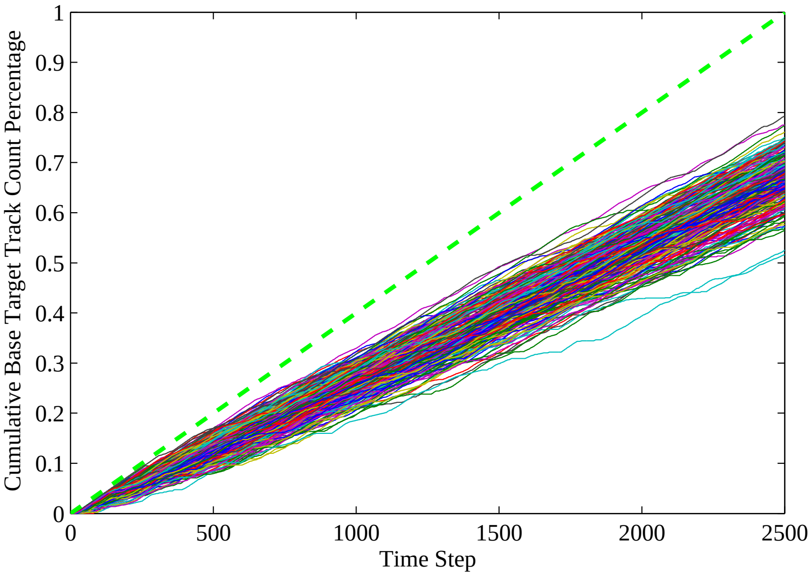

**Figure 20:** Time history of cumulative BTTCP metric for 250 replications

Tracking the time history of the system metrics is cumbersome, so only the value at the end of the simulation will be used. The distribution of the BTTCP values for the 250 replications is shown in Figure 21. This distribution corresponds with the distribution of values at the final time step from Figure 20. This figure shows that the system effectiveness ranged from about 50% to 80% with a mean value of 67%.

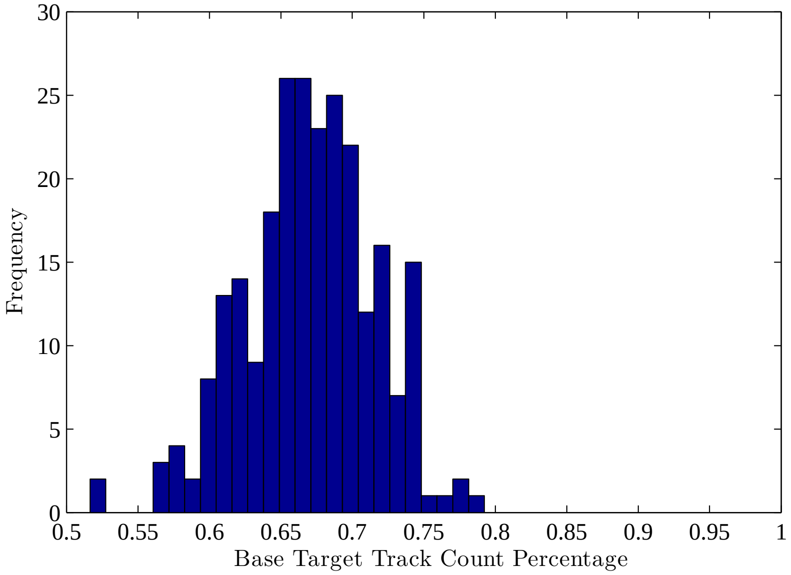

**Figure 21:** Base Target Track Count Percentage (BTTCP) distribution

Similarly, the distribution of the other system metrics can be examined. The TTTCP distribution is shown in Figure 22. This distribution shows that the targets were reliably tracked by the agents. This metric had a range from about 84% to 94% with a mean of 89%. As discussed earlier, this metric captures the effective rate that the agents were able to track the targets without consideration of the ability to relay that information back to the base. Since this metric does not consider the communication network, this distribution is well-behaved with a fairly narrow range of values.

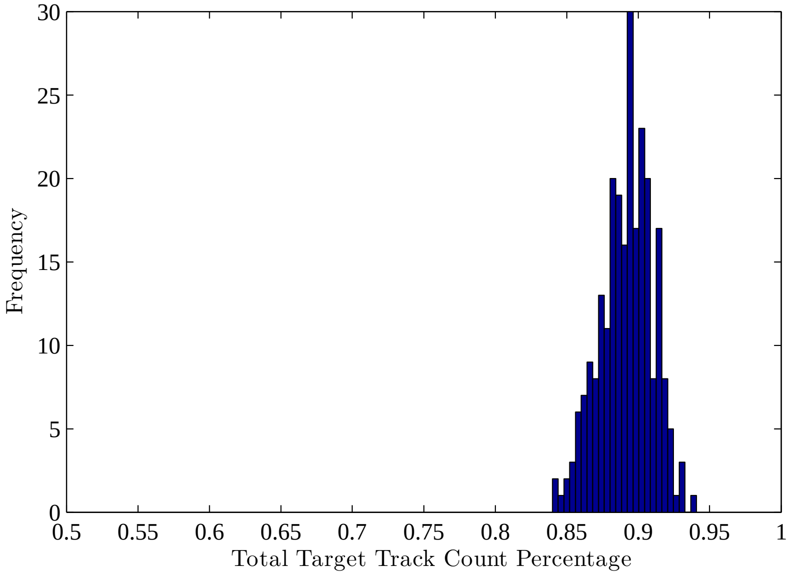

**Figure 22:** Total Target Track Count Percentage (TTTCP) distribution

The distribution of the final system metric, $\eta_{\text{network}}$, is shown in Figure 23. This distribution shows that the network efficiency ranged from about 59% to 85% with a mean of 75%.

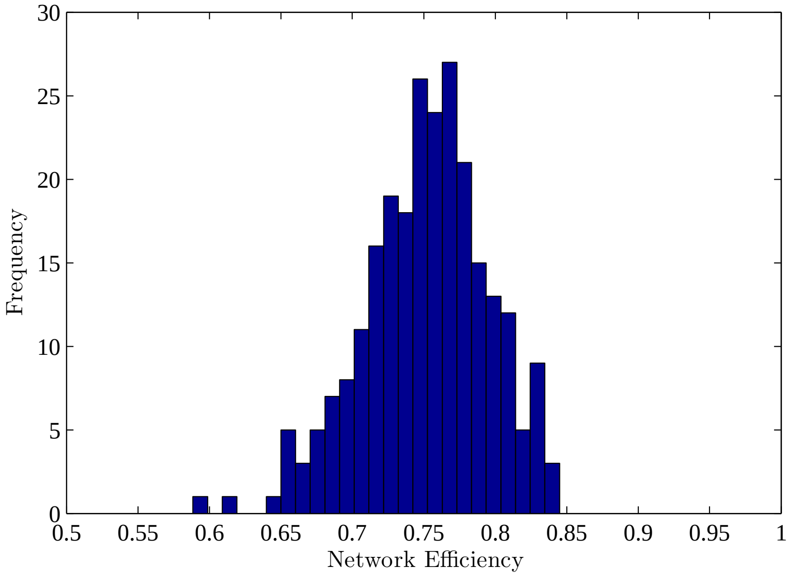

**Figure 23:** Network efficiency distribution

### 6.5.3 Design Space Exploration

While understanding the behavior of a single design point is important, the ultimate goal of the methodology presented in this thesis is to improve the design space exploration of complex systems. The first step toward this goal is to examine the behavior of the system over a small design space. For this initial exploration, the design space will be kept small so that it can be visualized and points of interest can be identified by inspection. The design variables chosen for this exploration are the number of agents and the communication range. The agent detection range is included as an implicit design variable. The detection range is not an independent design variable but rather assumed to be half of the value of the communication range. This is done to manipulate the agent's overall sensor performance without increasing the dimensionality of the problem. These variables are meant to help answer the question of whether the system is more effective with a smaller number of highly capable agents or a larger number of less capable agents. This is a fundamental question that a system designer must answer when looking at distributed autonomous systems.

**Table 5:** Design space exploration variables

| Design Variable | Min Value | Max Value |
|---|---|---|
| Number of Agents | 4 | 22 |
| Communication Range | 10 | 50 |
| Detection Range* | 5 | 25 |

* assumed to be 1/2 of the Communication Range

The design variables and their range of values used in this experiment are shown in Table 5. The number of targets was set to 5 and the simulation was executed for 2500 time steps. The remainder of the simulation parameters were kept at the baseline values shown in Tables 3. A Design of Experiments (DOE) was created for these two design variables at 10 levels per variable using a full-factorial design resulting in 100 design points. One hundred replications were done at each design point resulting in 10,000 executions of the simulation.

Although the primary goal of this subsection is to examine how the system level metrics change for various design points, it is illustrative to examine the aggregated behavior over the entire design space. The system metrics for all 10,000 simulations are shown in Figure 24. The most obvious characteristic of the system behavior is the multimodal behavior. In subsection 6.2.2 of this chapter, the ant foraging problem was introduced as an archetype of complex systems that exhibit emergent behavior and are of interest to scientists and engineers for their self-organizing behavior. It is interesting to observe that both the ant foraging simulation (Figure 25) and the Distributed-MASS simulation show this multimodal behavior with peaks at both the bottom and top end of the system metrics. The peak at the lower end of the metric shows that much of the design space is characterized by disorganized behavior. Intermediate values of the system metric are not particularly interesting and do not dominate the system behavior. Finally, another peak is seen at the high end of the metric. This peak represents highly effective systems that may be indicative of self-organizing behavior.

**Observation**: Complex multi-agent systems are often characterized by multimodal behavior over the design space. At lower end of the system measure, the system is dominated by disorganized or effectively random behavior. The peak on the upper end of the system measure is indicative of possible self-organization. This result also demonstrates that complex systems must often be carefully tuned in order to attain the desired performance.

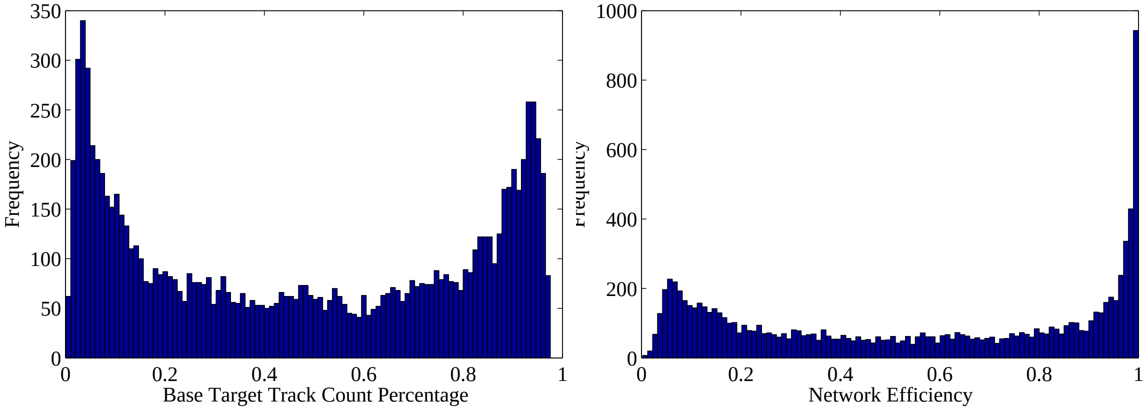

**Figure 24:** System metric distributions over entire design space

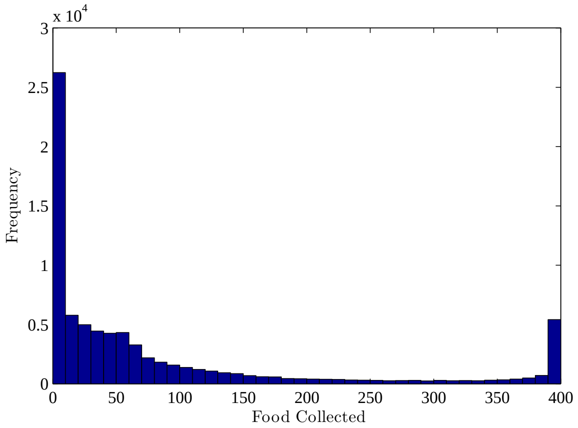

**Figure 25:** Ant foraging system metric distribution over entire design space

The next step is to investigate a range of design points and to examine how the system behavior changes across the design space. One way of characterizing the design space is to plot the variation of the system metrics across the design variable value range. In Figures 26–29, the network efficiency mean and standard deviation are shown for a range of values for the number of agents and the communication range. Due to the strong coupling between the two design variables, the data is shown in the form of box plots. These plots allow the display of both the mean and the spread of the data. However, it should be clarified that the data shown in these plots are the means and standard deviations of the underlying distributions and not the underlying distributions themselves. The whiskers in the box plot are not low-probability extremes but rather the upper and lower bounds for the estimates of the means at this design variable setting. In the context of design space exploration, this means that the desired value of the metric is achievable within the design space evaluated. The best way to interpret this graph is to imagine the ends of the whiskers forming the upper and lower envelope for the expected values for each of the measures evaluated.

Figure 26 shows that there generally is an increase in the system performance for increasing number of agents; however, the spread of the means is too large to be ignored. Due to the coupling between the number of agents and the communication range, every setting of the number of agents exhibits a wide range of performance. As shown in Figure 27, the communication range has a much stronger direct effect on the mean of the network efficiency metric. Compared to the number of agents design, the communication range strongly impacts the range of system performance seen at each level. Figure 28 shows that increasing the number of agents does decrease the mean of the standard deviations, but there is an inherently large range of values seen at each level for the number of agents. Figure 29 shows the variation of the standard deviation with the communication range. It is interesting to note the wide range of possible behaviors at various settings of the communication range value. The mean of the standard deviations has a local peak, however, increasing the communication range beyond this peak value introduces a lot more variability into the expected value of the standard deviations due to the coupling with the number of agents.

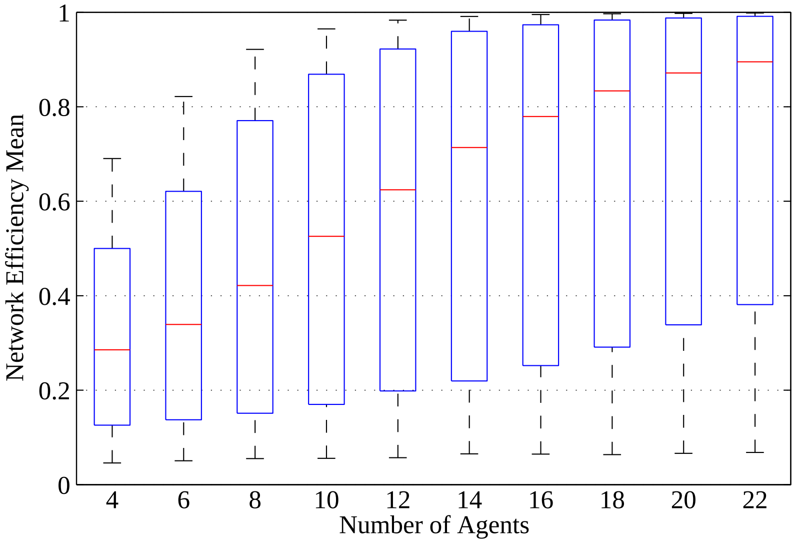

**Figure 26:** Variation of network efficiency mean against number of agents

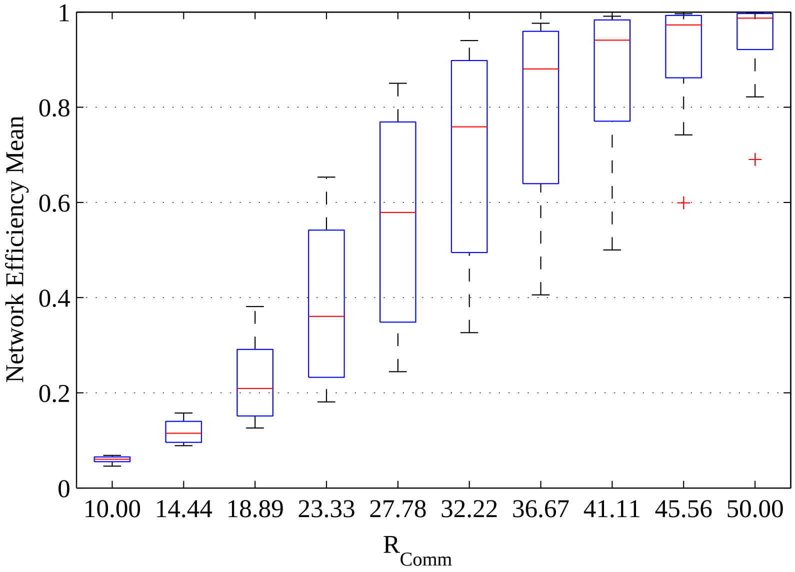

**Figure 27:** Variation of network efficiency mean against communication radius

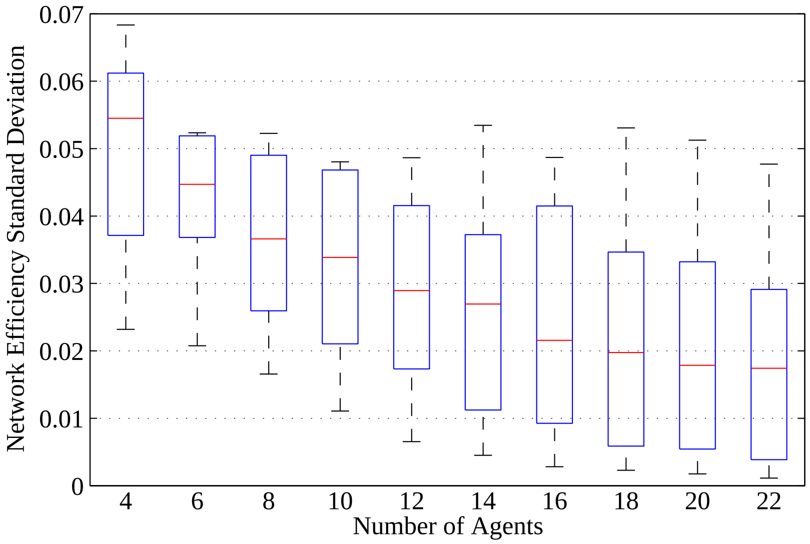

**Figure 28:** Variation of network efficiency standard deviation against number of agents

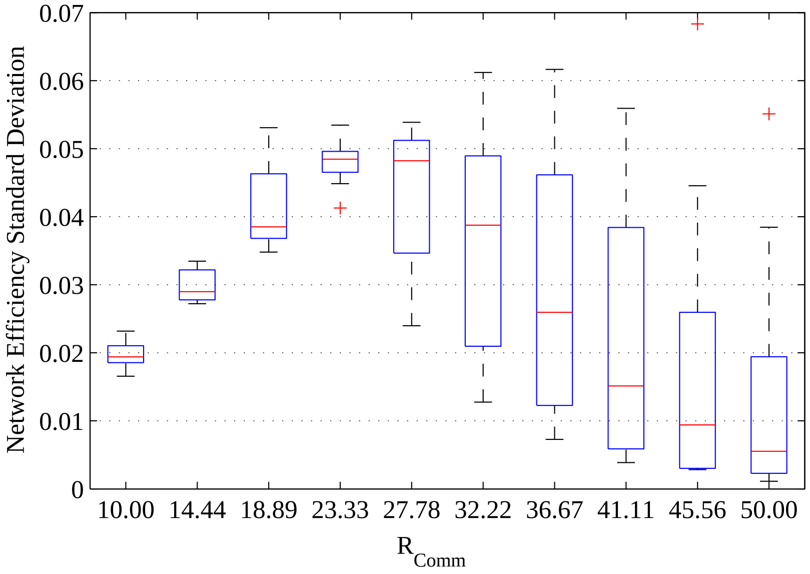

**Figure 29:** Variation of network efficiency standard deviation against communication radius

The previous discussion shows the difficulty when looking at the mean and variance separately or when considering the effect of one variable at a time. In order to attempt to look at the problem more holistically, Figure 30 shows a matrix of histograms which represent the distribution of the BTTCP system metric at various settings of the two design variables examined in this design space exploration. Each histogram depicts the same variable, with the BTTCP value on the horizontal axis and the frequency on the vertical axis. The position within the matrix corresponds with the combination of design variable values. Each row of histograms represents a different value for the agent's communication range and each column represents a different value for the number of agents in the simulation. The magnitude of the mean shift between two adjacent distributions is labeled between the two corresponding distributions. For example, when the communication range was set to 32.2, increasing the number of agents from 4 to 6 resulted in the mean BTTCP increasing by 0.093. Similarly, when the number of agents was kept fixed at 4, increasing the communication range from 32.2 to 36.7 resulted in a mean shift of 0.064. Each histogram has to be small to fit within the matrix, so the details of each distribution are obscured. These mean shift indicators are included to improve the ability to quickly interpret the data presented in the histogram matrix. Although we are interested in both the mean and the shape of the distribution, using a measure based on just the mean is sufficient to make observations about these results.

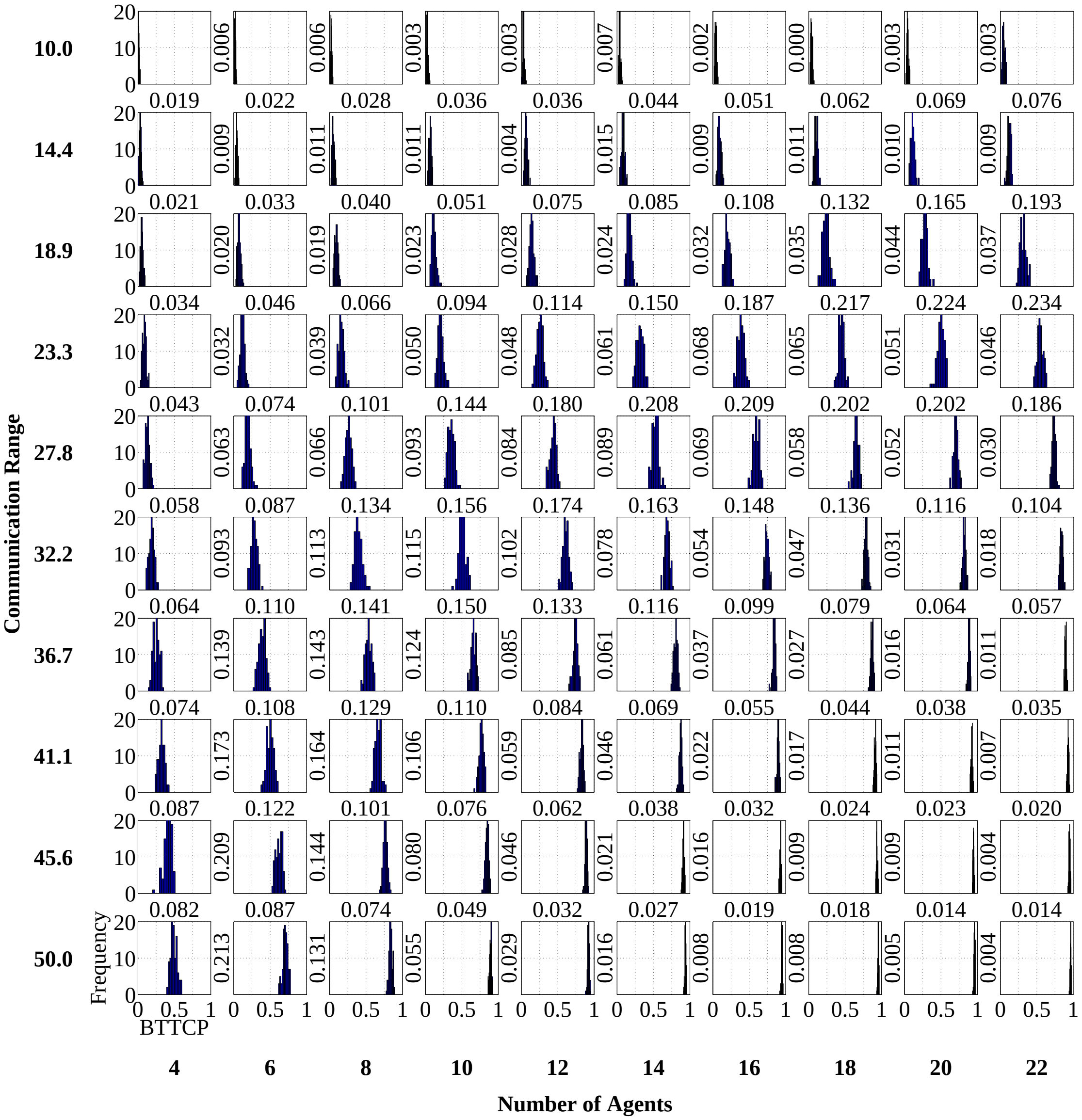

**Figure 30:** Design space histogram matrix with the magnitude of the mean shift indicated between adjacent graphs

The goal for this design space exploration is not to simply identify the values of the design variable that maximize the system metrics—that would be a trivial exercise by simply setting the communication range and number of agents to the maximum values. Arguably, the more interesting regions in the design space are regions that represent shifts in behavior as indicated by either rapid shifts in mean or in the shape of the distribution of the system metrics. Examining the data in Figure 30, there are several candidate points that are indicative of rapid changes of behavior. The largest mean shifts occur between the design points pairs represented by the two following pairs:

$$\textbf{Point A} : \{(R_{COMM} = 50, \text{Num Agents} = 4)\,, (R_{COMM} = 50, \text{Num Agents} = 6)\}$$

$$\textbf{Point B} : \{(R_{COMM} = 18.9, \text{Num Agents} = 22)\,, (R_{COMM} = 23.3, \text{Num Agents} = 22)\}$$

These two points represent the two intuitive approaches to the problem: 1) a small number of agents with a large communication radius, and 2) a large number of agents with a smaller communication radius. In Figure 31, we can examine the lower left corner from Figure 30, which corresponds with the first design region of interest. In this region, there are a small number of agents with a large communication range. An increase from 4 to 6 agents results in a mean shift of 0.213 of the BTTCP system metric. At this communication range design value, the system is very sensitive to the number of agents and the change from 4 to 6 agents can be interpreted as a threshold value where the system undergoes rapid changes in behavior. Although the increase in the mean moving from 6 to 8 agents is not as large as the shift going from 4 to 6 agents, it is also important to note that the distribution at 8 agents has a significantly lower standard deviation compared to 6 agents. For 6 agents, the standard deviation is 0.0369 while the standard deviation at 8 agents is 0.0228, a decrease of over 38%. The change in the shape of the distribution (as measured by the decreased variance) is a feature that would be missed if we were simply to look at metrics that are based on the mean. This decrease in variance is a possible indicator of self-organization.

**Observation**: Rapid shifts of both the mean and the shape of the distribution are important indicators for changes in the system behavior. They are possible indicators of emergent behavior and/or self-organization.

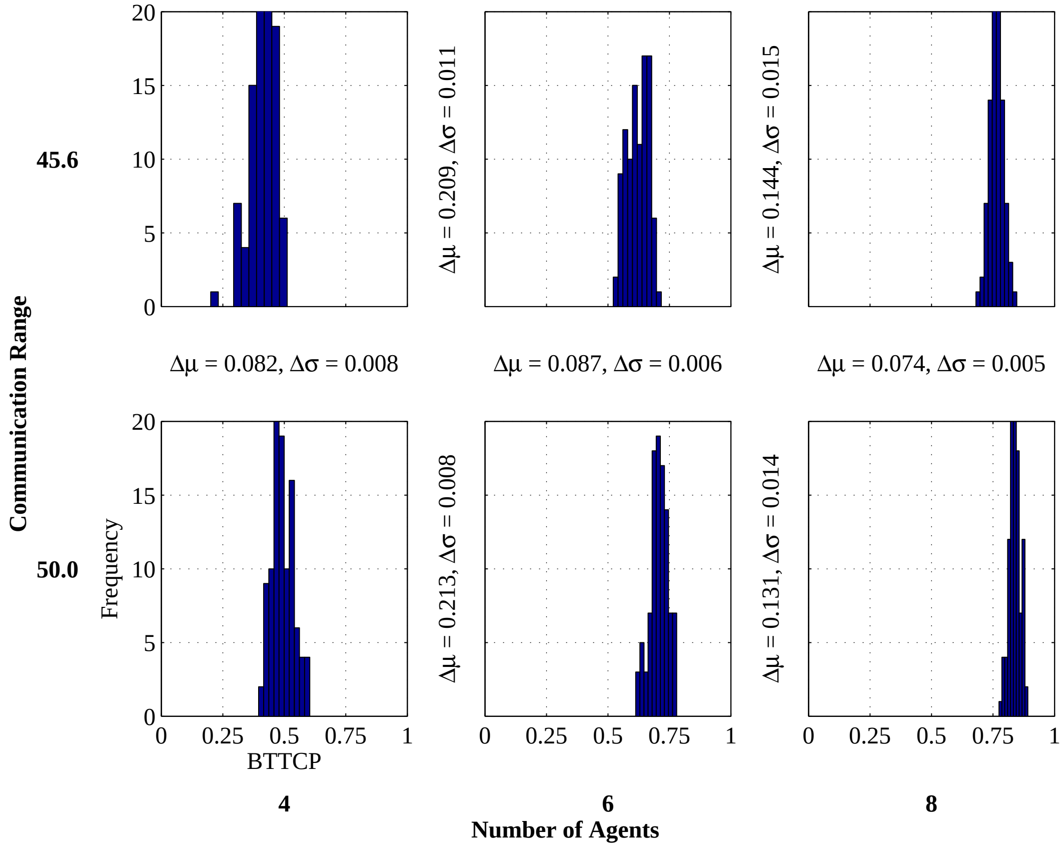

**Figure 31:** Design space histogram matrix with the magnitude of the mean shift and change in standard deviation indicated between adjacent graphs (Detail Point A)

The next region of interest corresponds with the design point that uses a large number of agents with a smaller communication radius. This region is potentially more interesting than the design region examined above, since a large number of agents is an enabler of complex behavior. The results from this simulation show that this is indeed the case: the greatest shifts in the mean and the standard deviation are seen when the largest number of agents are used. A detail of this region of the design space is shown in Figure 32. As introduced above, the design point of interest is the transition from a communication range of 18.9 to 23.3 with 22 agents. The mean shift between these two design points is the largest seen in the design space with a value of 0.234. Continuing to increase the communication range from 23.3 to 27.8 also continues to improve performance with a smaller mean shift of 0.186 but it is important to note that the shift in standard deviation (0.018) is also significant and represents an important change in system behavior.

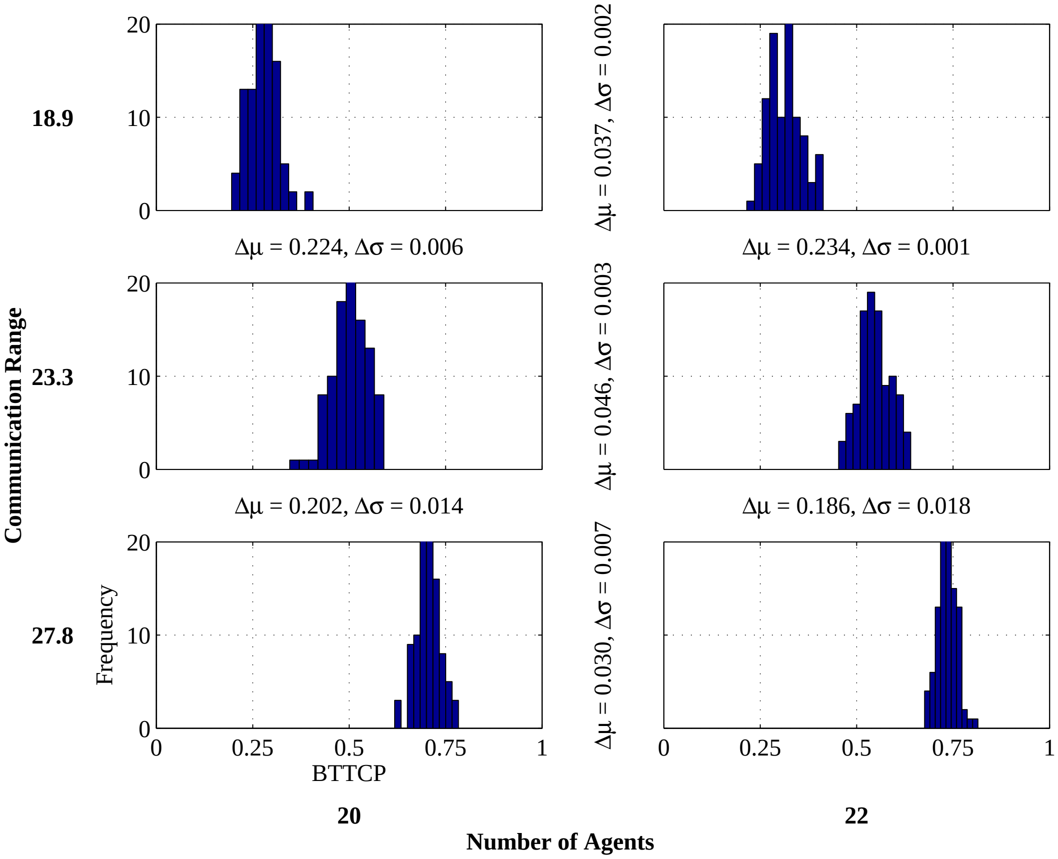

**Figure 32:** Design space histogram matrix with the magnitude of the mean shift and change in standard deviation indicated between adjacent graphs (Detail Point B)

A similar design space histogram matrix for the network efficiency system metric is shown in Figure 33. Overall, the trends are similar to those seen when looking at the BTTCP system metric. However, from a network efficiency standpoint, it becomes clear that the design points corresponding to a large number of agents with a smaller communication radius (Point B) is an even stronger candidate for self-organized behavior. This design point sees a mean shift of 0.272 in the network efficiency metric while the other region corresponding to Point A saw a mean shift of just 0.131. This makes sense since the dramatic increase in performance at Point B was due to improved network effects while the improvement at Point A was primarily due to individual agent performance. Since network effects in a distributed system are of great interest, Point B is the design point that the methodology presented in this thesis should help identify.

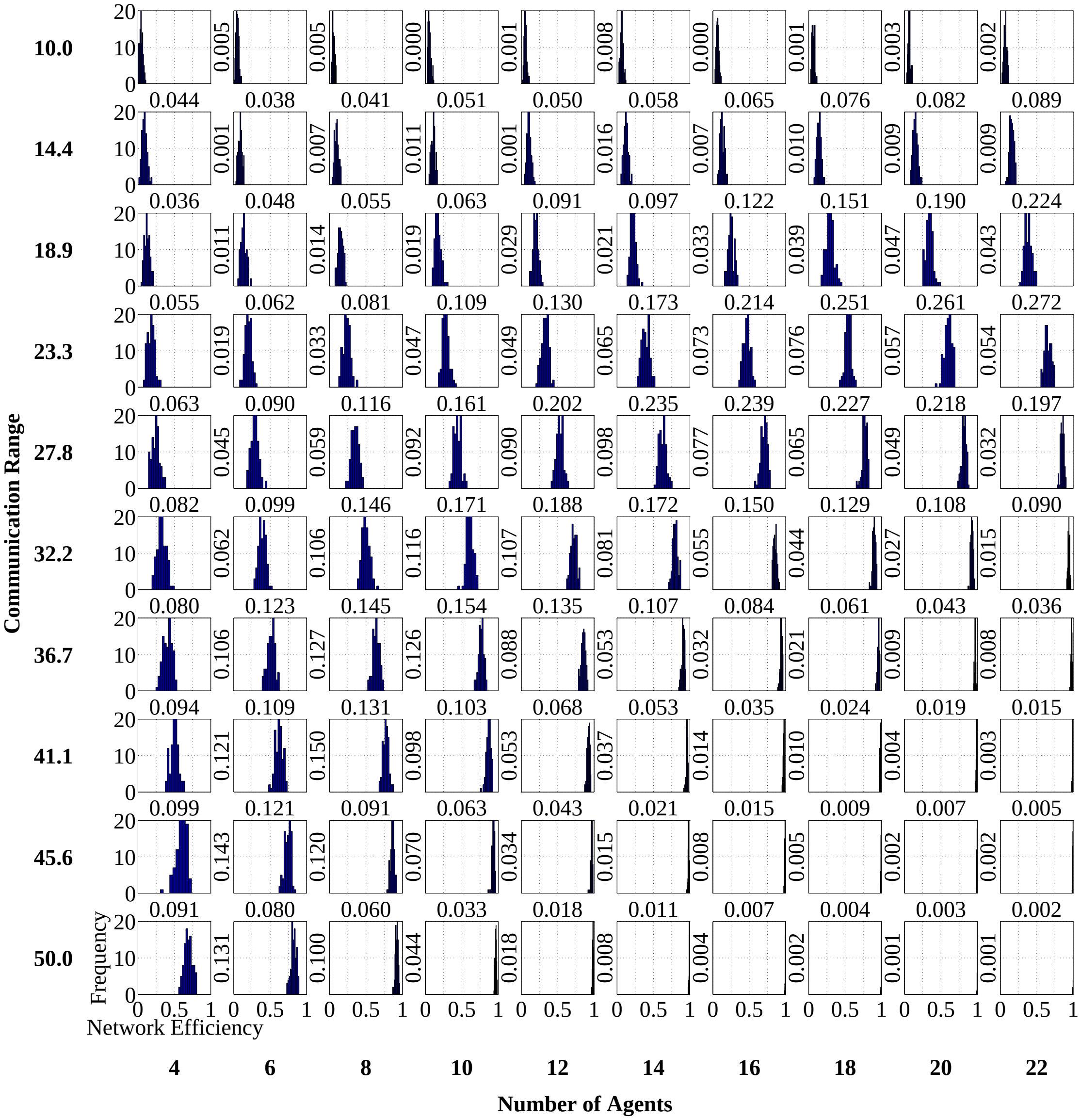

**Figure 33:** Design space histogram matrix for the network efficiency system metric with the magnitude of the mean shift indicated between adjacent graphs

A contour plot of the mean of the network efficiency at each design point evaluated is shown in Figure 34. The mean network efficiency is fairly well behaved with the efficiency improving with increasing number of agents and increasing communication range. The standard deviation of the network efficiency at each design point, as shown in Figure 35, shows that the behavior is much more complicated than if only the mean behavior is considered. The first observation is that the variance in the distribution is not homogeneous throughout the design space, a property known as heteroscedasticity. This property makes regression based analysis difficult, since homoscedasticity is a required property for many statistical estimations. The other observation is the existence of a band of higher variance that separates the two regions identified above. The low variance region at the top of standard deviation figure corresponds to the left peak of network efficiency distribution shown in Figure 24, while the low variance region at the bottom of the figure corresponds to the right peak. In between these two regions is the transition region where potentially interesting behavior occurs.

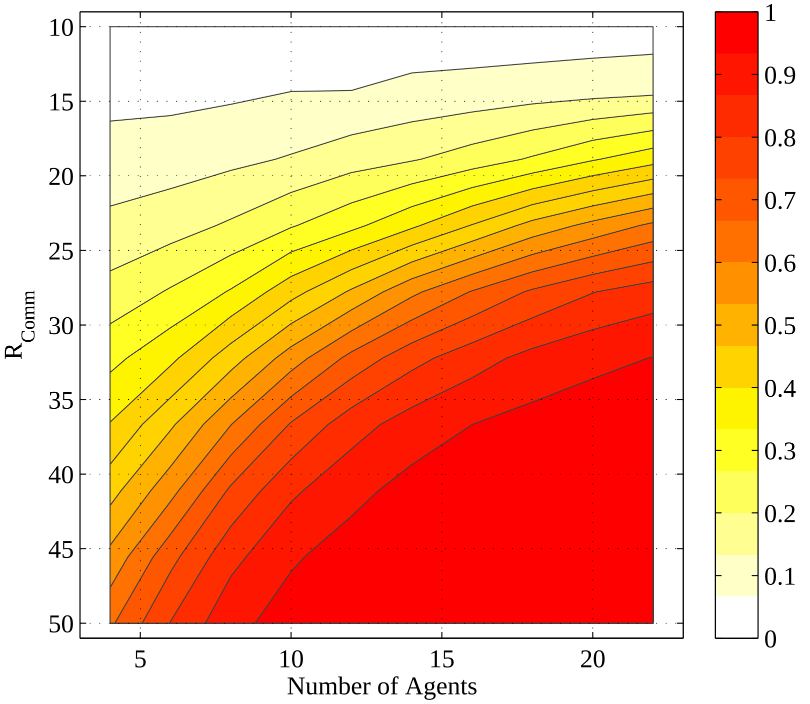

**Figure 34:** Design space contour plot of the network efficiency system metric mean

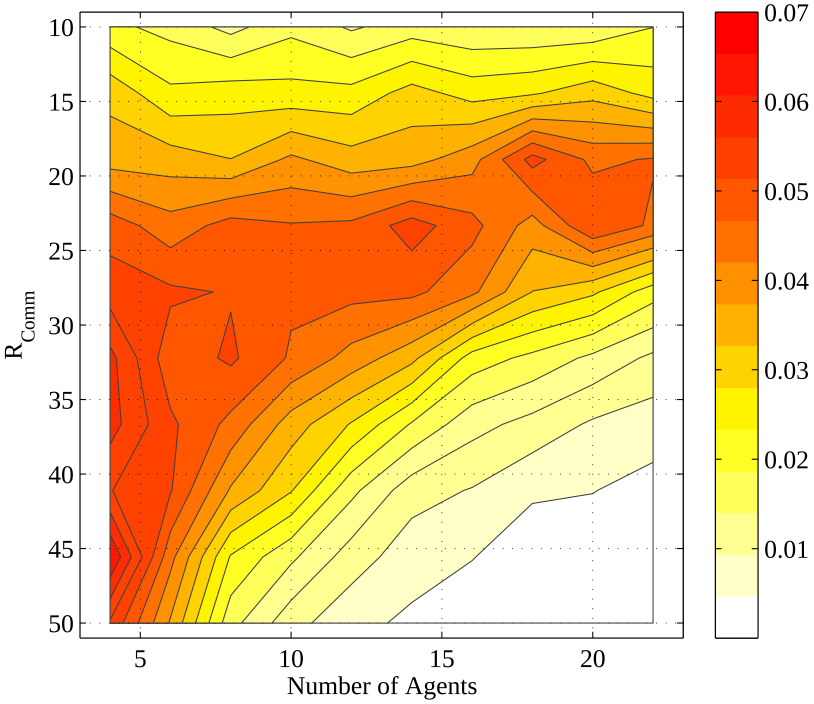

**Figure 35:** Design space contour plot of the network efficiency system metric standard deviation

To visualize how the mean and standard deviation shifts vary across the design space, these values were plotted as a contour plot. The magnitude of the mean shift between adjacent design points is shown in Figure 36; the standard deviation shifts is shown in Figure 37. In these contour plots, the design points are indicated by solid black dots. The change in the mean and the standard deviation for each pair of adjacent design points is calculated and then plotted at the geometric midpoint between the pair of points. The contour plot helps to confirm the design space regions of interest identified earlier in this chapter. The mean shifts are greatest in the upper right portion of the figure. In this region, the plot highlights the strong contrast between the mean shift in each design variable direction surrounding a design point. A small change in the communication range results in a significant shift in the distribution while a small change in the number of agents has relatively small effect. Examining the contour plot for the shifts in the standard deviation of the network efficiency metric shows that, while the highest magnitude was seen in the lower left corner (corresponding to Point A, identified earlier), the region in the upper right corner also saw significant shifts in the standard deviations. It is interesting to note that the shifts in the standard deviation are often complementary to the shifts in the mean. For example, examining the local peak in the contour plot of the mean shift along the $R_{COMM} = 30$ line, the maximum shifts in the mean are seen at Number of Agents $= 12$ with the mean shift magnitude decreasing as the number of agents increases above this point. However, the corresponding region in the standard deviation contour plot shows that the shifts in the standard deviation increase as the number of agents increases. This shows that beyond a certain point, adding more agents to the system reduces the variance of the system performance more than improves the magnitude of the network efficiency metric.

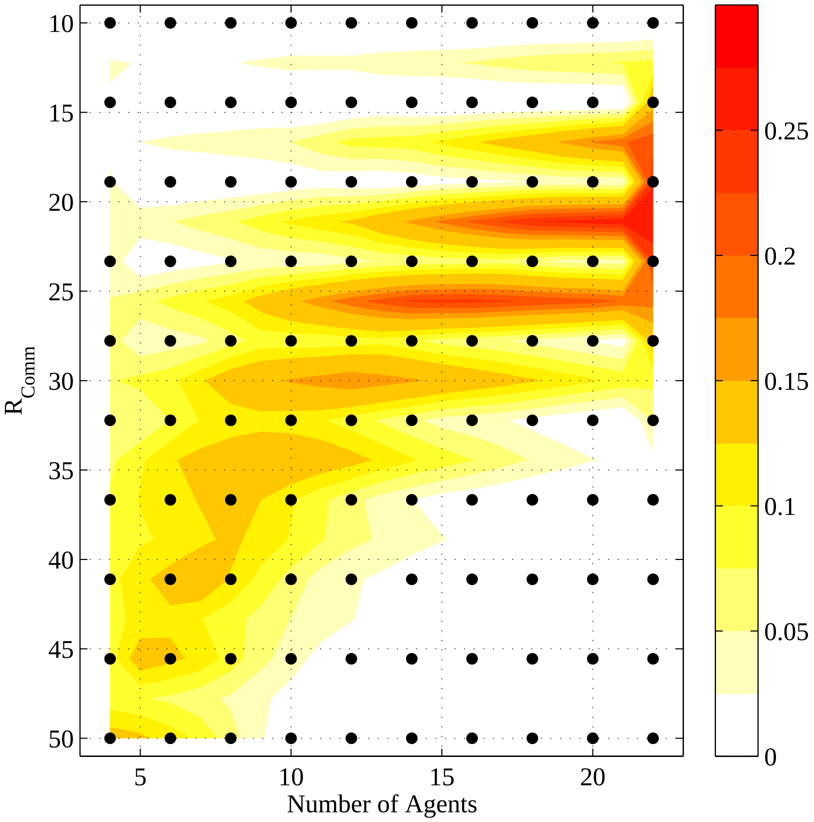

**Figure 36:** Contour plot of the magnitude of the mean shift of the network efficiency metric between adjacent design points

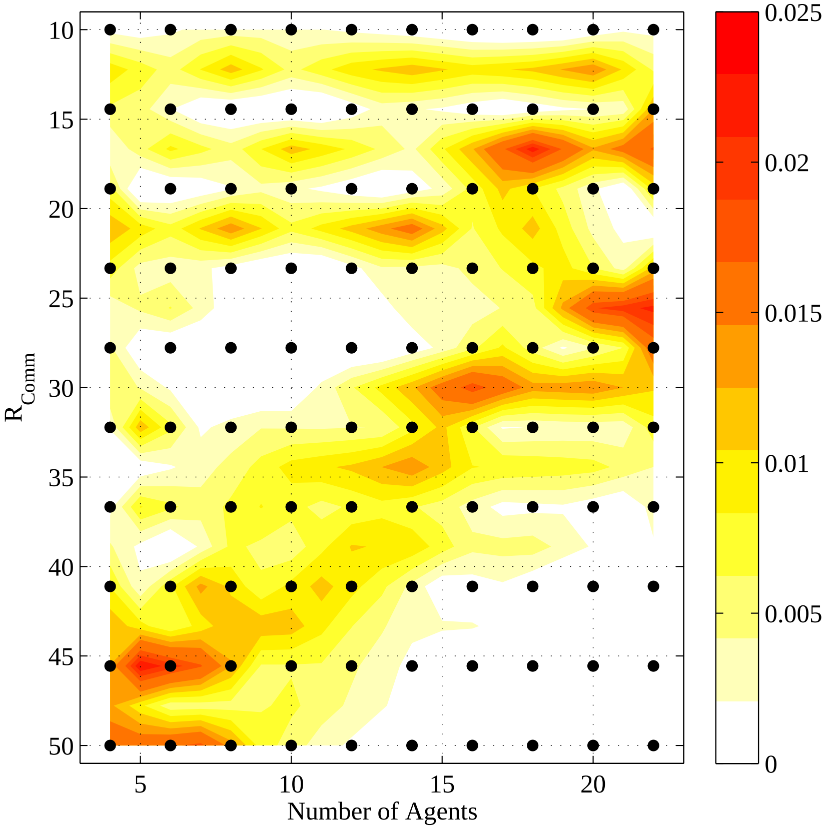

**Figure 37:** Contour plot of the magnitude of the standard deviation shift of the network efficiency metric between adjacent design points

### *6.6 The Design Space Exploration Problem*

Although the problems with performing design space exploration have already been discussed earlier in this thesis, some of these issues will be revisited using the context of the Distributed-MASS simulation. The first approach considered to identify candidate design points displaying emergent behavior is the inverse design method. In this approach, the desired outcomes are selected from a thorough sampling of the design space and then the design points that correspond with these outcomes are selected as candidate designs. The most fundamental flaw with the inverse design method for a complex system can be illustrated by returning to Figure 20. This figure illustrates the many-to-one relationship between the outputs and the inputs for a complex and probabilistic system. Selecting a subset of the outputs is not likely to uniquely determine a candidate design. Furthermore, there is no consideration for the likelihood of the candidate point producing the desired behavior. Cherry-picking the very best observations from a particular design point is fundamentally unsound and would lead to incorrect design decisions. The inverse design method can be modified by either requiring that *all* outcomes associated with a single design point are simultaneously selected or that desired outcomes are weighted by their likelihood of outcome. The former approach will be unwieldy while the latter approach would also be difficult to use and interpret.

Despite these limitations, an inverse design approach will be illustrated below to highlight some of these issues. All of the results from the design space exploration are filtered to select only those runs which had a network efficiency of 0.8 or greater. This value was chosen since it represents designs which are generally well performing. This value also captures Point B from the design space exploration performed in section 6.5.3. The filtered design points along with histograms showing the frequency distribution of the design variable inputs are shown in Figure 38. As was seen in the earlier design space exploration results, the best performing systems were those that simultaneously had the most number of agents and the largest communication range. The histograms in Figure 38 clearly show how these design points dominate the cases where the system had a network efficiency greater than 0.8. Despite the dominance of this overall trend, the histogram shows that there were a small number of points that were able to also meet the network efficiency at a relatively low number of agents or communication range. A designer would hopefully investigate these points as they represent an opportunity to meet the objective using fewer resources and at a lower overall system cost. However, this approach relies heavily on the designer being able to visualize the data, identify both the trends and the outliers, and interpret the data to identify candidate solutions. This becomes increasingly difficult as the dimensionality and stochasticity of the problem increase. It is also highly sensitive to the thresholds set—a higher threshold value for the network efficiency would eliminate Point B from the filtered data set and eliminate it from further investigation. This threshold approach also neglects emergent vulnerabilities. Focusing on the top performing systems blinds the designer to systems which may be severely degraded if system or environmental variables change to unexpected values. Furthermore, while setting a high threshold value will show the best performing design variable values, it will actually provide little understanding of the underlying system. It is obvious that simultaneously maximizing the number of agents and communication range would maximize the performance of the system. However, to understand the system, the design process should illuminate more than just the obvious solutions. These problems make inverse design for a complex system an ad hoc approach that is highly dependent on the designer. The goal of this thesis is to introduce a methodology that makes the design process repeatable and reliable in the face of increased problem dimensionality and stochasticity.

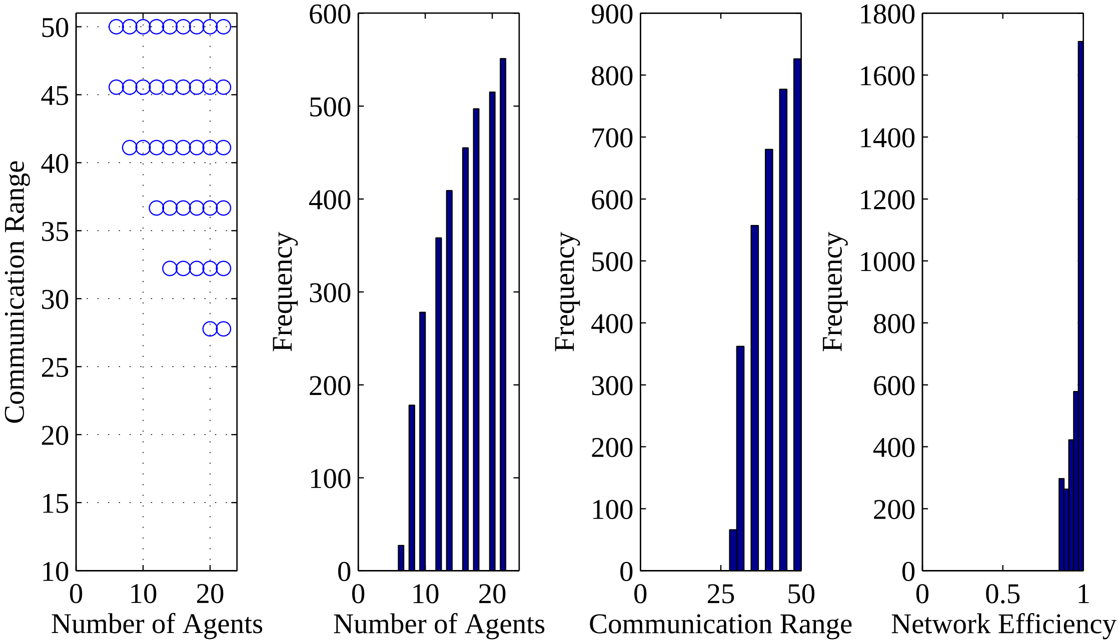

**Figure 38:** Design points filtered for Network Efficiency > 0.8

The design space exploration performed in section 6.5.3 leads to the identification of several regions of the design space that are candidates for further study. These points represented a transition region where the system performed significantly better for small changes in the design variables. Although these points were not strictly the best performing in the design space, they provide the most *information* about the system. Although the concept of information will be discussed in length in the chapter on information theory, the salient point is that information is the change in uncertainty regarding a set of outcomes.

If the outcomes (i.e., the distribution of the system metric) do not change much from one design point to the next, then there is little information being conveyed. On the other hand, if the outcomes change rapidly from one design point to the next, these points provide the most information about the underlying system. Identifying the design points that correspond to shifts in behavior directly leads to an overall characterization of the design space. These critical points are the boundaries separating regions of the design space where the behavior (as measured by the distribution of the system metrics) is similar. In this way, a topology of system behavior is generated across the design space.

**Claim**: When performing a design space exploration of a complex system, one of the primary goals of the process is to identify the critical regions that provide the most *information* about the design space and the underlying behavior of the system.

# CHAPTER VII

# METHODOLOGY

**Chapter Road Map**: The goal of this chapter is to synthesize a methodology that will enable the systematic exploration of a design space to identify design points potentially exhibiting emergent behavior. This chapter addresses research question **RQ3** and related sub-questions:

- **(RQ3)** What is the appropriate methodology for engineering complex systems?
  - **(RQ3.1**) What are the required features of a design methodology for emergence?
  - (**RQ3.2**) How can the design space of a complex system be systematically explored in order to identify emergent behavior?
  - (**RQ3.3**) What is the appropriate way to perform design space exploration on a complex, stochastic space?

## *7.1 Methodology Foundations*

The purpose of this research is to develop a systematic method of interrogating computer models of complex systems in order to identify conditions that lead to emergent behavior. This research proposes a novel use of modeling and simulation tools during analysis, design, and evaluation of complex systems. More specifically, this work will propose a method for systematically identifying regions of the design space which are possible indicators of emergent behavior in the system under consideration. It is assumed that these emergent behaviors are only apparent in subspaces of the entire design space being explored (i.e., a critical transition can be found). If the emergent behavior exists over the entire design space, the proposed method is not necessary to identify these behaviors.

Before diving into the formulation of the methodology, this chapter will discuss several of the cornerstones of the approach. These form both the philosophical and practical foundations of the methodology. These cornerstones are modeling and simulation, statistical methods, and information theory . Modeling and simulation is a cornerstone of modern engineering methods. Traditionally, this approach has been used for *prediction*: finding the output or result of a model given a set of input conditions. However, modeling and simulation can also be used as a tool for discovery of new relationships and principles—a use which is argued to be of equal or greater importance than just prediction [14]. As simulation tools grow in complexity and computing power increases, there becomes an overwhelming amount of data to observe. This leads to the use of *statistical methods* to facilitate the knowledge creation process by highlighting interesting or unexpected results in the data. Even more fundamental is the inductive nature of design. Tribus argues that "every problem in engineering design is a problem of inductive logic" [241]. Inference through statistical methods is the way we can gather knowledge in the design process. Furthermore, complex systems are inherently stochastic and they must be treated as *ensembles*. Finally, information theory has deep connections within the study of complex systems. It is useful for capturing both the structure and uncertainty that are characteristics of complexity. Information theory is also very practical and offers methods are widely applicable.

### 7.1.1 Modeling and Simulation

When a system's complexity makes obtaining analytical relations impractical or impossible, *simulation* takes the place of analytical relations. As Maier and Rechtin explain, "a simulation of a system is an analytical model of the system's behavior and performance in terms of the simulation parameters. The connection is just more complex and difficult to explicitly identify" [177]. Clayton and Davies explain that "the use of computer simulations as an experimental tool to model complex systems has encouraged the view that many features of the world cannot be foreseen from contemplating a set of underlying dynamical equations. Rather, they are discovered only from a systematic study of the *solutions* in the form of numerical simulations" [74, p. xi].

Axelrod [14] characterizes seven categorical purposes for the use of simulation: 1) prediction, 2) performance, 3) training, 4) entertainment, 5) education, 6) proof, and 7) discovery. He notes that science and engineering primarily rely on prediction, proof, and discovery. Furthermore, the use of simulation for prediction has been the most common purpose. He argues that the "use of simulation for the discovery of new relationships and principles is at least as important as proof or prediction" [14]. In simulations of complex systems, prediction is not a useful concept. In these cases, the discovery of relationships and behaviors is the more enlightening purpose.

Axelrod sees simulation as a "third way" of doing science compared to the methods of deduction and induction [14]. Like deduction, simulation relies on the execution of a set of logical statements and assumptions. In that way, we know that the results of a simulation are deductively true based on the logic of the program. However, we can also inductively study the outcome of a simulation in the same way we study natural phenomena. This becomes necessary when we deal in systems that exhibit emergent properties and deducing the consequences is often impossible [14]. As O. Holland argues, "the science of modeling and simulation is not only suitable as a platform for the parametric exploration of such systems, but may very well prove the discipline by which such systems must be implemented" [145].

Bedau's [34, 33] and Darley's [86] definition of emergence are based on the idea that emergence is a phenomena that can only be realized through simulation. Although the word simulation is used to mean the more general process of iterating a given model, it certainly also applies to the more common meaning of a simulation on a digital computer using a programmed model.

### 7.1.2 Exploratory Investigations

As has been argued throughout this dissertation, the goal of this research is to enable exploratory investigations of complex systems. The result of this methodology should enable a designer to gain insight into the behavior of the system. More so, it should help identify lever-points in the design space that provide effective means of achieving a desired outcome. In a complex system, emergent behavior is likely to associated with these lever-points that correspond to critical transitions in system behavior.

Lever-points and critical transitions evoke the concept of sensitivity analysis in helping to identify which parameters are most useful for achieving a change in behavior. In fact, it is possible to understand the methodology presented in this dissertation as a type of sensitivity analysis. Similarly, it is common to understand exploratory analysis as a series of "what-if" scenarios. Again, it may be useful to view this methodology as the identification of interesting "what-if" design points that correspond to critical transitions in behavior and as possible candidates of emergent behavior. Evans et al. [110] argue that the distinction between sensitivity analysis and "what-if" analysis is a distinction based primarily on intent and less on fundamentals. The methodology presented in this thesis uses a type of sensitivity analysis to identify the most important "what-if" scenarios.

### 7.1.3 Statistical Formulation

#### *7.1.3.1 Ensemble Approach*

Complex systems are inherently stochastic. Bar-Yam argues that "physical systems are only meaningful as ensembles rather than individual states. Emergent properties reside in the properties of the ensemble rather than of any individual state" [24]. In statistical physics, this concept is known as the *Gibbs ensemble*. Nicolis and Prigogine, both well-known in the field of complex systems and statistical mechanics, argue that complex systems must be understood using this statistical ensemble approach: "when the motion in phase space becomes very complex, as in the chaotic regime, it is no longer meaningful to argue in terms of individual trajectories" [204, pp. 85]. A Gibbs ensemble consists of a large number of identical systems, all subject to the same constraints [204, pp. 85].

In many complex systems, the behavior system is often path-dependent, which means that the time history of state of the simulation is important. Each replication of a simulation will have its own trajectory. The goal will be to run enough replications of the simulation in order to establish a distributional equivalence [14]. That is, the results of a simulation run are statistically equivalent to the ensemble. Ball et al. also recognized the need for a formulation based on ensembles when studying emergence [21]. They defined emergence based on difficulty of forecasting a realization of a system with respect to the ensemble of all possible outcomes. If the outcome “can be anticipated from observing other realizations,” then the observation is weakly emergent; on the other hand, if the observation “can only be forecast from the observation of the past of each particular instance” then the observation is strongly emergent [21].

As an example, ensembles are both a needed and useful approach within the field of climate science. The weather system is a classic example of a complex system that exhibits chaotic behavior. This means that any single prediction is highly improbable; therefore, the collection of all possible outcomes must be analyzed. Clearly, we must change our viewpoint from studying the results of individual trajectories to studying ensembles. The goal of this thesis is to help understand the question, “how does the ensemble of all possible outcomes of the system change through the design space?” The key to enable this type of analysis is the use of statistical methods.

#### *7.1.3.2 Statistical Methods*

Statistical methods in data analysis can be grouped into the following classes [134]:

- *Exploratory Data Analysis*: (e.g., scatter plot matrices, and other visual analytic techniques)
- *Predictive Modeling*: (e.g., regression, classification, and model building)
- *Descriptive Modeling*: (e.g., density estimation, cluster analysis, dependency modeling)
- *Pattern Detection*: (e.g., many data mining methods)

In the language of machine learning, predictive modeling is a supervised learning technique. In supervised learning, we attempt to define the relationship between input variables and output variables. More specifically, the output variables are *labeled*, either by a quantitative value or a categorical value. This provides the learning technique with the goal value of the output for a given input. On the other hand, descriptive modeling is an unsupervised learning technique. In unsupervised learning, there is no goal output value associated with an input. Therefore, the goal of descriptive modeling is to only characterize the input variables.

Exploratory Data Analysis is an interactive, visual, user-guided method for discovering trends, patterns, and outliers in data sets. However, this approach is difficult for high dimensional data sets and relies on the intuition and expertise of the individual performing the analysis. Scatter plot matrices are projections of high dimensional data sets into an array of bivariate plots of all pairwise combinations of dimensions. In each two-dimensional plot, information about all of the other dimensions are lost. The result is that only pairwise relationship can be examined, but higher dimensional relationships are concealed. The methodology developed in this research will use descriptive modeling techniques to enable and improve exploratory data analysis.

Descriptive modeling is a statistical approach to describing a set of input variables that do not have a corresponding output that we are trying to predict. Within the context of the methodology presented in this thesis, a density estimation technique is required. Density estimation, simply stated, is the determination of a probability distribution from discrete data points.

### 7.1.4 Tackling Dimensionality

Large dimensional problems that have many explanatory variables are very difficult to deal with. The *curse of dimensionality* means that it becomes exponentially more difficult to accurately sample the space with increasing dimensions. This requires very large data sets, otherwise, there is a risk of instability of the estimates of the data [134].

The *curse of dimensionality*, a phrase coined by Richard Bellman in 1961, describes the problem of the exponential growth in the volume of a space as the number of dimensions increase. This means that sampling a high dimensional space become more and more sparse with increasing dimensions. Additionally, with increasing dimensions, most data points are closer to a boundary of the domain than to any other data point [136, p. 23]. This makes it difficult to sample the interior of a design space thoroughly. Stated another way, in high dimensions, distances between any two points become more similar. The sampling density is proportional to $N^{1/p}$, where $p$ is the number of dimensions and $N$ is the sample size [136, p. 23]. The number of samples required to achieve a given accuracy grows exponentially [232, p. 93–94].

More specifically, density estimation becomes difficult with increasing number of dimensions. Kernel and nearest neighbor methods are local neighborhood models; however, as the number of dimensions increases, the "neighborhood" of a point becomes much larger and harder to define well. Additionally, the density in the tails becomes much more important in higher dimensions; in fact, almost every portion of the domain takes on the property of becoming low-probability space [232]. As Hand argues, "kernel models are really practical only for relatively low-dimensional problems" [134]. Silverman shows how the required sample size in order to achieve a given accuracy using kernel methods grows with the dimensionality of the problem. Assuming a unit multivariate normal distribution and using a normal kernel, selecting the window width to minimize the mean square error, and then estimating the density at the point $\mathbf{0}$ to within relative error of 0.1 requires only 4 data points in 1 dimension but 2,790 points in 6 dimensions and 842,000 data points for 10 dimensions [232, p. 93–94].

### 7.1.5 Summary of Foundational Issues

These foundational issues lead to the following sub-Research Questions:

**Research Question (RQ3.4)**: What is the appropriate way to represent and evaluate probability distributions?

- (**RQ3.4.1**): Should probability distributions be modeled as parametric or non-parametric distributions?

**Research Question (RQ3.5)**: How can emergent behavior be identified in higher-dimensional systems?

The issue of high dimensionality raised by **RQ3.5** has already been partially addressed. It is clear from the literature that multi-dimensional probability estimation is difficult. Therefore, the methodology presented in this chapter specifically avoids having to model multi-dimensional probability distributions explicitly. Rather, it treats the design space as multi-dimensional but allows the probability distribution of the system behavior to stay a univariate function of the response function.

**Claim:** Density estimation of high dimensional probabilities is difficult. A design space methodology that can take advantage of low-dimensional estimates will be more useful than trying to estimate the entire high-dimensional probability distribution.

## *7.2 Methodology Formulation*

The Systematic Exploration for Emergence Detection (SEED) methodology is shown below in Figure 39. The purpose and development of each step in the methodology will be discussed below. The methodology consists of three main phases: data collection, data analysis, and data exploration. The data collection phases consists of selecting the experimental design and executing the simulation to build a database of results. The data analysis phase evaluates the Design Space Divergence measure that was proposed as part of this research. The data exploration phase attempts to locate regions of high Design Space Divergence measures within the design space.

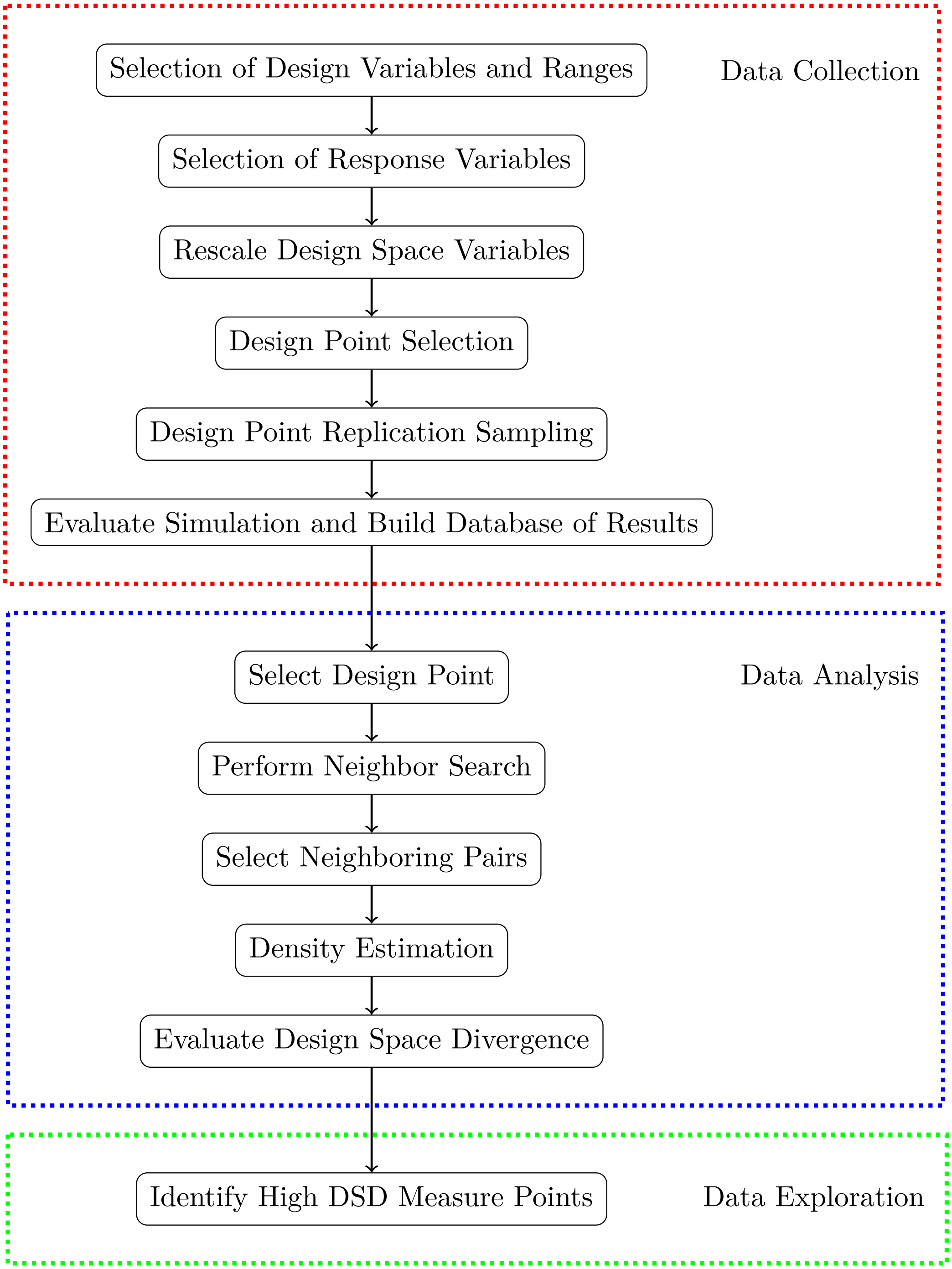
Selection of Design Variables and Ranges
Data Collection
Selection of Response Variables
Rescale Design Space Variables
Design Point Selection
Design Point Replication Sampling
Evaluate Simulation and Build Database of Results
Select Design Point
Data Analysis
Perform Neighbor Search
Select Neighboring Pairs
Density Estimation
Evaluate Design Space Divergence
Identify High DSD Measure Points
Data Exploration

**Figure 39:** Phases of SEED methodology

This methodology performs a systematic exploration of the design space for the variables and ranges specified. The experimental design, which consists of the DOE design and the number of replications at each point, is selected to efficiently sample the design space and to characterize the probabilistic response(s) at each point. The simulation under investigation is then executed according to the experimental design. The results of the simulation are stored in a database for all of the responses of interest and for every replication. Since this is a data-driven approach, all of the data must be saved and not just the statistics. Once the database of results is complete, the data analysis portion of the methodology is executed. The data analysis phase can be computationally expensive due to evaluating every neighbor pair at every design point. The result of this is another large database of Design Space Divergence measures for every local neighbor pair.

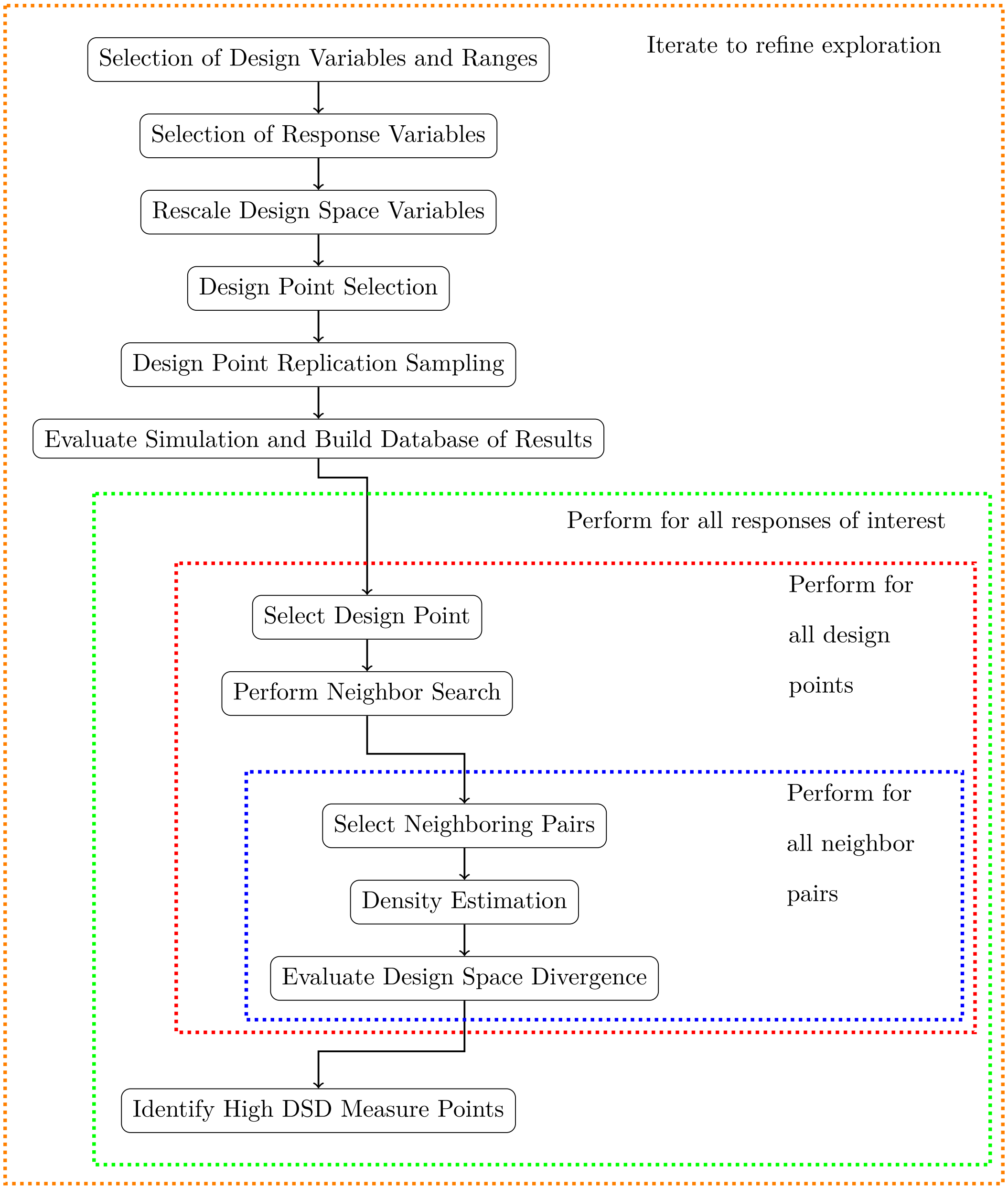
Iterate to refine exploration
Selection of Design Variables and Ranges
Selection of Response Variables
Rescale Design Space Variables
Design Point Selection
Design Point Replication Sampling
Evaluate Simulation and Build Database of Results
Perform for all responses of interest
Perform for all design points
Select Design Point
Perform Neighbor Search
Perform for all neighbor pairs
Select Neighboring Pairs
Density Estimation
Evaluate Design Space Divergence
Identify High DSD Measure Points

**Figure 40:** The iterative nature of the SEED methodology

The final phase of the SEED methodology is the data exploration phase. In this phase, the goal is to identify regions of the design space that have a large DSD measure. The techniques used in this phase will depend on the dimensionality of the problem and the type of structure in the results. For small problems, visualization techniques may be the most straight forward approach. For a large number of dimensions, machine learning and data mining techniques may be needed to assist the designer in identifying trends in the data.

### 7.2.1 Selection of Design Variables and Ranges

The design variables are the system parameters of interest during the design process. They represent the variables that will be investigated for their effect on the system behavior as measured by the response variables. Design space exploration is typically sequential in nature [200, pp. 10–12] where variables and their ranges are modified based on significance and regions of interest in an iterative process. If some variables are determined to be less important in influencing the responses of interest, they should be removed from the exploration to reduce the dimensionality of the problem. The ranges for each variable may also need to be adjusted to eliminate uninteresting or infeasible values from the design space. A common technique for downselecting variables is through the use of screening designs to test the influence of all of the variables on the variability of the response. However, common designs like $2^k$ factorial designs may be flawed for the design spaces of complex systems due to strong nonlinearity and localized effects (i.e., variables that exhibit conditional dependence, which may be common for emergent effects). A $2^k$ factorial design tests each variable at only the high and low values and assumes linearity in between. Estimates of the main effects of each variable are based on an averaging across all of the other factors and levels. If the local sensitivity of a variable is significantly different from the global sensitivity, this averaging may mask the local effects and lead to variables being incorrectly classified as insignificant.

The variables should either represent continuous quantities or at least be on an interval scale. Since the denominator in the Design Space Divergence measure is the distance between design points, the difference between two values should represent a meaningful quantity. Although continuous variables work best, discrete values are acceptable if the variable is on the interval scale so that differences between values is meaningful. However, binary values or other coded values (e.g., a variable that takes on the discrete values 1-5 that represents a system's operating mode) would most likely lead to less interpretable results. Although there is no absolute restriction against using coded variables, their use should done with care.

It is also necessary to select the range of values for each variable. This information is necessary for the design space point selection step as well as the design space rescaling step. The range of values should represent the range of interest for the exploration.

### 7.2.2 Selection of Response Variables

The response variables are the measures of system performance or function. They should be a measure of the system's high-level behavior. The Design Space Divergence measure can only be applied to a single variable; however, Design Space Divergence calculations can be carried out for each response variable of interest. This may be useful if the best measures of system performance have not been identified yet or if various behaviors manifest themselves only in one system metric and not another.

### 7.2.3 Rescale Design Space Variables

The set of all design variables forms an $n$-dimensional hypercube where $n$ is the number of design variables and each dimension represents a design variable. However, it is important to re-scale each dimension to remove coordinate scaling effects. For example, if two design variables in an aircraft design problem include the wing planform area and thrust to weight ratio, without rescaling, the distance between design points in this design space will be dominated by the wing area while the thrust to weight ratio will be negligible. To get around this problem, each variable is transformed so that the design variable's interval of interest is mapped onto the unit scale. This is a straightforward step where each variable is shifted left by the minimum value over the interval of interest so that the scale starts at zero. The second step is to divide by the interval's maximum value so that the new transformed scale is on the unit interval $[0, 1]$. This is shown graphically in Figure 41.

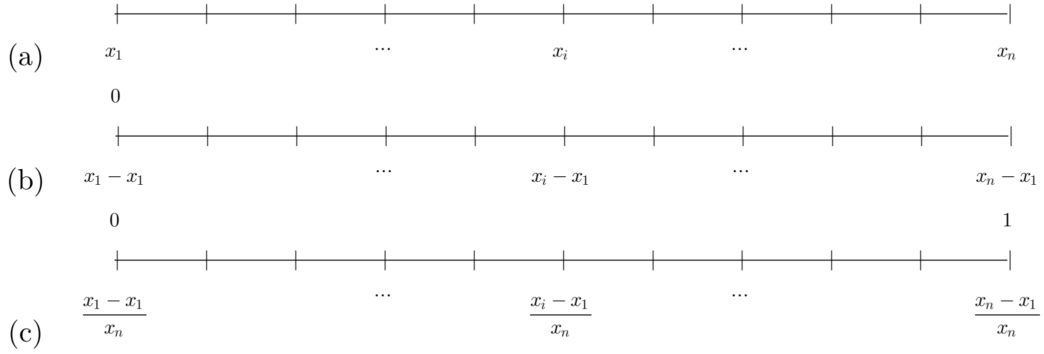

**Figure 41:** Design variable coordinate transformation: (a) original, (b) shifted, (c) shifted and scaled

This is performed for each variable resulting in an $n$-dimensional unit hypercube of design variables $\mathbf{x} \in \mathbf{I}^n$ where $\mathbf{I}^n$ is the $n$-dimensional unit interval. Since the data set is discrete, the $i$th position of the rescaled variable $\hat{x}$ is calculated as shown in Equation 13.

$$\hat{x_i} = \frac{x_i - x_{min}}{x_{max}} \tag{13}$$

### 7.2.4 Design Point Selection

This purpose of this methodology is for design space exploration, so design points should be selected to sample the space as thoroughly as possible. Design of Experiments (DOE) methods provide a way to efficiently sample points from a space. Traditional DOE designs are based on full factorial, fractional factorial, and other designs that primarily sample from the boundaries of the design space.

Full factorial designs generate points in an $n$-dimensional lattice. This type of design samples the space very thoroughly but the number of points grows exponentially with the number of dimensions. This design generates design points at all combinations of levels for all factors. The number of design points is $m^n$, where $m$ is the number of levels and $n$ is the number of factors (i.e., dimensions). This makes this design impractical for problems with more than a small number of dimensions. For many experimental designs, two levels are used resulting in an $2^n$ design; however, since there are only two levels per variable, this design requires linearity over the interval in order to be valid.

Fractional factorial designs use a fraction of the full factorial design. If the design is used for the purpose of factor estimation (i.e., building a regression model) and the sparsity of effects principle is valid (i.e., only a small number of factors are important and higher-order interactions are not important), a fractional factorial design can be used [10].

Space-filling designs attempt to fill the interior of the design space as uniformly as possible given the number of design points required. Response surface designs are often used for sequential designs, where progressively smaller regions of the design space are modeled using the polynomial model. However, response surface designs are often not appropriate for capturing model over the entire design space or for more complex models. Space-filling designs are better suited for complex models and where the entire design space is required [200, 28, p. 483]. Latin hypercube sampling (LHS) [182] is a popular space-filling design. In LHS, each dimension is broken up into bins of equal probability. For a given number of samples, each sample is placed randomly inside a bin and the points are distributed so that there is only one sample per bin for any one dimensional projection [126].

A number DOE designs exist for a specific modeling purpose. Screening designs sample variables at the low and high values in order to determine the relative effect that variable has on the response. For example, Plackett-Burman designs can be used to select important main effects while neglecting interactions [10]. Central-composite designs sample the upper and lower values plus central points. This design is good for creating polynomial response surface models [22]; however, the sparse interior sampling requires the system to be relatively well behaved in order to make any conclusions about how the variables affect the response and where within that interval interesting behavior may be found.

As an alternative to these DOE designs, adaptive design point selection is also possible. However, this option would require additional research in order to develop an algorithm that includes a convergence/stopping criterion and samples points in a way that maintains good discriminability in the results. Adaptive sampling methods will not be examined in this research and is left for future research. A summary of the methodological alternatives for the design point selection is shown in Table 6.

**Table 6:** Methodological alternatives: design point selection

| **Methodology Step** | **Primary Alternatives** | **Secondary Alternatives** |
|---|---|---|
| Design Point Selection | Design of Experiments (DOE) | Full Factorial |
| | | Space Filling Design |
| | Adaptive | - |

#### *7.2.4.1 DOE Designs*

Several DOE designs will be investigated in this research. Since this is a new methodology, only several basic DOE designs will be investigated as this time: full factorial designs and space-filling designs with varying numbers of design points. Two representative DOE designs are shown in Figure 42.

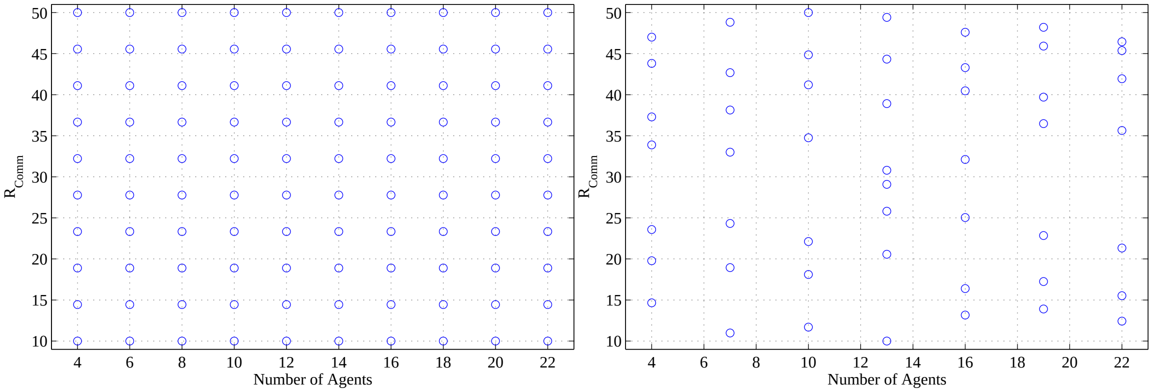

**Figure 42:** Full factorial design for 2 variables, 10 levels (left); space filling design for 2 variables, 50 runs (right)

The appropriateness of the various DOE designs will be tested as part of **Experiment A**.

**Experiment**: This investigation will be carried out as **Experiment A: Design Space Sampling** (Section 8.1). See the corresponding section in the Experiments chapter for the results and discussion of this experiment.

### 7.2.5 Design Point Replication Sampling

Since complex systems are inherently stochastic, each design point must be replicated many times in order to build a representative probability distribution describing the response.

For physical experiments, replication is the use of repeated runs of the experiment in order to estimate experimental error [10]; however, in computer simulations, the same type of experimental error (e.g., bias due to experimenter, day-to-day variation in experimental conditions, variation in material suppliers) is not present. For computer simulations, the stochasticity is inherent in the simulation model. Two possible approaches for determining the number of replications are using a predetermined number or using an adaptive method that stops the replications once the probability distribution has converged according to a given criterion.

#### *7.2.5.1 Fixed Number*

The fixed number of replication samples is the simplest choice. Each design point is replicated the specified number of times. If a fixed number of replications produces acceptable results across the entire design space, this method is an adequate choice. For this research, a fixed number of replications will be used.

#### *7.2.5.2 Adaptive Number*

There are many examples of complex systems having a landscape of behaviors, where one type of behavior is dominant in one region of the design space and a qualitatively different behavior is dominant in another. In this situation, it is likely that each behavior will be best sampled a different number of times. For example, if one region produces either deterministic or purely random outcomes, a smaller number of replications are necessary. On the other hand, more "interesting" behavior will require more samples in order to fully characterize features like multimodality and fat tails. Additional research will be necessary to use an adaptive approach for selecting the number of replications at each design point. A convergence/stopping criterion would need to be developed.

A summary of the methodological alternatives for the design point replication sampling is shown in Table 7.

**Table 7:** Methodological alternatives: design point replication sampling

| Methodology Step | Primary Alternatives | Secondary Alternatives |
|---|---|---|
| Design Point Replication Sampling | Fixed Number | - |
| | Adaptive Number | - |

### 7.2.6 Search for Nearest Neighbors

Since this method relies on evaluating neighboring pairs of design points, identifying the neighbors of a design point is necessary. A neighbor can be defined as either the closest $m$ points or all of the points that fall within a specified distance of a point in $k$-dimensional space with sample size $N$. Many techniques exist for performing this search. The most straightforward approach is the exhaustive search where all pairs are evaluated as potential neighbors. However, exhaustive searches are both computationally and memory intensive, so partitioning methods exist that cut down on the number of evaluations that have to be done.

#### *7.2.6.1 Exhaustive*

Exhaustive searches for neighbors is straightforward. Every pair of data points is selected and the distance between them is evaluated and stored. This list is then searched to find points that are neighbors as defined by the search criteria. However, as the number of points in the data set grows, the number of evaluations required explodes. An exhaustive search requires an evaluation and storage of every pair of data points. The computational complexity of an exhaustive search will be given by the combination of all the data points. This can be calculated using the binomial coefficient as shown in Equation 14, where $n$ will be the number of data points and $k = 2$ to find all of the pair-wise combinations.

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \tag{14}$$

$$\binom{n}{2} = \frac{n!}{2\,(n-2)!} = \frac{n\,(n-1)\,(n-2)!}{2\,(n-2)!} = \frac{n\,(n-1)}{2}$$

$$\binom{n}{2} = \mathcal{O}\left(n^2\right)$$

#### *7.2.6.2 Partitioning Method*

Partitioning methods attempt to get around the explosion in the number of evaluations in the exhaustive search by pruning possible candidates for neighbors. Friedman et al. [116] proposed a nearest neighbor search algorithm based on a $k$-dimensional tree search. $k$-dimensional (shortened to $k$-d) trees are a generalization of the binary tree. The data is partitioned along a discriminating dimension at a partition value. The goal is to partition the data so that the average number of comparisons is small compared to the original data set. Since the distribution of data points is not known a priori, the algorithm is optimized to minimize the expected cost of the search (by adjusting the discriminating dimension, the partition value, and the number of data points at each terminal node in the tree). At every nonterminal node, the dimension with the largest spread in values is chosen as the discrimination variable. Along this discrimination dimension, the median value is chosen as the partition value, as it maximizes the information content in a binary choice when the two alternatives are equally likely. The outcome of the algorithm is a partitioning scheme which divides the space into hypercubes, each containing a similar number of data points. The algorithm computational complexity is proportional to $kN \log N$ while the storage requirement is proportional to $N$. The $k$-d tree approach can work with a variety of distance measures as discussed in the next section.

A summary of the methodological alternatives for performing the search for nearest neighbors is shown in Table 8.

**Table 8:** Methodological alternatives: search for nearest neighbors

| **Methodology Step** | **Primary Alternatives** | **Secondary Alternatives** |
|---|---|---|
| Nearest Neighbor Search | Exhaustive | - |
| | Partitioning Method | $k$-d trees |
| | | Heuristic methods |

### 7.2.7 Neighbor Distance Measure

Distance is measure of dissimilarity between two points. If this dissimilarity measure has the properties of symmetry, monotonicity, and obeys the triangle inequality, it is a metric distance. Distance can be defined using a number of different measures. One of the most commonly used measures are the vector space $p$-norms (generally equivalent to $L^p$-space), as shown in Equation 15.

$$D_p\left(X,Y\right)=\left[\sum_{i=1}^{k}\left|X\left(i\right)-Y\left(i\right)\right|^p\right]^{1/p} \tag{15}$$

The Euclidean ($L^2$-norm) distance is well-known and often used. However, the $L^1$ and $L^\infty$-norms are also possible choices. Table 9 shows a list of vectors from 1 to 10 dimensions where one unit step is taken in each available dimension. Figure 43 shows the distance of each vector from the origin versus the number of dimensions for the $L^1$, $L^2$, and $L^\infty$-norms. This figure shows that using an $L^1$-norm, each additional dimensional adds an equal increment to the distance metric. The $L^2$-norm is quadratic and each additional dimension adds a smaller increment as the number of dimensions increases. This has the negative consequence that, for high dimensional spaces, points become more similar to each other in distance. The $L^\infty$-norm is not dependent on the number of dimensions and each vector has the same distance from the origin.

The choice of which distance measure to use in a design space is not obvious. Using the $L^2$-norm is often recommended because it is well understood and generally well-behaved. However, in the context of a parameter space, it certainly warrants careful examination if the $L^2$-norm really captures the "closeness" of two neighbors. As an example, in a 4 dimensional space using the $L^2$-norm, taking 1 step in each dimension has a distance of 2 units from the origin. However, this means that taking 2 steps in one dimension is the same as taking 1 step in each of 4 dimensions.

**Table 9:** Multi-dimensional vectors with a unit step in each dimension

| |
|---|
| X1 = [1] |
| X2 = [1, 1] |
| X3 = [1, 1, 1] |
| X4 = [1, 1, 1, 1] |
| X5 = [1, 1, 1, 1, 1] |
| X6 = [1, 1, 1, 1, 1, 1] |
| X7 = [1, 1, 1, 1, 1, 1, 1] |
| X8 = [1, 1, 1, 1, 1, 1, 1, 1] |
| X9 = [1, 1, 1, 1, 1, 1, 1, 1, 1] |
| X10 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1] |

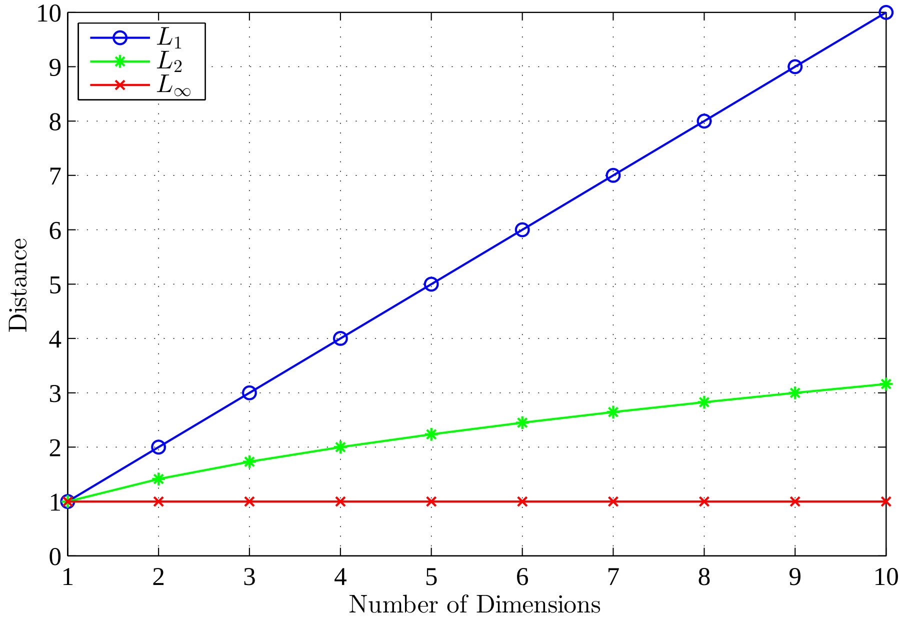

**Figure 43:** Distance measures from origin versus number of dimensions

Within the context of this methodology, the choice of distance measure is used as a selection criterion for determining which design points fall within a local neighborhood. This is shown in Figure 44 where the Chebyshev neighborhood includes corner points in a grid structure while the Euclidean neighborhood excludes corner points.

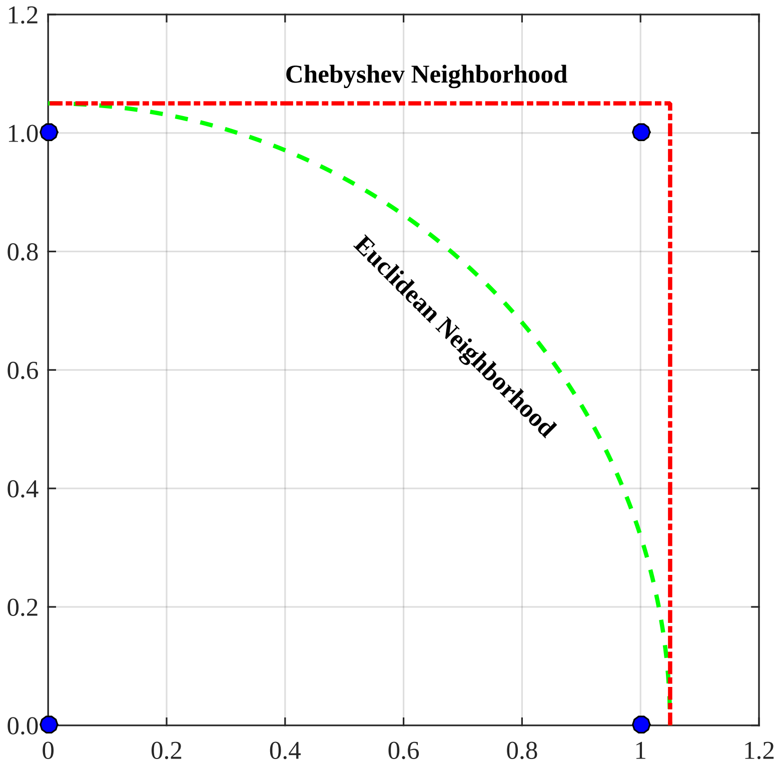

**Figure 44:** Neighborhoods using Euclidean and Chebyshev distance measures

#### *7.2.7.1 Euclidean ($L^2$-norm)*

Using the $L^2$-norm when finding the nearest neighbors is demonstrated on the four variable design space for the Distributed-MASS problem in Figure 45. In this problem, the design space points are selected using a full factorial design with an equal number of levels for each variable. This results in an equally spaced lattice with neighbors at equal distances in all dimensions. When the neighbor search is performed, one neighbor at the next higher level and the nearest lower level is found, if available within the design space. The important point to notice is that corner points are not selected since they are at a greater distance than any of the single dimension components. The $L^2$-norm can be visualized as the search for points within a hypersphere with a fixed radius from the current design point. This results in the $L^2$ measure generating pairs of neighbors along one dimension. This has the effect of performing a one-variable-at-a-time analysis.

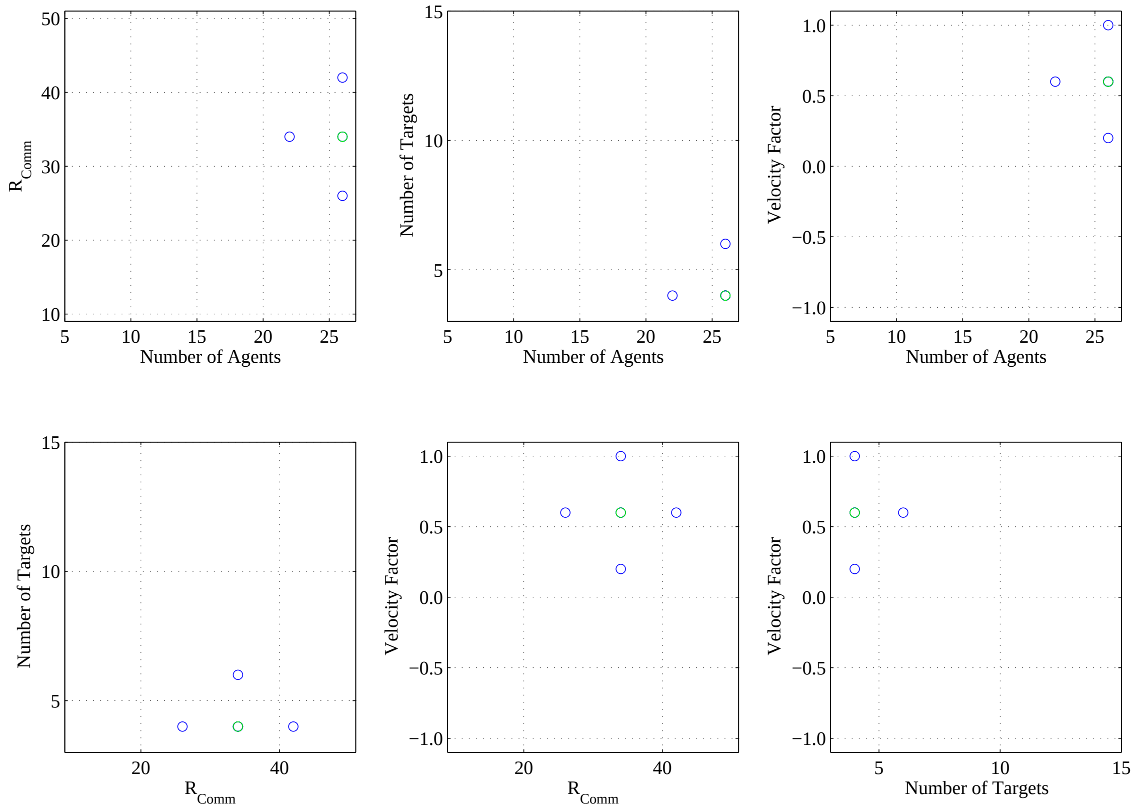

**Figure 45:** Nearest neighbors using Euclidean distance measure

### *7.2.7.2 Chebyshev ($L^\infty$-norm)*

Since the $L^\infty$-norm is based on the maximum distance along any one dimension, the search for neighbors can be visualized as the selection of all points within a hypercube of fixed dimension centered at the current design point. This search will return corner points of all possible combinations of dimensions. This allows the evaluation of the same neighboring points as the $L^2$-norm plus all of the interactions between those points. However, the curse of dimensionality leads to an explosion in the number of neighboring points for high dimensional spaces.

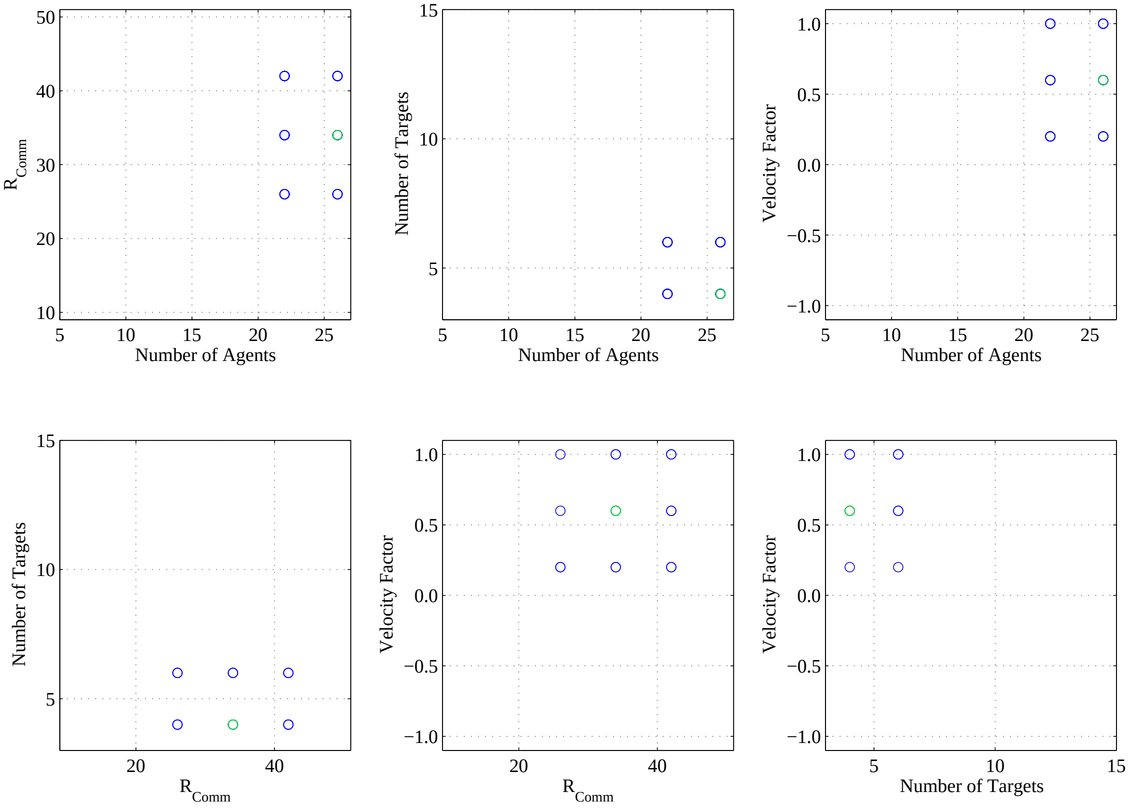

**Figure 46:** Nearest neighbors using Chebyshev distance measure

A summary of the methodological alternatives for selecting the distance measure is shown in Table 10.

**Table 10:** Methodological alternatives: neighbor distance measure

| **Methodology Step** | **Primary Alternatives** | **Secondary Alternatives** |
|---|---|---|
| Neighbor Distance Measure | Manhattan ($L^1$) | - |
| | Euclidean ($L^2$) | - |
| | Chebyshev ($L^\infty$) | - |

### 7.2.8 Density Estimation Problem: Parametric versus Nonparametric Distributions

Density estimation attempts to uncover the underlying probability density function of a random variable. The goal of density estimation is to develop the best estimate that we can of the true probability density function, $p(x)$. Our best estimate is denoted $\hat{p}(x)$. The estimation is performed using observations of the random variable $x$, which is a vector of $[x_1, \ldots, x_n]$ where $x \in \mathbb{R}^r$, where $r$ is the dimension of the data and $n$ is the number of observations.

Density estimation can be broken down into *parametric* and *nonparametric* approaches. In the parametric approach, a functional form of the probability distribution (e.g., uniform, Gaussian, Poisson, exponential, power-law) is assumed. As the name implies, a parametric distribution can be completely characterized using a small number of parameters. Therefore, the goal of parametric density estimation is to estimate the parameters that best describe the underlying distribution. On the other hand, for situations where a parametric probability distribution cannot be assumed (as has been noted when dealing with complex systems), the nonparametric approach must be used.

Score functions allow us to choose a model which "best" describe a model. In a linear regression, the method of least squares acts as the scoring function used. However, for descriptive models with no target/predicted value, scoring is more difficult. Maximum likelihood are one of the most common approaches used. One of the problems with the maximum likelihood approach is that the error can be dominated by errors in the tails of the density function where the probabilities are negligible and, therefore, do not significantly contribute to the utility of the model [134, Ch. 7.3].

We desire a number of properties for a measure to be good:

- *Unbiased*: $E\left[\hat{p}\left(x\right)\right] = p\left(x\right)$, that is, our estimate approaches the true value at a given location.

- *Consistent*: $\hat{p}\left(x\right) \to p\left(x\right)$ for every $x \in \mathbb{R}^r$, our estimate approaches the true value over the entire domain.

- *Convergence rate*: Our measure has a good convergence rate as a function of the number of observations.

Many different performance measures exist, a number of which are shown are Table 11. These measures are used to evaluate the goodness of fit between the estimate and the true distribution. Of course, the true distribution is usually unknown, so the measures are between an estimate of the distribution and the observed data.

**Table 11:** Density estimation performance measures

| **Performance Measure** | **Equation** |
|---|---|
| Integrated Mean-Squared Error | $\int \left[\hat{p}\left(x\right) - p\left(x\right)\right]^2 dx$ |
| Integrated Absolute Error | $\int \left\|\hat{p}\left(x\right) - p\left(x\right)\right\| dx$ |
| Kullback-Leibler Relative Entropy | $\int \hat{p}\left(x\right) \log\left\{\frac{\hat{p}(x)}{p(x)}\right\} dx$ |
| Hellinger Distance | $\mathrm{HD}\left(m\right) = \left\{\int \left(\left[\hat{p}\left(x\right)\right]^{1/m} - \left[p\left(x\right)\right]^{1/m}\right)^m dx\right\}^{1/m}$ |

The $L^2$-based approaches are the most popular for nonparametric density estimation performance measures [151, p. 78]. $p$ is assumed to be square integrable. Mean-squared error, a commonly used measure, is integrated for all $x \in \mathbb{R}^r$ to yield the integrated mean-squared error (IMSE). In certain situations, a measure based on an $L^1$ measure is preferred. The $L^2$ approach gives less preference to the behavior at the tails of the density, which can lead to poor tail behavior. A common $L^1$ measure is the integrated absolute error (IAE), also known as total variation. Izenman notes that $L^1$ measures are often harder to compute than analogous $L^2$ results [151, p. 79]. Another issue is that $L^1$ measures can have discontinuities in their derivatives, while the squared-error $L^2$ measures are better behaved analytically [136, p. 20].

All performance measures require knowledge of the true distribution. Since the true distribution is not known, these performance measures cannot be calculated and will not be evaluated in this research. However, the information is presented here for background information.

#### *7.2.8.1 Parametric Distributions*

In general, the probability distributions of metrics of complex systems cannot be described using standard parametric distributions (e.g., Gaussian, Poisson, exponential). It is certainly possible that a parametric distribution can accurately capture a particular metric of interest; however, there is no reason to assume that this will be possible in the general case. In many complex systems, the distribution changes shape drastically throughout the design space in both mean, variance, and the existence of long-tails. The second point is closely related to the first: if the probability distributions cannot be captured by a parametric distribution (in which the probability distributions are reduced to the parameters that best match the observed data), a nonparametric representation must be used in order to capture the distribution of interest. Kernstine came to the same conclusion that complex systems are characterized by nonparametric and heteroscedastic distributions [155]. However, Kernstine attempted to capture the behavior of the mean and variance statistics across the design space without capturing the entire distribution. By only capturing the two distribution statistics, many possible indicators of emergent behavior, such as multimodal distributions, are neglected. In addition, unlike with parametric distribution, where the shape parameters uniquely define a distribution, there are potentially an infinite number of nonparametric distributions that can be represented by the same magnitude of a mean or variance measure.

**Observation**: Complex system behaviors must be characterized by the *complete* probability distribution for all possible outcomes.

While in some situations the Gaussian is an accurate and useful representation of a probability distribution of interest, complex systems are often characterized by a variety of other distributions. Erdi describes other common distributions in complex systems including multimodal distributions, lognormal and power law, including some of the mechanisms which might explain why these distribution shapes are found [109, ch. 6]. We simply have no way of predicting or assuming a distribution type a priori—we must assume that the distribution can take on any shape.

Throughout this thesis, it has been argued that a nonparametric approach should be taken when characterizing the probability distributions. I have argued earlier that this approach is more general and more conservative (i.e., safer). However, the approach of assuming a parametric distribution and then testing the validity of the assumption is a defensible alternative. Although assuming a nonparametric distribution is safer, this approach may needlessly decrease the statistical performance. On the other hand, assuming a parametric distribution when it is not warranted may increase the error in the estimates and lead to incorrect inferences. Regardless of the approach taken, the assumption of either a nonparametric or parametric distribution should be tested.

**Experiment**: This investigation will be carried out as **Experiment C: Parametric vs. Nonparametric Probability Distributions** (Chapter 8.3). See the corresponding section in the Experiments chapter for the results and discussion of this experiment.

**Conclusion**: For the system under investigation in this research, it is clear from the results of the experiment that a nonparametric approach is necessary. Although the result will often be context-dependent, it is quite likely that complex system behavior is characterized by nonparametric distributions.

#### *7.2.8.2 Nonparametric Methods*

There are a number of nonparametric density estimation methods available. The simplest form is the well-known histogram method, where the frequency of data points lying within fixed intervals (i.e., bins) is used to estimate the probability density. The naive estimator is an improvement from the histogram method by allowing the locations of the bins to be centered on the data points. Kernel methods use kernel weighting functions that allow for the creation of smooth probability density estimates. These nonparametric methods will be investigated below.

One of the simplest nonparametric approaches is the histogram method. Probability densities are estimated based on frequency counts in interval bins that span the domain. Despite their ubiquity, histograms suffer from a number of problems. The first is that histograms are sensitive to the choice of origin, which can lead to a substantially different estimates [151, pp. 80–81]. This effect is shown in Figure 47.

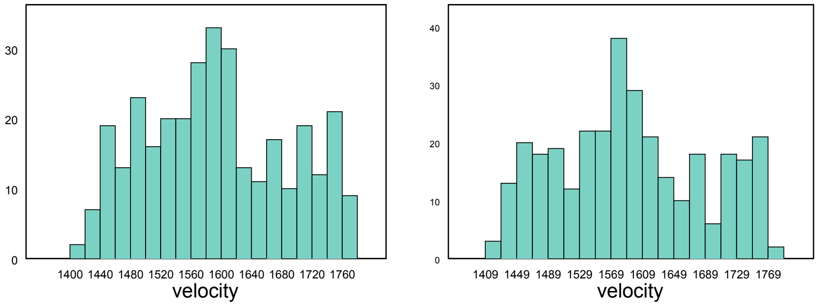

**Figure 47:** A shift in origin leads to significantly different estimate (from [151, p. 81])

The second major issue with histograms is that they require large sample sizes. As the dimensionality of the problem increases, the more difficult it is to have a significantly many number of observations in each bin due to the curse of dimensionality. A two dimensional histogram is shown in Figure 48.

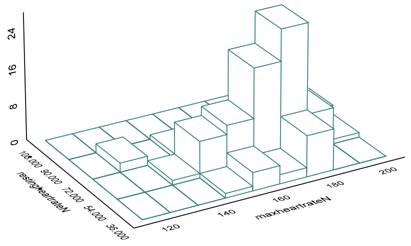

**Figure 48:** A multidimensional histogram (from [151, p. 86])

Other issues with histograms include its relatively poor convergence rate. Using the optimal bin width that yields the asymptotically optimal integrated mean-squared error, the convergence rate is $\mathcal{O}\left(n^{-2/3}\right)$, which is slower than most density estimators [151, p. 84]. Additionally, histograms are not smooth and discontinuities cause issues if derivatives of the probability density estimate are needed. Silverman [232, p. 10] argues that histograms are a poor choice as a density estimate when the estimates are used as intermediate steps within a methodology. These issues mean that a histogram is a poor choice as a nonparametric density estimate for the proposed methodology.

The next step up in sophistication from the histogram method is the naive estimator [232]. The form of the naive estimator is shown in Equation 16, where $n$ is the number of observations, $h$ is the bin width, $X_i$ is the $i$th observation, and $w$ is the weight function that is equal to $\frac{1}{2}$ if $|x| < 1$ and equal to 0 otherwise. The naive estimator is a generalization of the histogram where the bins are centered at each data point; therefore, the naive estimator still suffers from discontinuities.

$$\hat{f}\left(x\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}w\left(\frac{x - X_i}{h}\right) \tag{16}$$

#### *7.2.8.3 Nearest Neighbor Estimator*

The nearest neighbor method is attempt to adapt the amount of smoothing to the local density of the data [232]. The generalized kth-nearest neighbor estimator is shown in Equation 17, where $K$ is a kernel function integrating to unity.

$$\hat{f}\left(t\right) = \frac{1}{nd_k\left(t\right)}\sum_{i=1}^{n}K\left(\frac{t - X_i}{d_k\left(t\right)}\right) \tag{17}$$

The nearest neighbor estimator is not smooth, so it suffers from the same issues as the naive estimator. Also, estimates are sensitive to local noise, tend to have very heavy tails, and are generally an unsatisfactory overall estimator [232, p. 97]. These issues are not important when estimating the density at a single point; however, since the objective of the estimation of the entire density function, the kernel method is a better approach compared to the nearest neighbor method [232, p. 97].

#### *7.2.8.4 Kernel Methods*

Kernel methods are one of the most popular nonparametric approaches. Kernel methods are an improvement on the $k$-nearest neighbor approach, instead using weights that decrease smoothly based on distance from the target point [136, p. 17]. Given $n$ i.i.d. univariate observations $x_1$, $x_2$, ..., $x_n$, drawn from the density $p\left(x\right)$, the kernel density estimator $\hat{p}_h\left(x\right)$ for $x \in \mathbb{R}$ with $h > 0$ is given by Equation 18. $K$ is the kernel weighting function and $h$ is the window width (also called the bandwidth or smoothing parameter) of the local neighborhood which determines the smoothness of the density estimate. The kernel weight function integrates to unity: $\int_{-\infty}^{\infty} K\left(x\right)dx = 1$.

$$\hat{p}_h\left(x\right) = \frac{1}{nh}\sum_{i=1}^{n}K\left(\frac{x - x_i}{h}\right) \tag{18}$$

In this research, kernel methods will be considered sufficient for the requirements of the methodology. Other more advanced techniques exist such as orthogonal series estimation [107] and penalized likelihood density estimation [233]. However, unless a specific application requires the use of those methods, standard kernel methods should be sufficient. Kernel methods are memory-based methods, where the model is the data-set itself. Evaluation of the kernel function involves access to the entire data set. Radial basis functions, an approach used for regression, are similar to the kernel method with a symmetric $p$-dimensional kernel function.

#### *7.2.8.5 Selection of Kernel Function*

A number of widely used kernel functions are listed in Table 12. The Epanechnikov kernel is a weight function which minimizes the mean integrated square error of the estimate and has an efficiency of unity. It is clear that most kernel functions have efficiencies that are almost as good as the Epanechnikov. Although the bias of the estimate does depend on the function selected, the choice of kernel is not critical and can be selected on criteria other than efficiency [232].

**Table 12:** Kernel functions

| **Kernel** | **Functional Form** $K(t)$ | **Efficiency** |
|---|---|---|
| Biweight | $\frac{15}{16}\left(1-t^2\right)^2$ for $\lvert t\rvert < 1$ | 0.994 |
| Rectangular | $\frac{1}{2}$ for $\lvert t\rvert < 1$ | 0.930 |
| Triangular | $1-\lvert t\rvert$ for $\lvert t\rvert < 1$ | 0.986 |
| Gaussian | $\frac{1}{\sqrt{2\pi}}e^{-1/2t^2}$ | 0.951 |
| Epanechnikov | $\frac{3}{4}(1-\frac{1}{5}t^2)/\sqrt{5}$ for $\lvert t\rvert < \sqrt{5}$, 0 otherwise | 1.000 |

#### *7.2.8.6 Selection of Smoothing Parameter*

The window width (also called smoothing parameter or bandwidth) is an important parameter when performing the density estimation. The value of the window width balances the trade-off between random and systematic error [232, p. 40]. The optimal window width depends explicitly on unknown density $p(x)$, so the optimal window width cannot be determined a priori [151, p. 95]. Several methods exist to determine the window size. The first is *rule-of-thumb*, which is the simplest. Another method is *cross-validation*, but it is computationally intensive; therefore, it is a poor choice for large data sets. Least-squares cross-validation is an automatic method and aims at an optimal value of the smoothing parameter to minimize the integrated square error [232, p. 51]. Likelihood cross-validation uses leave-one-observation-out approach but can be sensitive to outliers [232]. If an optimal value of the smoothing parameter cannot be used, Silverman argues that it is generally better to under-smooth than to over-smooth and risk losing important features in the data [232, p. 43].

Scott [225, p. 131] describes the development of the normal kernel reference rule, shown in Equation 19, which estimates the optimal bandwidth parameter that minimizes the mean integrated squared error when using the normal kernel for univariate data. However, the optimal parameter depends on the *true* distribution (specifically the standard deviation, $\sigma$); however, since the true distribution is not known, the estimate for optimal bandwidth uses the estimated standard deviation, $\hat{\sigma}$. In fact, the large number of approaches that exist in the literature are often due to the fact that the quantities in the expressions are random variables and must be estimated from the data at hand. This means that any estimate is only optimal on average (or via some other statistic). Scott [225, p. 162] argues that AMISE-derived optimal bandwidth $h^*$ is an appropriate target.

$$h^* = \left(\frac{4}{3}\right)^{1/5} \sigma n^{-1/5} \approx 1.06\hat{\sigma} n^{-1/5} \tag{19}$$

Venables and Ripley [245] use a slightly modified form of the rule, shown in Equation 20, uses the minimum of either the estimated standard deviation or a value based on the interquartile range (IQR).

$$h = 1.06 n^{-1/5} \min\left(\hat{\sigma}, {}^{\mathrm{IQR}}/_{1.34}\right) \tag{20}$$

A summary of the methodological alternatives for performing the density estimation is shown in Table 13.

**Table 13:** Methodological alternatives: density estimation

| Methodology Step | Primary Alternatives | Secondary Alternatives |
|---|---|---|
| Density Estimation | Parametric | Distribution Family |
| | Histogram | - |
| | Kernel Method | Kernel function selection |
| | | Bandwidth parameter selection |

### 7.2.9 Statistical Distance

There are a number of ways to quantify the difference between two probability distributions on the same probability space. These quantities are zero when the two probability distributions are identical and increase as the distributions become more dissimilar. Since some of these quantities do not satisfy the requirement of symmetry and/or the triangle inequality, they are called divergences instead of distances. A number of distances and divergences, listed below, will be examined for their suitability in this research.

- Kullback-Leibler divergence
- Jensen-Shannon divergence
- Hellinger distance
- Bhattacharyya distance

The goal will be to identify which measures offer the best discriminability between two probability distributions while being well-behaved and relatively insensitive to noise. The measure must be equally sensitive to both shifts in mean and variation.

The probability density is estimated from the underlying data using the kernel method described above. The probability estimate is discretized over the support interval of the combined distributions. This allows the probability from each distribution to be evaluated at discrete points $p_i$ and $q_i$ from distributions $P$ and $Q$, respectively. This process is illustrated in Figure 49.

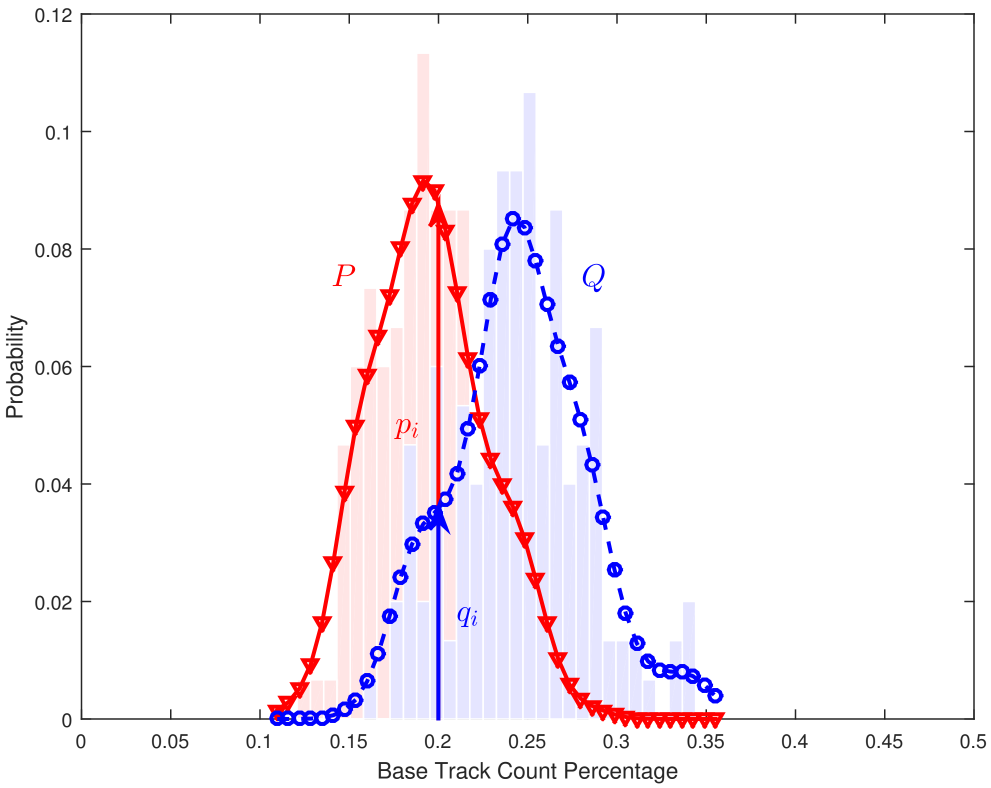

**Figure 49:** Discrete probability estimate

The discrete Kullback-Leibler divergence, $\mathrm{KL}\,(p, q)$, is shown below in Equation 33.

$$\mathrm{KL}\,(p, q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{21}$$

Since Kullback-Leibler is not symmetric with respect to $p$ and $q$, the simplest way to achieve a symmetric measure, $\mathrm{KL}_2$, is by taking the average of $\mathrm{KL}\,(p, q)$ and $\mathrm{KL}\,(q, p)$, as shown in Equation 22.

$$\mathrm{KL}_2\,(p, q) = \frac{1}{2}\mathrm{KL}\,(p, q) + \frac{1}{2}\mathrm{KL}\,(q, p) \tag{22}$$

$\mathrm{KL}_2$ is symmetric but is not finite-valued. To overcome this limitation, Jensen-Shannon divergence is based on the Kullback-Leibler divergence of each distribution relative to the average distribution of $p$ and $q$.

$$\begin{cases} \mathrm{JS}\,(p, q) = \frac{1}{2}\mathrm{KL}\,(p, r) + \frac{1}{2}\mathrm{KL}\,(q, r) \\ r = \frac{1}{2}\,(p + q) \end{cases} \tag{23}$$

The Hellinger distance, $\mathrm{Hel}\left(p,q\right)$, is shown below in Equation 24.

$$\mathrm{Hel}\left(p,q\right)=\frac{1}{\sqrt{2}}\sqrt{\sum_{i}\left(\sqrt{p_i}-\sqrt{q_i}\right)^2} \tag{24}$$

The Bhattacharyya distance, $\mathrm{BD}\left(p,q\right)$, is shown below in Equation 25, where BC is the Bhattacharyya coefficient. The Hellinger distance is related to the Bhattacharyya coefficient where $\mathrm{Hel}\left(p,q\right)=\sqrt{1-\mathrm{BC}\left(p,q\right)}$.

$$\begin{cases}\mathrm{BD}\left(p,q\right)=-\ln\left(\mathrm{BC}\left(p,q\right)\right)\\ \mathrm{BC}\left(p,q\right)=\sum_{i}\sqrt{p_i q_i}\end{cases} \tag{25}$$

A summary of the methodological alternatives for calculating the statistical distance is shown in Table 14.

**Table 14:** Methodological alternatives: statistical distance

| **Methodology Step** | **Primary Alternatives** | **Secondary Alternatives** |
|---|---|---|
| Statistical Distance | Kullback-Leibler divergence | - |
| | Jensen-Shannon divergence | - |
| | Hellinger distance | - |
| | Bhattacharyya distance | - |

Since there are a number of candidate statistical distances, an experiment will be carried out to characterize their performance and to help downselect the appropriate distance for use in the methodology.

**Experiment**: This investigation will be carried out as **Experiment B: Statistical Distance Measures** (Chapter 8.2). See the corresponding section in the Experiments chapter for the results and discussion of this experiment.

### 7.2.10 Evaluate Design Space Divergence

In order to formalize the methodology, following the approach taken in Morris and Mitchell [194], the response of the system, $Y$, is assumed to be a stochastic function defined over the design space $\mathbf{X}$. More specifically, the Design Space Divergence ($DSD$) between responses at two design points is a function of the distance between the two points. This is shown in Equation 26.

$$\mathrm{DSD}\left(Y\left(x_p\right)\|Y\left(x_q\right)\right) = F\left(d\left(x_p, x_q\right)\right) \tag{26}$$

Evans et al. [110] argue that there are two important steps when using simulations for exploratory design: finding “important” patterns and then determining causality. The proposed design space emergence measure attempts to capture elements of both of these tasks. The numerator in the measure captures the “importance” while the denominator captures an element of the causality issue. The Design Space Divergence measure proposed in this methodology is the ratio of the statistical distance to the design space distance between two points. Recalling that the statistical distance is a measure of the difference between two probability distributions, the probability distributions are defined as $P = Y\left(x_p\right)$ and $Q = Y\left(x_q\right)$. The design space distance is evaluated on the rescaled coordinate space as described in Chapter 7.2.3. The equation for the Design Space Divergence measure is shown below in 27.

$$\mathrm{DSD}\left(x\right) = \frac{\mathrm{Statistical\,Distance}\left(\mathrm{P},\mathrm{Q}\right)}{\mathrm{Design\,Space\,Distance}\left(\hat{x}_p, \hat{x}_q\right)} \tag{27}$$

This measure is evaluated according to the methodology steps defined earlier. The statistical distance function and the corresponding options were described earlier and were studied in Experiment B. The design space distance function also follows from the discussion above in the section on Neighbor Distance (Section 7.2.7). It is important to note that the Design Space Distance measure is independent of the distance measure used in the neighbor search portion of the methodology. In the neighbor search step, the distance measure was used for the *selection* of the neighboring points; however, any other distance measure can be used to evaluate the distance between points. As will be done in the Case Study, the neighbor search will carried out using the Chebyshev distance but the Euclidean distance will be used when evaluating the DSD measure. The Chebyshev measure considers the distance to be the maximum difference along any single dimension. This provides an easy way of selecting all of the neighbors including corner points; however, this measure does not match our intuitive understanding of how close two points are to each other. Therefore, using the Euclidean distance when evaluating the DSD measure is a more appropriate choice.

### 7.2.11 Identify High Design Space Divergence Measure Regions

The final step in the methodology is the exploration of the DSD measure data set to identify regions of the design space with high DSD values. There are a variety of techniques available to perform this step. If the number of dimensions is small, visual analytic techniques can be used to locate the data using common visualizations and interactive plotting. For higher dimensional data, machine learning techniques like clustering or dimensionality reduction techniques can be used to assist the user in identifying trends in the data. A summary of the methodological alternatives for performing the identification of high DSD measure regions is shown in Table 15.

**Table 15:** Methodological alternatives: identify high DSD measure points

| Methodology Step | Primary Alternatives | Secondary Alternatives |
|---|---|---|
| Identify High DSD Measure Points | Visual Analytics | - |
| | Machine Learning | Clustering |
| | | Dimensionality Reduction |
| | | ...(other) |

### 7.2.12 Iterate to Refine Exploration

The final step of the methodology is to use the information gained during the data exploration phase to refine design space. This refinement step can be used for a number of different purposes. Once a candidate region has been identified, a more detailed exploration in a smaller region can be performed to get a higher resolution picture of the search space. Design variables can be added or removed and their ranges modified to take advantage of a new understanding of the problem. This step of the methodology may be optional if additional refinement is deemed unnecessary.

## *7.3 Methodology Implementation*

The first phase (data collection) of the SEED methodology is relatively straight forward and will not need implementation details. The data exploration phase will depend on the resulting data, so it would be difficult to come up with an algorithm that will always be appropriate. However, the data analysis phase of the methodology is involved and will be detailed here. The pseudo-code for data analysis phase is given below. This algorithm was implemented in MathWorks MATLAB due to its data processing capabilities and the ability to access the results from the Distributed-MASS simulation runs.

**Algorithm 4** SEED Methodology Data Analysis

```
procedure ProcessData(DataResults, DOE)
    $X \leftarrow DOE$    ▷ Read in design point values
    $X_{scaled} \leftarrow ReScaleDesign(X)$    ▷ Rescale design space
    for $iDesignPt \leftarrow 1, nDesignPt$ do    ▷ Iterate through all design points
        $XDesignPt_{scaled} \leftarrow ReScaleDesign(XDesignPt)$    ▷ Rescale design point
        $Neighbors \leftarrow FindNeighbors(X_{scaled}, XDesignPt_{scaled})$    ▷ Find neighbors in design space
        for all $iNeighborPt \in Neighbors$ do
            $DataP \leftarrow$ DataResults($iDesignPt$)    ▷ Get simulation data
            $DataQ \leftarrow$ DataResults($iNeighborPt$)    ▷ Get simulation data
            $DensityP \leftarrow$ KernelMethod($DataP$,h)    ▷ Estimate density
            $DensityQ \leftarrow$ KernelMethod($DataQ$,h)    ▷ Estimate density
            $DataStatDist(iDesignPt, iNeighborPt) \leftarrow$ StatDist($DensityP, DensityQ$)    ▷ Calculate Stat. Dist.
            $XNeighbor_{scaled} \leftarrow X_{scaled}(iNeighborPt)$    ▷ Get neighbor location
            $NeighborDistance \leftarrow Distance(XDesignPt_{scaled}, XNeighbor_{scaled})$    ▷ Get neighbor distance
            $DataDSD(iDesignPt, iNeighborPt) \leftarrow DataStatDist/NeighborDistance$    ▷ Calculate Stat. Dist.
            $XMid(iDesignPt, iNeighborPt) \leftarrow$ CalculateMidPt(
                $XNeighbor_{scaled}, XDesignPt_{scaled}$)    ▷ Get midpoint between neighbors
        end for
    end for
end procedure
```

## 7.4 Methodology Evaluation

The final step is to test and evaluate the performance of the proposed methodology. This will done as part of the final experiment in this dissertation.

**Experiment**: This investigation will be carried out as **Experiment D: Evaluating the Methodology Performance** (Chapter 8.4). See the corresponding section in the Experiments chapter for the results and discussion of this experiment.

# CHAPTER VIII

# EXPERIMENTS

**Chapter Road Map**: The goal of this chapter is to present the results of the experiments conducted as part of this research.

## *8.1 Experiment A: Design Space Sampling*

**Research Question** for **Experiment A**.

- (**RQ3.2**): How can the design space of a complex system be systematically explored in order to identify emergent behavior?
    - (**RQ3.2.1**): How should the design points be distributed in the design space?

Three cases will be compared for this experiment. A full factorial design will be the first case. Although a full factorial exploration is usually not the most efficient, it explores the space uniformly and has the potential to improve statistical inferences due to its uniform sampling. Space-filling designs (e.g., Latin Hypercube Sampling) is known to be efficient for surrogate modeling, but its efficiency for divergence-based measures has yet to be proven. The resulting divergence measures will be compared to see if space-filling designs produce reliable and interpretable results (i.e., comparable resolution and discriminability compared to the full factorial design). The comparison will be made by visual inspection of the contour plots in the 2D design space.

The hypothesis for research question **RQ3.2.1** is akin to a null hypothesis where there is no significant difference between the two sampling strategies. The goal of this experiment is therefore to test whether a significant difference exists or not.

**Hypothesis 2 (HYP2)**: If there is a similar sampling density, there will be no significant difference in discriminability between sampling methods (i.e., both full factorial and space filling designs will yield the same similar divergence measures).

### 8.1.1 Full Factorial - 100 Points

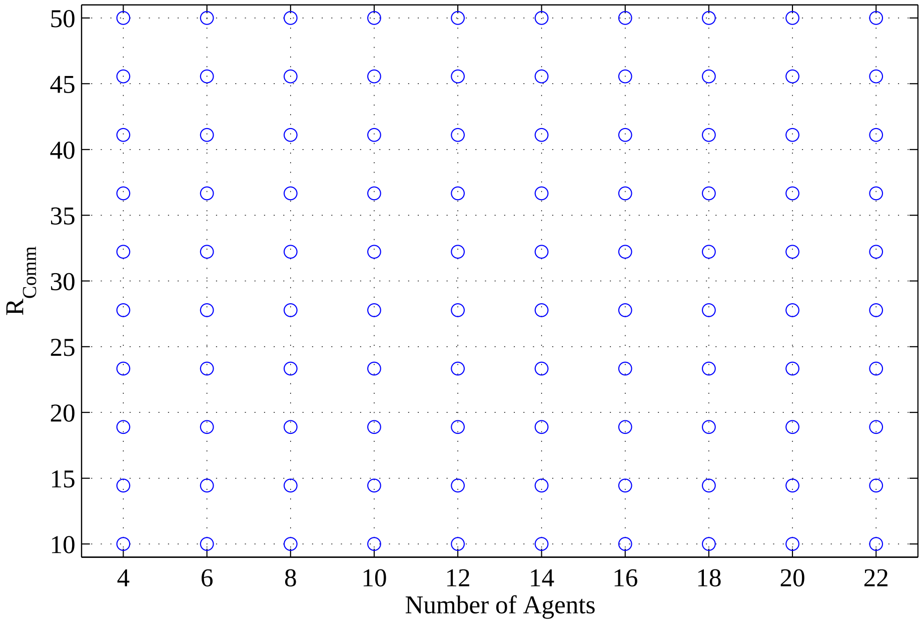

**Figure 50:** Full factorial design for 2 variables, 10 levels

Summary statistics for the execution of this design are shown in Table 16.

**Table 16:** Summary statistics for 100 case full factorial design

| **Parameter** | **Value** |
|---|---|
| Number of Cases | 100 |
| Number of Replications | 100 |

The variation of the mean and standard deviation of the BTTCP measure of system performance is shown in Figure 51.

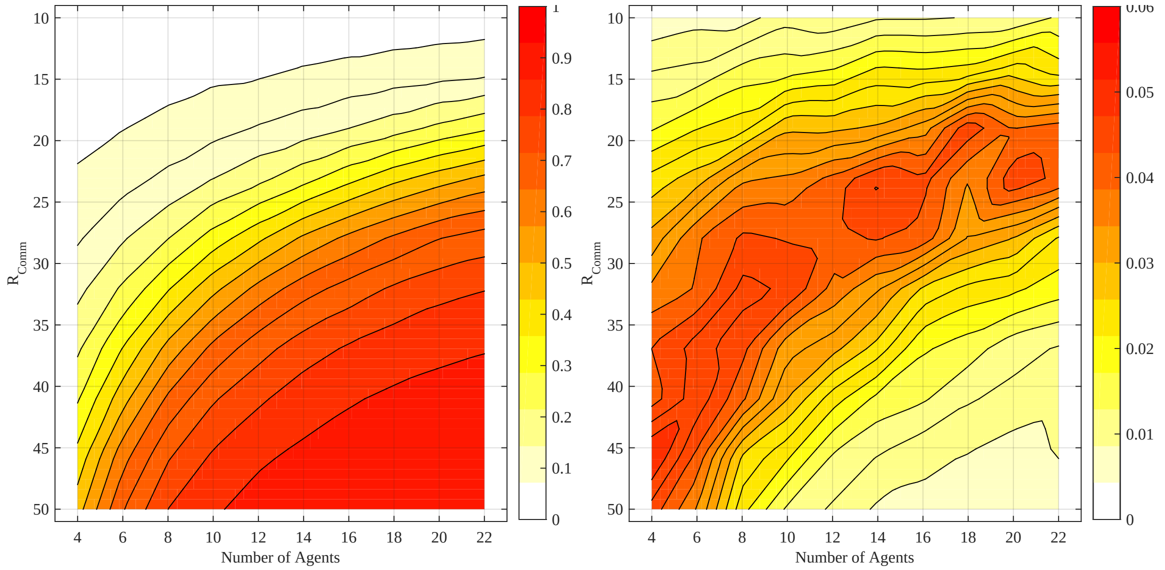

**Figure 51:** Contour plot of the mean (left) and standard deviation (right) of BTTCP as a function of design variables

As was shown in the initial characterization of the design space for the Distributed Multi-Agent Surveillance System problem (Chapter 6.3), a grid differencing scheme can be used as a way of understanding how the probability distribution changes across the design space. This approach is possible due to the full factorial design, which makes it easy to find neighbors and to ignore design space distances since the grid is consistent. This approach also only considers neighbors along each dimension and does not consider corner points in the neighbor search. The resulting contour plots are shown in Figure 52. This figure can be used as a reference for future analysis conducted as part of this experiment.

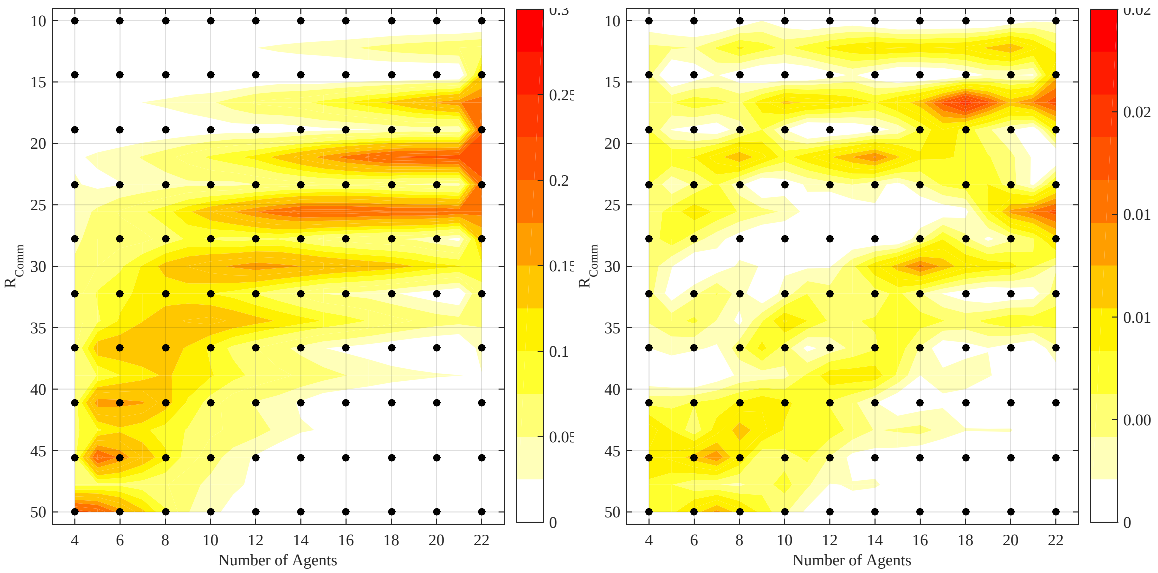

**Figure 52:** Contour plot of the grid differencing mean shift (left) and standard deviation shift (right) of BTTCP as a function of design variables

Carrying out the evaluation of the design space divergence measure leads to the results shown in the contour plot in Figure 53. This contour plot is generally as expected and provides the clearest picture of the divergence measure since the design points are relatively dense and the grid is consistent. This result will set the benchmark when evaluating the appropriateness of the space filling designs.

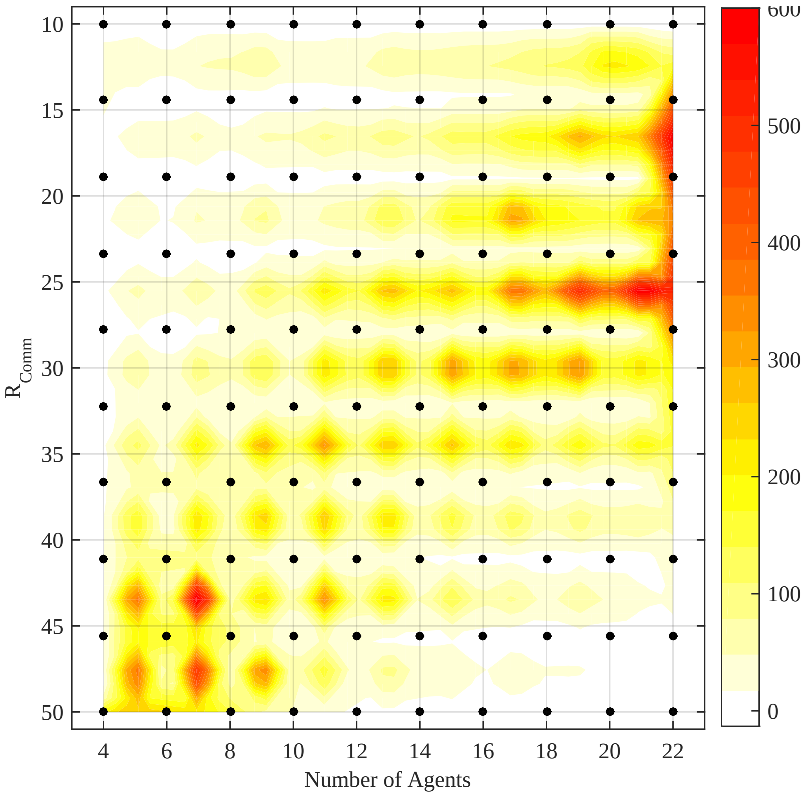

**Figure 53:** Contour plot of design space divergence measure as a function of design variables

### 8.1.2 Full Factorial - 100 Points (Refined)

An additional experiment was carried out to investigate whether the results are consistent and would refine with a finer grid. A 100 point full factorial design over a smaller portion of the design space was created and run. The range for the design variables is shown in Table 17. The resulting design is shown in Figure 54. Summary statistics for the execution of this design are shown in Table 18.

**Table 17:** Design space exploration variables for 100 case full factorial design, refined

| Design Variable | Min Value | Max Value |
|---|---|---|
| Number of Agents | 15 | 24 |
| Communication Range | 14 | 32 |
| Detection Range* | 7 | 16 |

* assumed to be 1/2 of the Communication Range

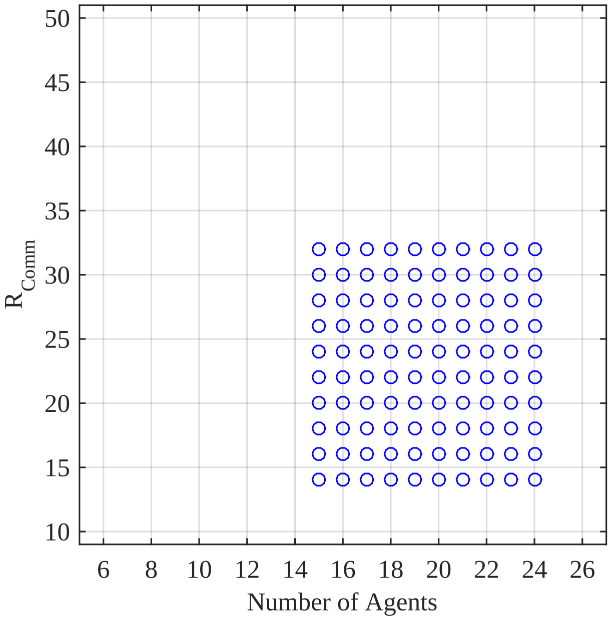

**Figure 54:** Full factorial design for 2 variables, 10 levels, refined

**Table 18:** Summary statistics for 100 case full factorial design, refined

| **Parameter** | **Value** |
|---|---|
| Number of Cases | 100 |
| Number of Replications | 200 |

The resulting Design Space Divergence measure is shown in Figure 55. The original plot of the DSD measure is shown on the left while the refined results are shown on the right. This shows that the DSD does refine and generally provides consistent results. Since the refined exploration is being performed over a much finer grid, neighboring points are closer together in both response and design space distance. The magnitude of the DSD measure is significantly smaller; however, the major features remain. In particular, there is a significant shift in behavior near a communication range of 25 and when the number of agents is approximately 20. However, it should be emphasized that a very fine grid is not necessarily desirable. Unless the transition behavior is very sharp (i.e., almost discontinuous), a very fine grid will not yield interesting results. There is an intermediate level of resolution that is fine enough to capture transition regions but sparse enough that there is a statistically significant difference of behavior between points. In the limit, the DSD measure is expected to approach $\frac{0}{0}$ as both the difference in statistical distributes is expected to go to zero while the design space distance also approaches zero. However, before that point is reached, the DSD measure is expected to be dominated by sampling noise.

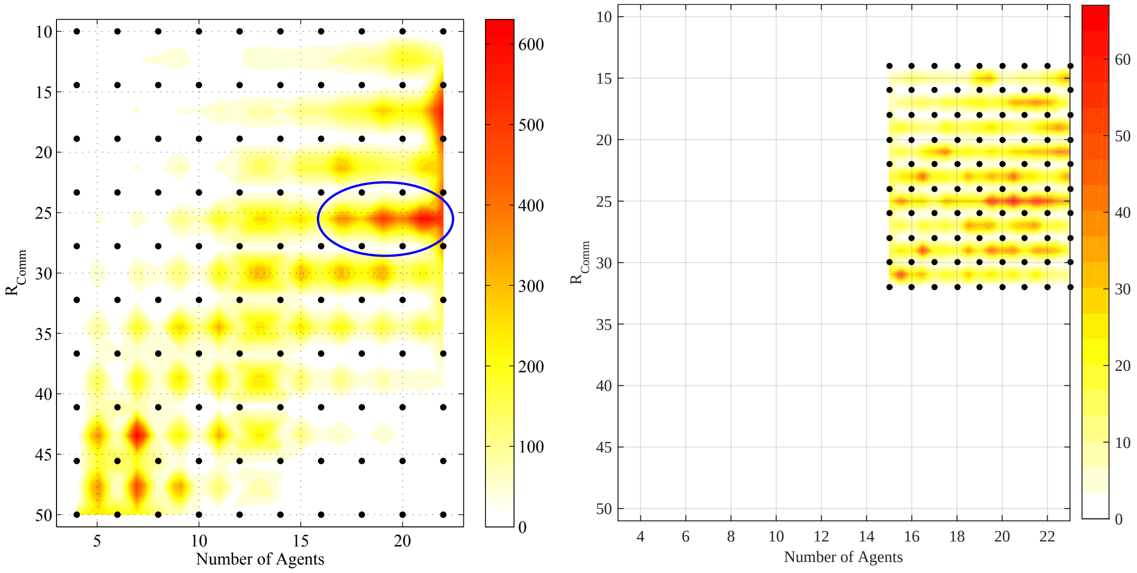

**Figure 55:** Contour plot of Design Space Divergence measure as a function of design variables for 100 case full factorial: original (left), refined (right)

### 8.1.3 Space Filling Design - 100 Points

The goal of this portion of the experiment is to evaluate the appropriateness of using space-filling designs to evaluate the design space divergence measure. In this experiment, a 100 point design space design was used. The space filling design was generated in JMP12 using 2 factors where the communication range was treated as a continuous variable and the number of agents was treated as a 7-level categorical variable. The resulting design is shown in Figure 56.

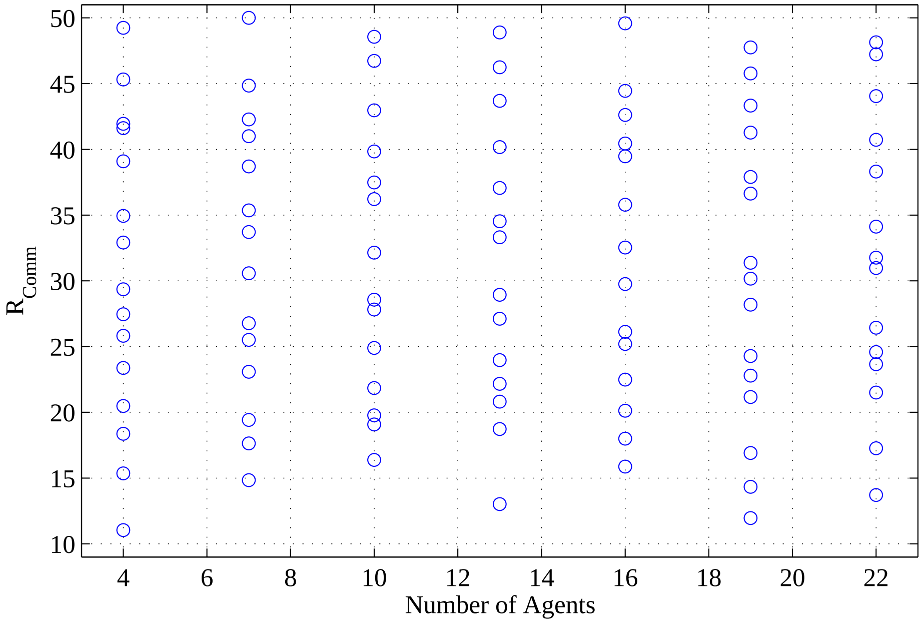

**Figure 56:** Space filling design for 2 variables, 100 runs

Summary statistics for the execution of this design are shown in Table 19.

**Table 19:** Summary statistics for 100 case space filling design

| **Parameter** | **Value** | **Units** |
|---|---|---|
| Number of Cases | 100 | |
| Number of Replications | 100 | |
| Total Run Time for Experiment | 16:03:08:28 | DD:HH:MM:SS |
| Average Run Time Per Replication | 139.37 | seconds |

The contour plot of the design space divergence measure for the 100 case space filling design is shown in Figure 57. Although there are similarities to the 100 case full factorial design, the result for the 100 case space filling design shows unexpected irregularities.

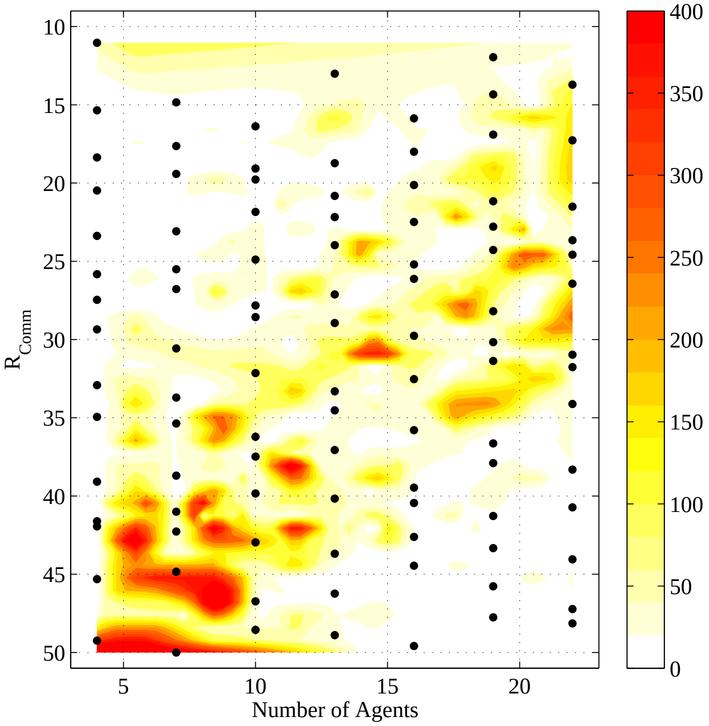

**Figure 57:** Contour plot of design space divergence measure as a function of design variables for 100 case space filling design

It appears that the cause of the irregularities is due to the irregular spacing and clustering of neighboring points when performing the analysis. A typical neighborhood of points is shown in Figure 58. This type of arrangement causes a large of number of solutions to be calculated in close proximity, which makes visualization of the results difficult. Some design points are also very close to each other and the differences in the underlying probabilities may be due mostly to sampling noise. Although it is expected that design points will have underlying distributions which converge to each other as two design points approach each other, it is likely that anything less than a large amount of samples will have sampling noise which keeps the divergence measure from reaching zero. This is further amplified by the design space divergence measure since the solutions have a very small design space distance between them.

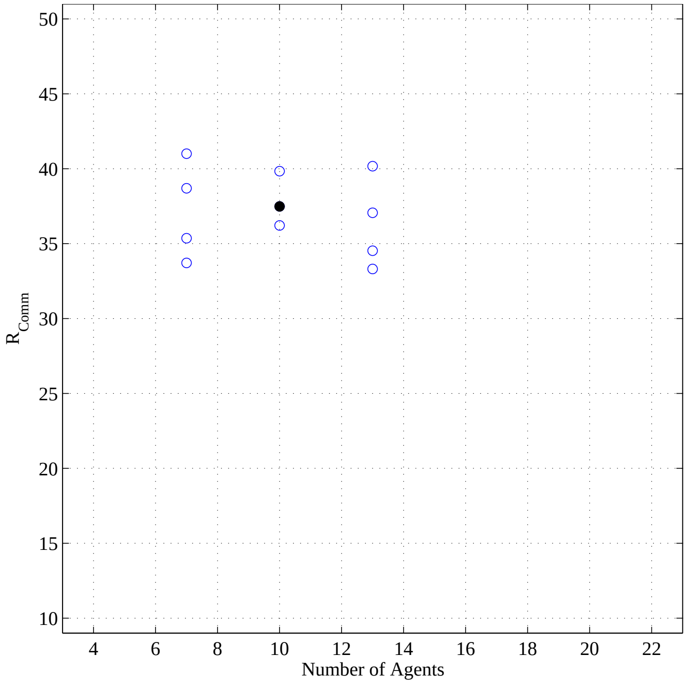

**Figure 58:** Representative nearest neighbor cluster for 100 case space filling design

### 8.1.4 Space Filling Design - 75 Points

A space filling design using 75 points, shown in Figure 59, will be analyzed in this portion of the experiment.

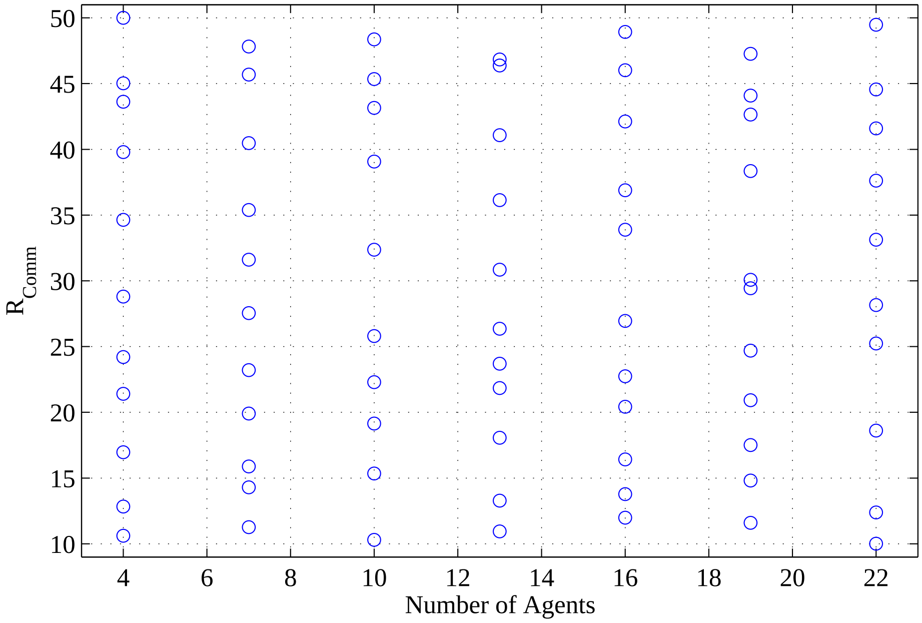

**Figure 59:** Space filling design for 2 variables, 75 runs

Summary statistics for the execution of this design are shown in Table 20.

**Table 20:** Summary statistics for 75 case space filling design

| **Parameter** | **Value** | **Units** |
|---|---|---|
| Number of Cases | 75 | |
| Number of Replications | 100 | |
| Total Run Time for Experiment | 11:13:23:14 | DD:HH:MM:SS |
| Average Run Time Per Replication | 133.15 | seconds |

The contour plot of the design space divergence measure for the 75 case space filling design is shown in Figure 60. This figure shows the expected trends as demonstrated in the 100 case full factorial design.

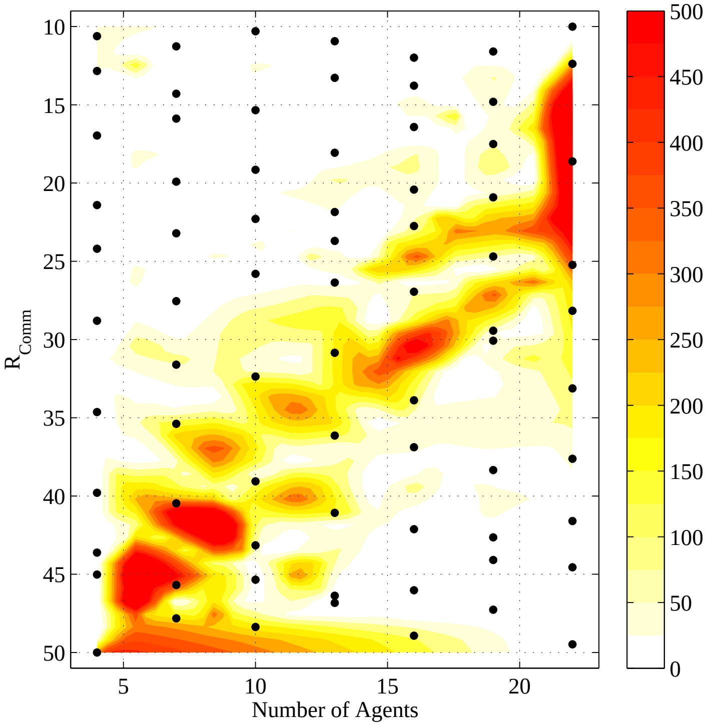

**Figure 60:** Contour plot of design space divergence measure as a function of design variables for 75 case space filling design

These results do not suffer from the irregularities seen in the 100 case space filling design. A typical neighborhood of points for this design is shown in Figure 61. This distribution of points in the neighborhood is close to the ideal scenario where neighboring points are evenly distributed in the space surrounding the reference design point.

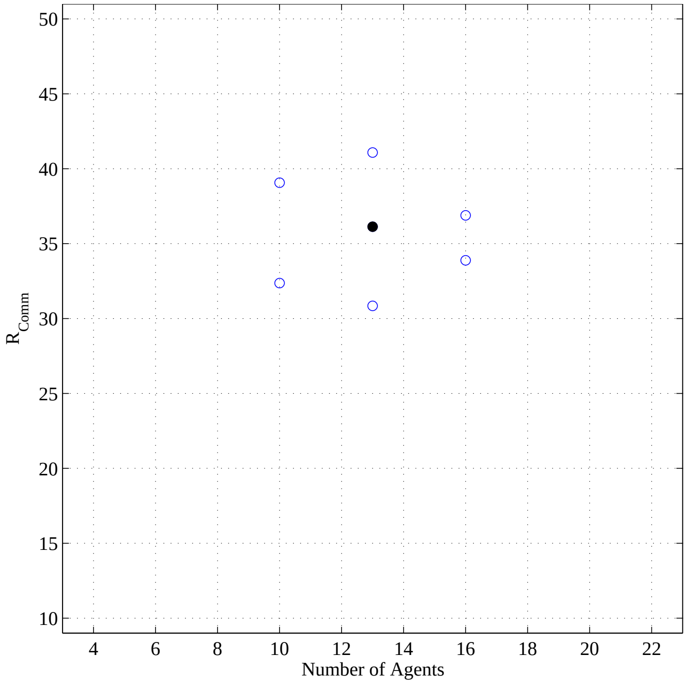

**Figure 61:** Representative nearest neighbor cluster for 75 case space filling design

### 8.1.5 Space Filling Design - 50 Points

A space filling design using 50 points, shown in Figure 62, will be analyzed in this portion of the experiment.

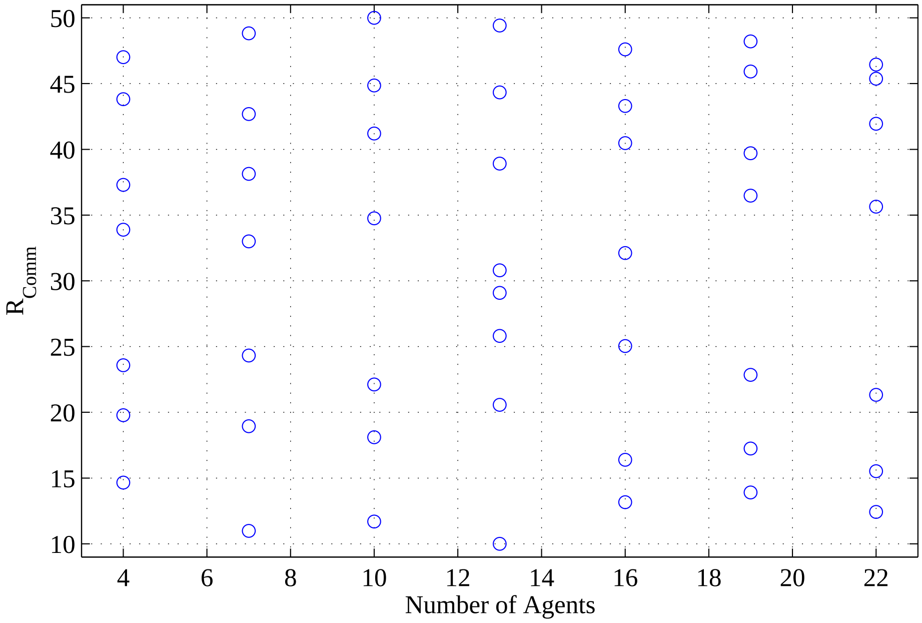

**Figure 62:** Space filling design for 2 variables, 50 runs

Summary statistics for the execution of this design are shown in Table 21.

**Table 21:** Summary statistics for 50 case space filling design

| **Parameter** | **Value** | **Units** |
|---|---|---|
| Number of Cases | 50 | |
| Number of Replications | 100 | |
| Total Run Time for Experiment | 08:12:00:19 | DD:HH:MM:SS |
| Average Run Time Per Replication | 146.88 | seconds |

The contour plot of the design space divergence measure for the 50 case space filling design is shown in Figure 63. This figure shows some undesirable behavior due to large gaps and uneven spacing in the sampling.

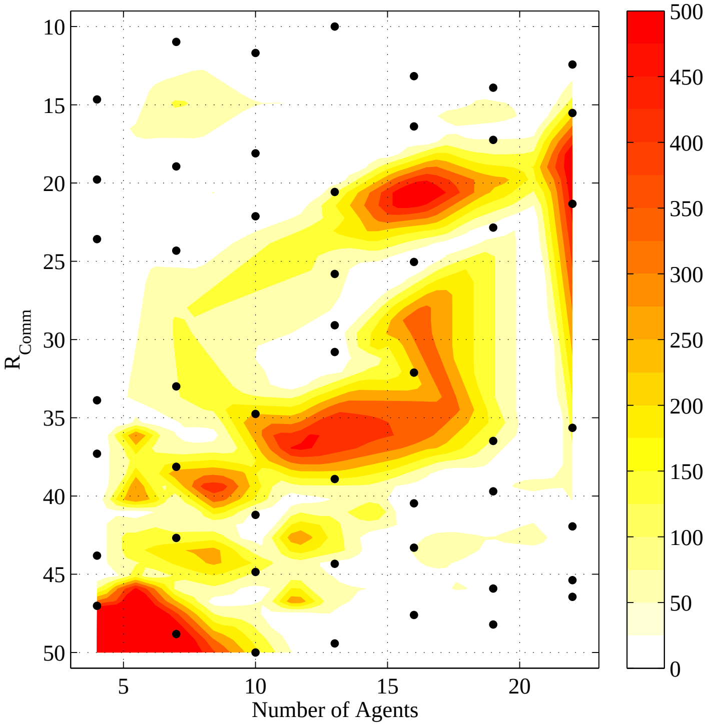

**Figure 63:** Contour plot of design space divergence measure as a function of design variables for 50 case space filling design

The large uneven spaces and large gaps between points make identifying neighbors difficult. A typical neighborhood of points for this design is shown in Figure 64. This distribution of points in the neighborhood shows that some directions do not have near neighbors and the design space distance between points is not consistent for this design.

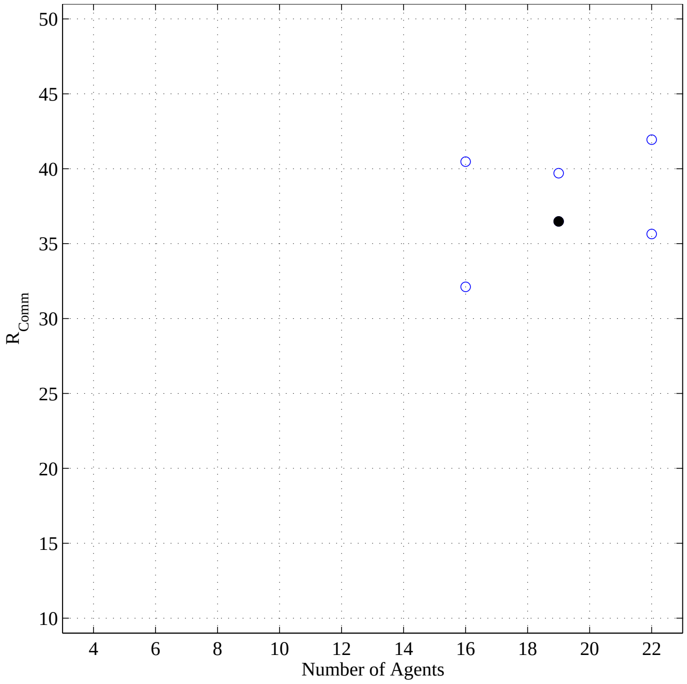

**Figure 64:** Representative nearest neighbor cluster for 50 case space filling design

### 8.1.6 Experiment Conclusions

The space filling design using 75 points provided the best result when compared to the full factorial design. The 75 point space filling design had excellent discriminability and captured the desired trends. However, the 100 point space filling design resulted in a lot of irregularities due to the clustering of neighboring points. On the other hand, the 50 point space filling design had large gaps in the design space and identified structure that was not present in the full factorial design. It appears that the most successful designs have points that are evenly distributed in the design point's neighborhood.

**Conclusion**: If the design points are distributed across the design space in a regular pattern, both full factorial and space filling designs will both work well within this methodology. The hypothesis is partially substantiated: although both full factorial and space filling are capable of achieving good results, both sampling density and the spacing within the neighborhood are important. Sampling density is not sufficient since uniformity is also required for good discriminability.

## *8.2 Experiment B: Statistical Distance Measures*

**Research Question** for **Experiment B**.

- **(RQ3.4)**: What is the appropriate way to represent and evaluate probability distributions?
  - (**RQ3.4.2**): What is the appropriate statistical distance measure to use when evaluating probability distribution similarity?

The purpose of this experiment is to characterize the statistical distance measures in order to provide guidance for the appropriateness of each measure in the overall methodology. The statistical distance for two distributions with varying means and variances will be examined. The Kullback-Leibler divergence (KL), symmetric Kullback-Leibler divergence ($\mathrm{KL}_2$), Jensen-Shannon divergence (JS), Hellinger distance (Hel), and Bhattacharyya distance (BD) will be examined using both a canonical example (a parameterized normal distribution) and using the distributions generated from the complex system under investigation in this research (the Distributed Multi-Agent Surveillance Simulation).

The hypothesis for research question **RQ3.4.2** is akin to a null hypothesis where there is no significant difference between the statistical distance measures. The goal of this experiment is therefore to test whether a significant difference exists or not.

**Hypothesis 3** (**HYP3**): There will be no qualitative difference in discriminability between statistical distance measures.

### 8.2.1 Mean Shift

In the first portion of this experiment, the effect of a mean shift on the statistical distances will be measured. A standard normal distribution $\mathcal{N}(0,1)$ is assigned to the distribution for $P$ while $Q$ will be a normal distribution of increasing mean with a unity standard deviation, $\mathcal{N}(Q_\mu, 1)$. This is shown in Figure 65.

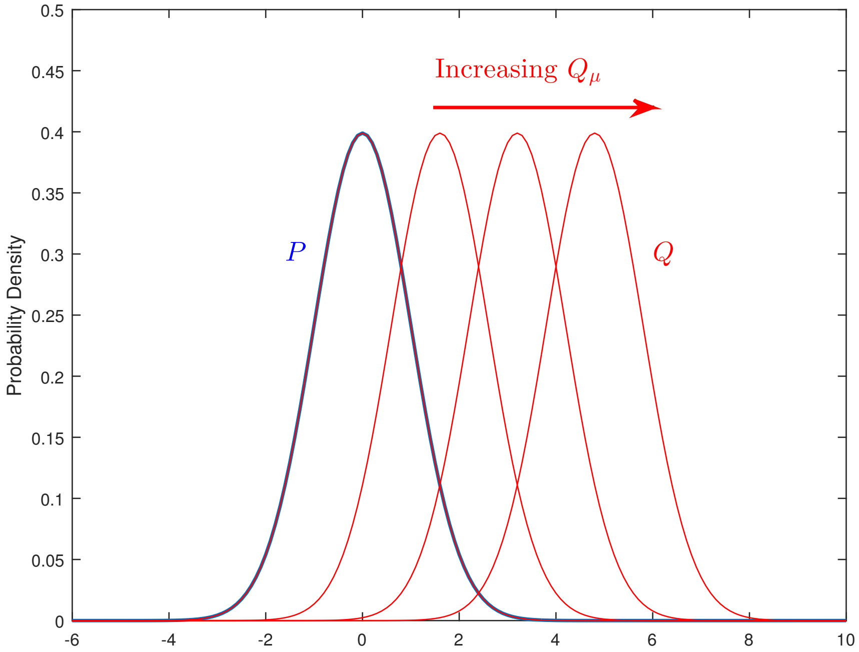

**Figure 65:** Distribution mean shift

The Kullback-Leibler divergence (KL), symmetric Kullback-Leibler divergence (KL$_2$), Jensen-Shannon divergence (JS), Hellinger distance (Hel), and Bhattacharyya distance (BD) are measured as $Q_\mu$ is increased while $P = \mathcal{N}(0,1)$ is held fixed. The plot of these measures as a function of $Q_\mu$ is shown in Figure 66. Since the two distributions are identical except for the mean, both KL and KL$_2$ are equivalent and grow quickly and unbounded in magnitude. BD is also not upper-bound but grows more slowly than either of the KL measures. Jensen-Shannon and Hellinger distance are both bound; however, it can be seen that these two measures offer very little discriminability for large differences between distributions.

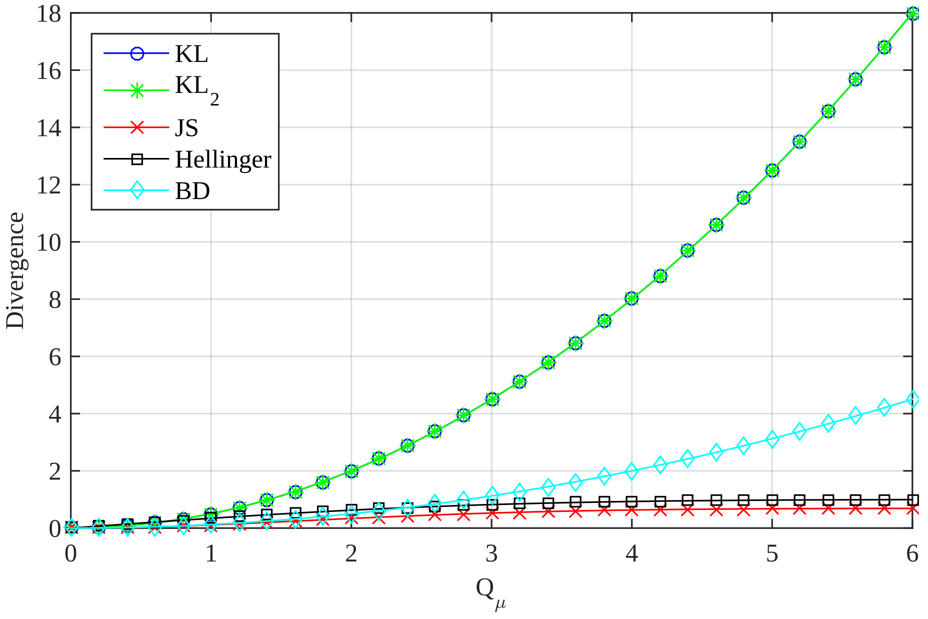

**Figure 66:** Divergence measures versus mean shift

Figure 67 shows a detail from Figure 66 for small values of mean shift. Although Hellinger distance (Hel) has poor discriminability for large mean shifts, it provides better discriminability for small mean shifts compared to the other distance measures. If the data set is comprised of mostly similar distributions, Hellinger distance would be well suited for detecting smaller shifts in the distributions while BD and JS offer very little discriminability for small differences.

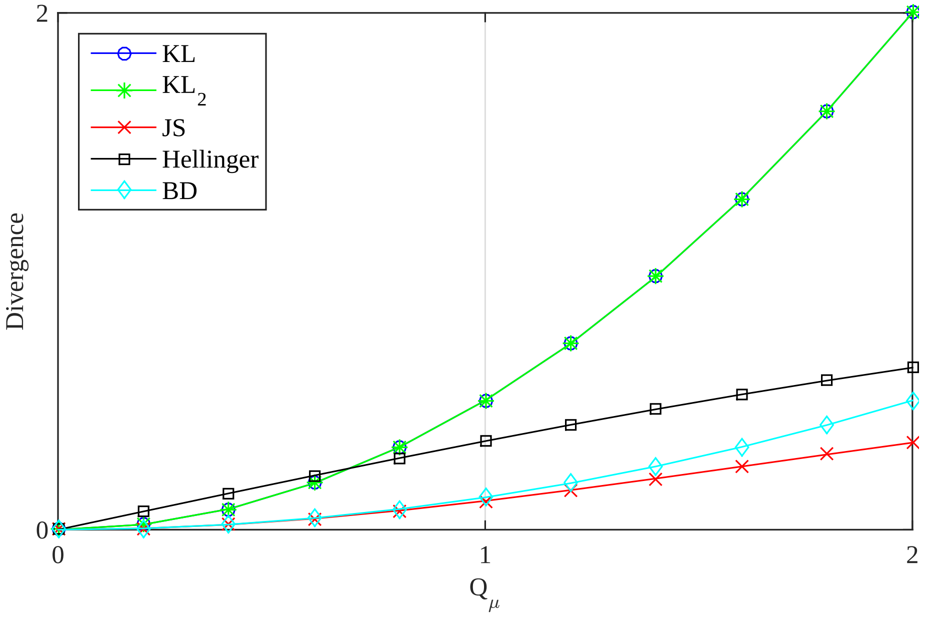

**Figure 67:** Divergence measures versus mean shift (detail)

### 8.2.2 Variance Shift

In this second part of this experiment, the effect of shifts in variance on the distance measures will be investigated. A standard normal distribution $\mathcal{N}(0, 1)$ is assigned to the distribution for $P$ while $Q$ will be a normal distribution of varying variance with a mean of zero, $\mathcal{N}(0, Q_\sigma)$. This is shown in Figure 68.

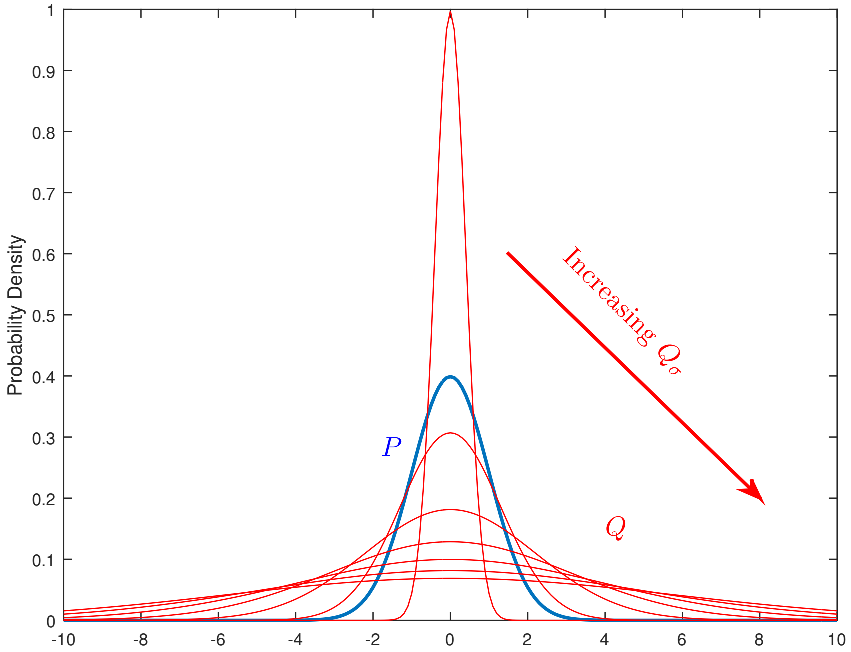

**Figure 68:** Distribution variance shift

The Kullback-Leibler divergence (KL), symmetric Kullback-Leibler divergence ($KL_2$), Jensen-Shannon divergence (JS), Hellinger distance (Hel), and Bhattacharyya distance (BD) are measured as $Q_\sigma$ is varied while $P = \mathcal{N}(0,1)$ is held fixed. The plot of these measures as a function of $Q_\sigma$ is shown in Figure 69. As expected, all of the measures approach zero as the two distributions become identical at $Q_\sigma = 0$. KL and $KL_2$ are both unbound in magnitude.

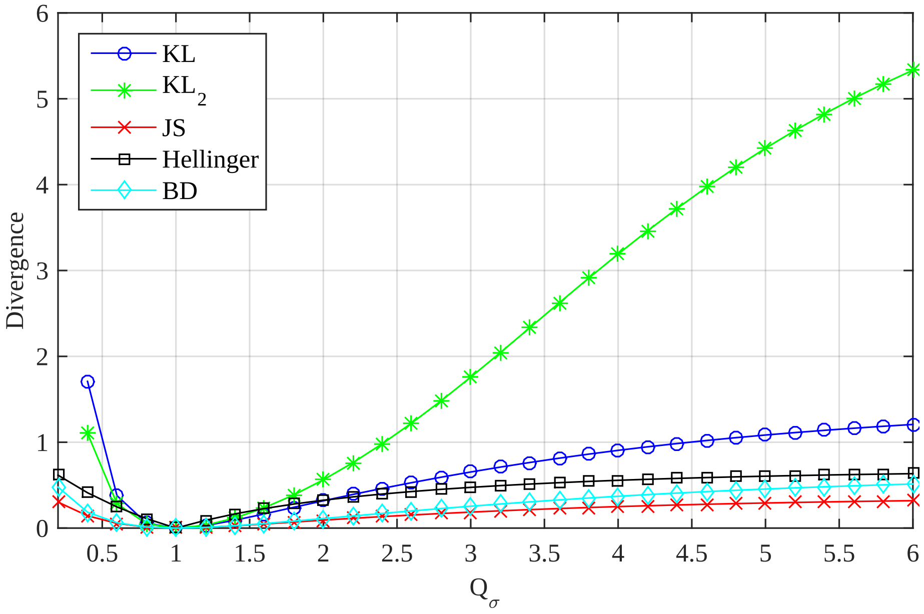

**Figure 69:** Divergence measures versus variance shift

Zooming into the region of small differences between distributions is shown in Figure 70. Jensen-Shannon and BD distances offer very little discriminability for small differences between distributions. Hellinger provides a better distance measure for detecting small differences between distributions since it has a higher sensitivity.

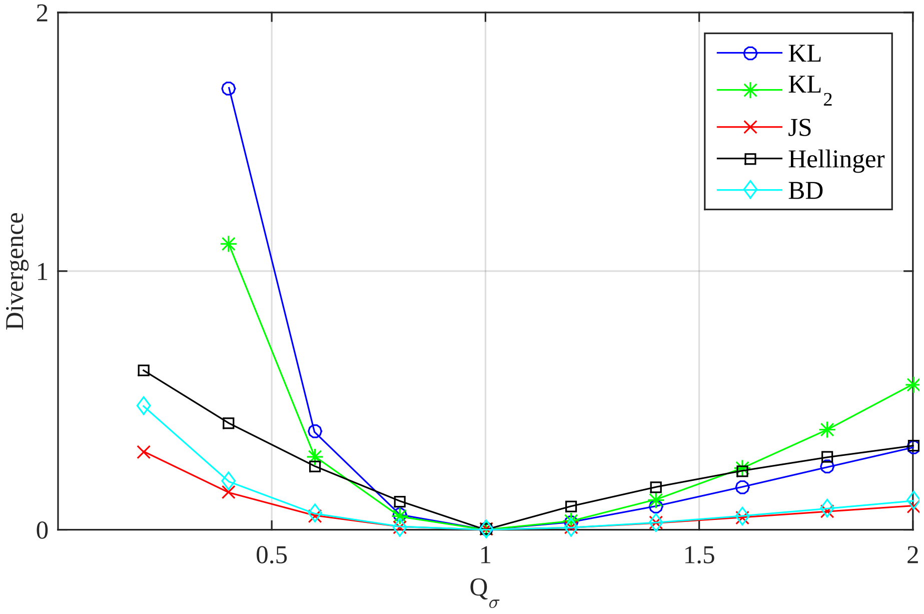

**Figure 70:** Divergence measures versus variance shift (detail)

### 8.2.3 Design Space Divergence

The final step of this experiment is to evaluate each of the divergence measures within the context of measuring the design space divergence of the test problem. The design space divergence is measured for the 2D Distributed Multi-Agent Surveillance System problem. The symmetric Kullback-Leibler divergence ($\mathrm{KL}_2$), Jensen-Shannon divergence (JS), Hellinger distance (Hel), and Bhattacharyya distance (BD) is shown in Figure 71.

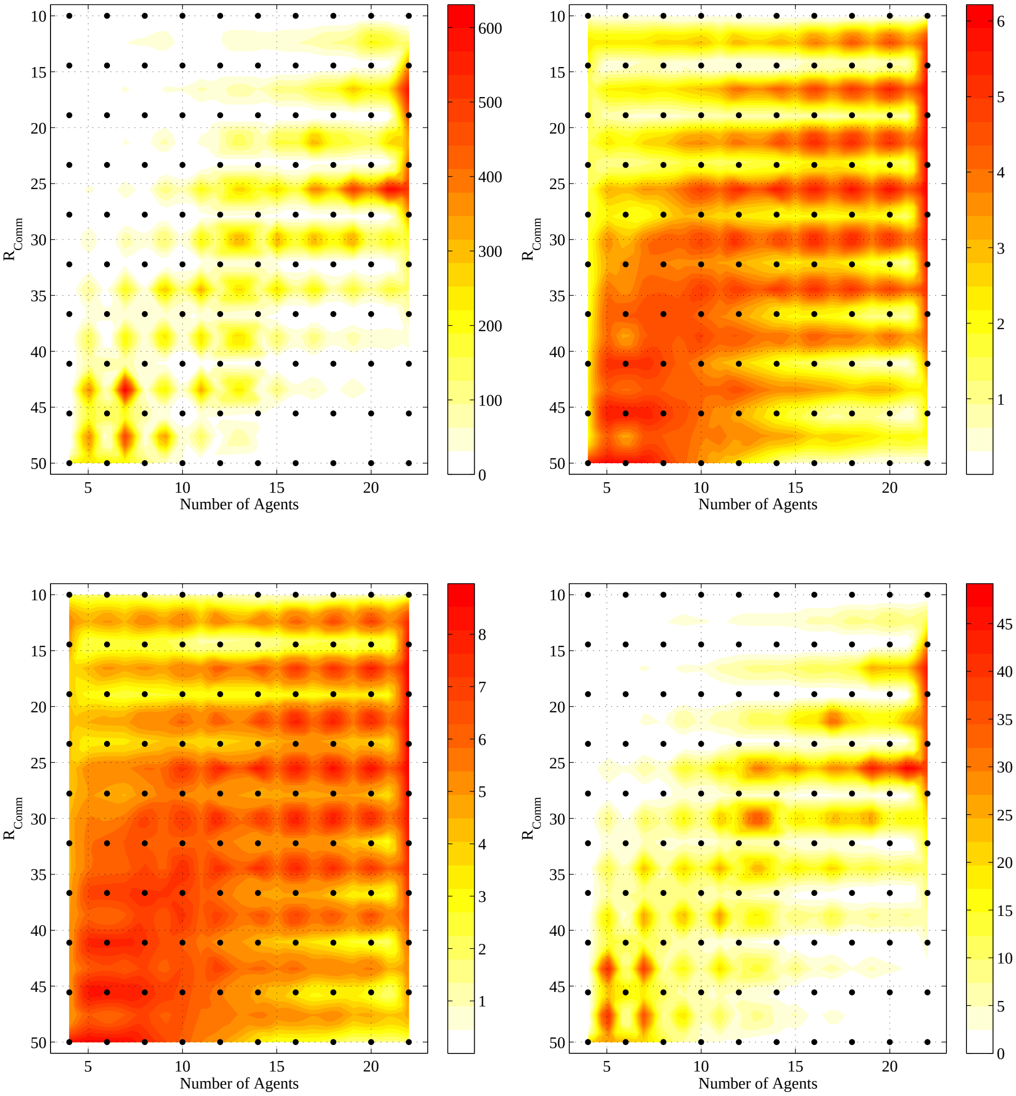

**Figure 71:** Design space divergence measure using various divergence measures: symmetric Kullback-Leibler divergence ($KL_2$) (top left), Jensen-Shannon divergence (JS) (top right), Hellinger distance (Hel) (bottom left), and Bhattacharyya distance (BD) (bottom right)

### 8.2.4 Experiment Conclusions

**Conclusion**: There does appear to be a qualitative difference in discriminability between the statistical distances measures; therefore, **HYP3** is not substantiated and has been falsified.

It is clear that there are two qualitatively different results. Symmetric Kullback-Leibler divergence (KL$_2$) and Bhattacharyya distance (BD) are qualitatively similar while Jensen-Shannon divergence (JS) and Hellinger distance (Hel) form another qualitatively similar pair. The commonality in measures within each pair is whether or not the measure is bounded. KL$_2$ and Bhattacharyya distance BD are unbounded and are able to discriminate among strongly dissimilar probability distributions. On the other hand, JS and Hel measures are bounded and tend to get saturated for design spaces with strongly dissimilar distributions. This suggests that the appropriate statistical distance to use is context dependent. If the design space is characterized by relatively small shifts in mean and variance, a more sensitive measure, especially Hel, would be more appropriate. From the results of the parametric normal portion of the experiment, Hel generated the largest measure among all of the ones tested for very small shifts in mean and variance. On the other hand, if the design space is characterized by strongly varying distributions (as is the case in the D-MASS problem), KL$_2$ or BD would be appropriate measures to use.

## *8.3 Experiment C: Parametric vs. Nonparametric Probability Distributions*

**Research Question** for **Experiment C**.

- **(RQ3.4)**: What is the appropriate way to represent probability distributions?
  - (**RQ3.4.1**): Should probability distributions be modeled as parametric or nonparametric distributions?

It has been argued earlier in this dissertation that a nonparametric approach may be more appropriate based on grounds related to the philosophical approach of doing statistical analysis. However, the purpose of this experiment is to demonstrate scenarios where a nonparametric approach is necessary because a parametric approach is severely flawed.

**Hypothesis 4** (**HYP4**): In situations where probability distributions are constrained or when a large degree of flexibility is needed, a nonparametric probability distribution, specifically the data-driven kernel method, provides a better model for capturing probabilistic features.

The probability distributions observed will depend on the underlying process and is therefore context dependent. In this dissertation, the problem being investigated is the Distributed Multi-Agent Surveillance Simulation. Using the results from this simulation, the distributions from two design points where the a priori assumption of a parametric distribution is problematic are shown in Figure 72. Although a cursory review of the probability distributions shows that many design points could conceivably be modeled as normal distributions, there are clearly design points where the assumption of normality is not valid. Figure 72 shows several examples of distributions in which an assumption of normality is not a good one. In the left figure, the distribution is not symmetric due to a constraint: reaching a limiting value of maximum efficiency. Assuming a normal distribution and using the observed mean and variance would imply values that are above the maximum value possible. The subfigure on the right shows a distribution that exhibits skewness. Although it is possible that more samples (than the current 150) would yield a more symmetric distribution, at this time, there is not enough evidence that a normal distribution is a good model for this data. A nonparametric approach ensures that any inferences made are consistent with the data. Taking inspiration from Jaynes, the goal is to be "maximally noncommittal with regards to missing information" [152].

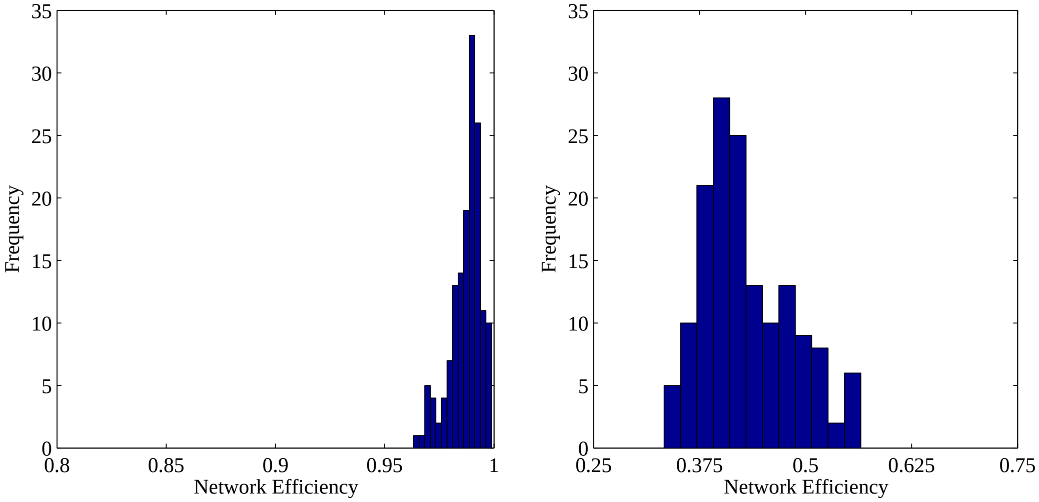

**Figure 72:** Two design point distributions from Distributed Multi-Agent Surveillance Simulation exhibiting non-normal distributions

In order to make this investigation more rigorous, a systematic test of normality at each design point was carried out. Four tests were performed at three levels of significance. The Jarque-Bera, Lilliefors, Anderson-Darling, and Chi-Square Goodness-of-Fit can all be used to test whether a data sample comes from a normal distribution whose parameters are not specified and must be determined using the sample estimates. Each of these tests uses a null hypothesis that the sample comes from a normal distribution; the hypothesis is either accepted or rejected based on a specified significance level. These tests were performed at a significance level of 0.1%, 1%, and 5%. The results, shown in Table 22, indicate that a large percentage of design points where the null hypothesis was rejected, indicating that the data did not come from a normal distribution.

It should be obvious that distributions other than normal could also have been tested. Rather, these results are intended to show that, in general, parametric distributions lack flexibility. Even looking at the most optimistic interpretation of this portion of the experiment, about 20% of the design space cannot be treated as Gaussian even if the remaining portion can be.

**Table 22:** Tests for normality of design points in Distributed-MASS design space

| **Test** | **Percent Non-Normal** ($p = 0.001$) | **Percent Non-Normal** ($p = 0.01$) | **Percent Non-Normal** ($p = 0.05$) |
|---|---|---|---|
| Jarque-Bera | 16.13% | 29.01% | 46.06% |
| Lilliefors | 23.07% | 35.34% | 51.62% |
| Anderson-Darling | 28.09% | 40.97% | 54.17% |
| Chi-Square Goodness-of-Fit | 13.81% | 29.01% | 44.98% |

The histogram, kernel density estimate, and the estimate assuming a normal distribution is shown in Figure 73. It is clear that assuming a normal distribution results in a poor qualitative fit of the data. Although there are other parametric distributions which might have provided for a better fit than the normal distribution, there is little a priori justification for any particular distribution for a given design point. This figure show that the kernel density estimate qualitatively captures the shape of the distribution and provides a good estimate of the underlying distribution.

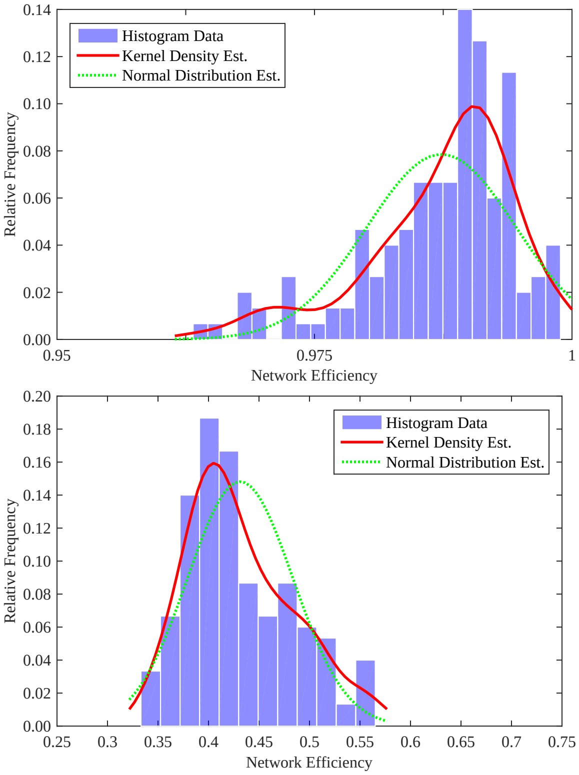

**Figure 73:** Two design point distributions from Distributed Multi-Agent Surveillance Simulation exhibiting non-normal distributions including kernel density estimates and normal distribution estimates

**Conclusion**: The results show that nonparametric approaches are more appropriate when maximum flexibility is needed or trying to model distributions that are constrained. **HYP4** is substantiated and is not falsified.

## *8.4 Experiment D: Evaluating the Methodology Performance*

**Research Question** for **Experiment D:**

- **(RQ3)** What is the appropriate methodology for engineering complex systems?

The goal for **Experiment D** is to test the overall methodology under the umbrella of **RQ3**, which is the research question that drove the development of the methodology. The natural hypothesis for this experiment is that the methodology does what it was intended to do: identify design points that are candidates of emergent behavior.

**Hypothesis 5 (HYP5)**: The proposed methodology is capable of identifying conditions for candidate emergent behavior in a complex system model.

This experiment results will be shown and discussed in the Case Study (Chapter 9).

**Conclusion**: The results show that the proposed methodology is capable of identifying a small handful of candidate points that correspond to critical transitions in system behavior and are interesting candidates of emergent behavior.

# CHAPTER IX

# CASE STUDY: IMPLEMENTING SEED METHODOLOGY

**Chapter Road Map**: The goal of this chapter is to demonstrate the methodology developed in this thesis toward an engineering design problem.

In this chapter, the design space for the Distributed Multi-Agent Surveillance Simulation, initially described in Chapter 6.3, will be explored using the SEED methodology.

## *9.1 SEED Methodology – Phase 1: Data Collection*

### 9.1.1 Selection of Design Variables and Ranges

To make the investigation tractable, the design space will be limited to just four design variables. Like the initial design space exploration developed in Chapter 6, the number of agents and the communication range will be examined. These two design variables represent one of the most fundamental trade studies to be addressed when designing a multi-agent system: the inherent tradeoff between the number of agents and the individual capabilities of each agent. These design variables help to answer the question whether it is better to have a large number of less-capable agents or a smaller number of more-capable agents.

In this case study, the communication range is varied from 10-50 units. Since the environment is assumed to be a 100 × 100 unit square, the upper end of the communication range allows a single agent to cover almost all of the environment (i.e., a circle of 100 diameter inscribed in a square with an edge length of 100). This is kept as the upper limit since a defining characteristic of distributed multi-agent systems with *cooperative* behavior is that an individual agent is not capable of accomplishing the system task individually [164]. As in the earlier investigation, the each agent's detection range is assumed to be half of the communication range. This is done to help reduce the dimensionality of the design problem by making the communication range a surrogate for the entire sensor suite capability. This enables consideration of subsystem capabilities at the system level with a fidelity that is more appropriate for conceptual design.

It is difficult to define the required number of agents to constitute a swarm since the number would be highly dependent on the application and the coordination mechanisms used. Sahin [221] argues that 10-20 agents is a sufficient number to constitute a swarm. The Office of Naval Research LOCUST project envisions 30 UAVs in a swarm [249]. Similarly, the Naval Postgraduate School's Advanced Robotic Systems Engineering Laboratory (ARSENL) swarm is envisaged with up to 50 UAVs [89]. For this case study, the number of agents is varied from 6 to 26. The upper end of this range is representative of the size of problems of interest for multi-UAV problems. From a practical point of view, this number cannot be made arbitrarily large due to the computational burden placed on the simulation for large number of agents. Once the system is physically implemented, the agents can take advantage of the reduced computational requirements made possible by the distributed architecture; however, our simulation of this system still requires the simulation of each agent and the resulting network which requires more computational resources as the system grows in size.

Aksaray, in her investigation of the multi-UAV persistent surveillance problem, found that the agent's velocity and communication radius were two of the most important variables determining the system performance [1, 2]. Like the communication range, the maximum velocity is a measure of the capability of the agent's capability and performance. While the communication range is a measure of the agent's sensor capability, the maximum velocity is a measure of the performance of the propulsion subsystem and airframe aerodynamics. Again, these variables serve as surrogates for capturing the various trades between subdisciplines during the design process. In Aksaray's investigation, the targets to be tracked were a fixed number of stationary waypoints. In the Distributed-MASS problem, the targets are mobile with their own maximum velocity. In order to reduce the dimensionality of the problem, instead of considering the maximum velocities of the agents and targets independently, these two quantities will be linked together using a new variable called the Agent-Target Velocity Factor. The baseline velocity for the agents and targets is set to 1 and is modified by the Agent-Target Velocity Factor. The mapping between Agent-Target Velocity Ratio and the settings in the simulation for the target and agent maximum velocities are shown below in equations 28 and 29, respectively.

$$\text{Target}\, V_{max} = \begin{cases} 1 + |\text{Agent Target Velocity Factor}| & \text{if Agent Target Velocity Factor} < 0 \\ 1 & \text{if Agent Target Velocity Factor} > 0 \end{cases} \tag{28}$$

$$\text{Agent}\, V_{max} = \begin{cases} 1 & \text{if Agent Target Velocity Factor} < 0 \\ 1 + |\text{Agent Target Velocity Factor}| & \text{if Agent Target Velocity Factor} > 0 \end{cases} \tag{29}$$

In addition to reducing the dimensionality of the problem, this mapping also captures the more salient aspect of the problem: the *relative* velocity between the agents and the targets. This approach is similar to those taken in predator-prey simulations [3], where the relative velocity is one of the most important factors for determining the outcome of the simulation. However, unlike the predator-prey model where the predator has to physically intercept the prey, agents in the Distributed-MASS problem are able to track the targets within a limited distance. This design variable enables the investigation of performance requirements: how fast must an individual agent be in order for the system to be effective? Recalling the motivation of distributed multi-agent systems, the goal is cooperative behavior; therefore, it will be interesting to investigate if the information sharing between connected agents results in cooperative behavior where individual agent velocity becomes less relevant because agents are able to hand-off tracking duties to other nearby agents. In this case study, the Agent-Target Velocity Factor is varied from -1 to 1. When the Agent-Target Velocity Factor is -1, the target is twice as fast as the agent; when the Agent-Target Velocity Factor is equal to 1, the agent is twice as fast as the target. The relative velocity between the agent and target varies linearly with the Agent-Target Velocity Factor variable between these two extremes.

Finally, the fourth design variable will be the number of targets to be tracked in the environment. It should be clear that the number of targets is not a design variable in the same sense as the communication range or number of agents. However, it is introduced into this problem in order to investigate the robustness of the multi-agent system. As system designers, we are not able to anticipate how many targets may exist in the environment; however, we certainly must be able to demonstrate that our system is capable of handling a wide range in number of targets with potentially more targets than agents to track them. The number of targets is varied from 4 to 14.

In summary, the design variables explored in this case study and their ranges are shown in Table 23.

**Table 23:** Design space exploration variables

| Design Variable | Min Value | Max Value |
|---|---|---|
| Number of Agents | 6 | 26 |
| Communication Range | 10 | 50 |
| Detection Range* | 5 | 25 |
| Number of Targets | 4 | 14 |
| Agent-Target Velocity Factor | -1 | 1 |

* assumed to be 1/2 of the Communication Range

### 9.1.2 Selection of Response Variables

The data analysis and exploration phases of the SEED methodology are applied on all responses of interest. The response variables of interest should be measures that are both important measures of system performance and are likely to be directly influenced by emergent behavior. As an example using the Distributed-MASS problem, the time to locate the first target may be of interest to the system analyst; however, it is unlikely that this metric would be strongly indicative of emergent behavior. This metric would be more likely to be influenced by individual agent performance and not the cooperative behavior of the system. Therefore, performing the SEED methodology using this as the measure of system effectiveness would not yield interesting results. For this case study, the measure of system effectiveness will be the Base Target Track Count Percentage (BTTCP). This measure best captures the overall system effectiveness.

### 9.1.3 Design Point Selection

The design point selection step involves the choice of sampling design. As described in the methodology chapter, Design of Experiments (DOE) techniques can be used to thoroughly sample the design space. For this case study, a full factorial design was selected. Although Experiment A showed that a space-filling design is capable of generating results comparable to the full-factorial design if the spacing is relatively uniform, a full factorial design provides a more reliable approach at the expense of more design points to evaluate. For this case study, a full-factorial design at six levels per variable was created. The resulting design is shown in Figure 74.

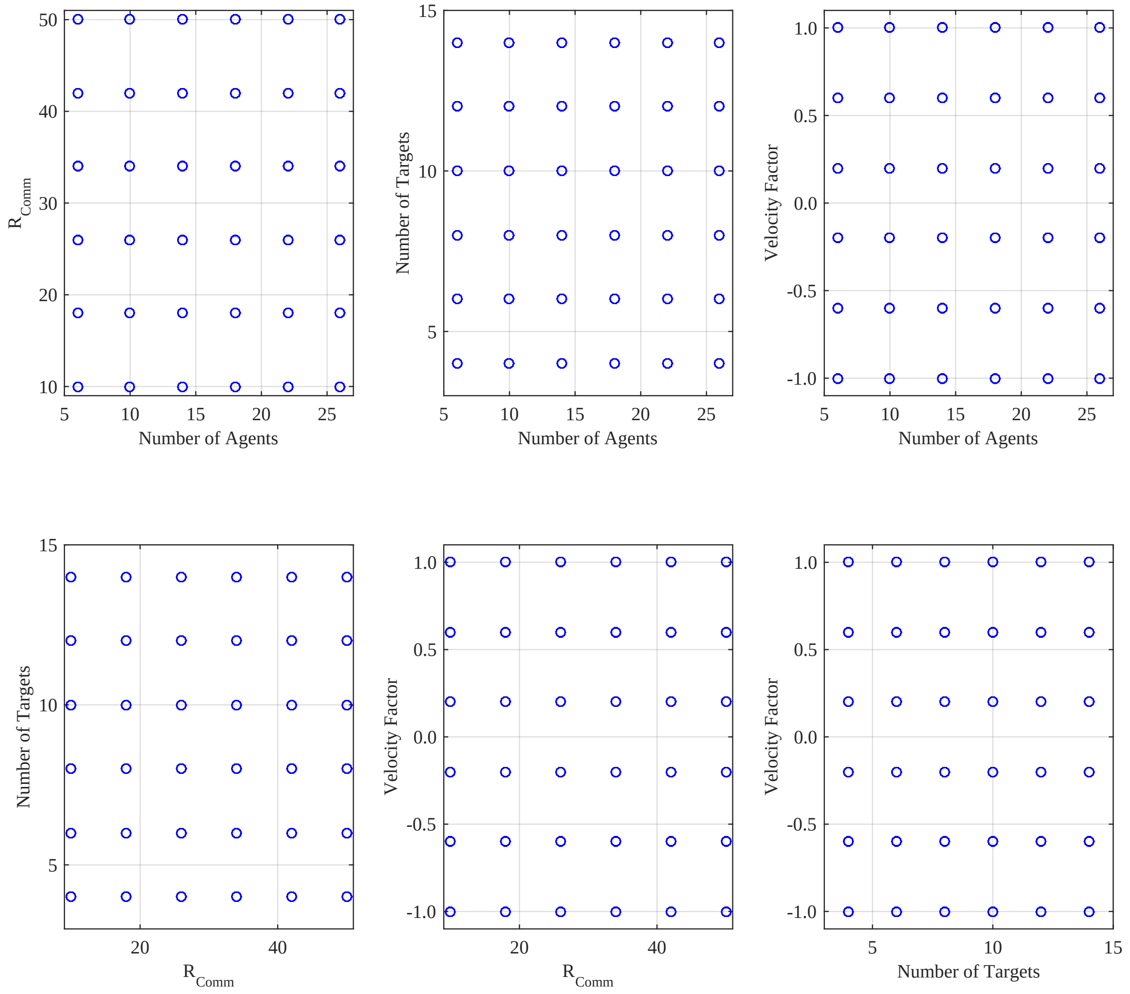

**Figure 74:** Full factorial design

Only six levels per variable were chosen for the initial evaluation due to limited computational resources. Due to the local neighborhood search of this methodology, neither an overly fine nor an overly coarse search grid is desirable. A grid that is too coarse will result in a situation where all neighboring points have a large divergence measure and it will be difficult to distinguish behaviors which are a result of critical transitions and those that are simply due to large changes in the underlying conditions. On the other hand, a search grid that is too fine also has a number of drawbacks. A very fine grid will result in a large number of design points which increase the number of required evaluations and computational burden of the methodology. Both the numerator and denominator in the Design Space Divergence measure will tend to approach zero as the grid becomes finer. However, since the numerator is estimated from a statistical sample, it is likely to be dominated by sampling noise for arbitrarily close design points.

The SEED methodology includes a high-level iteration to allow for refinement of the design space. The results from this exploration can be used to refine the search space using a smaller range to improve the resolution and fine tune the design variables.

### 9.1.4 Design Point Replication Sampling

The number of samples at each design point must be selected. The number of samples should be sufficient to adequately characterize the underlying probability distribution of the response of interest. As discussed in the methodology chapter, either a fixed number of samples or an adaptive sampling approach can be used. For simplicity and due to limited computational resources, a fixed number of samples were used in this case study. It is important to test whether the selected number of samples is sufficient to characterize the probability distribution for the purposes of this methodology. Using a representative design point, a large number of samples were carried out in order to test the convergence of the distribution. The convergence of the statistics of the representative design point are shown in Figure 75. The mean, standard deviation, and kurtosis converge quickly. The sample skewness does not converge as nicely; however, the magnitude of the sampling noise seems to be fairly constant after about 150 samples. This figure indicates that 150 samples is adequate to get an accurate estimate of the distribution statistics.

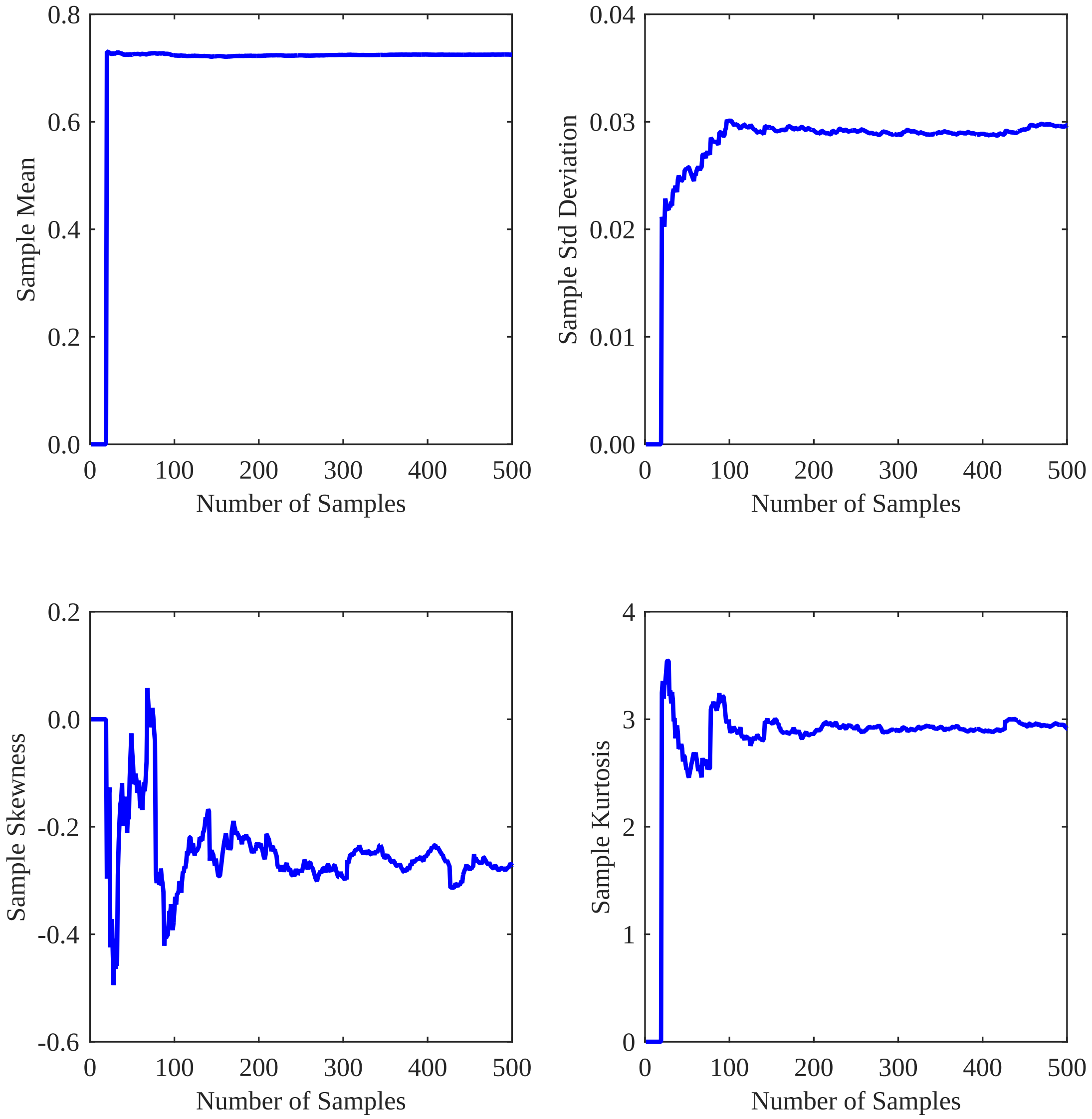

**Figure 75:** Probability distribution statistics convergence

Since the methodology is based on the use of statistical distance measures, it is more appropriate to test the convergence using a statistical distance. In Figure 76, the convergence of the $\mathrm{KL}_2$ statistical distance between the sampled distribution and a reference distribution based on 700 samples is shown. This figure shows that 150 samples is sufficient and that most of the variation due to sampling is eliminated with that many samples.

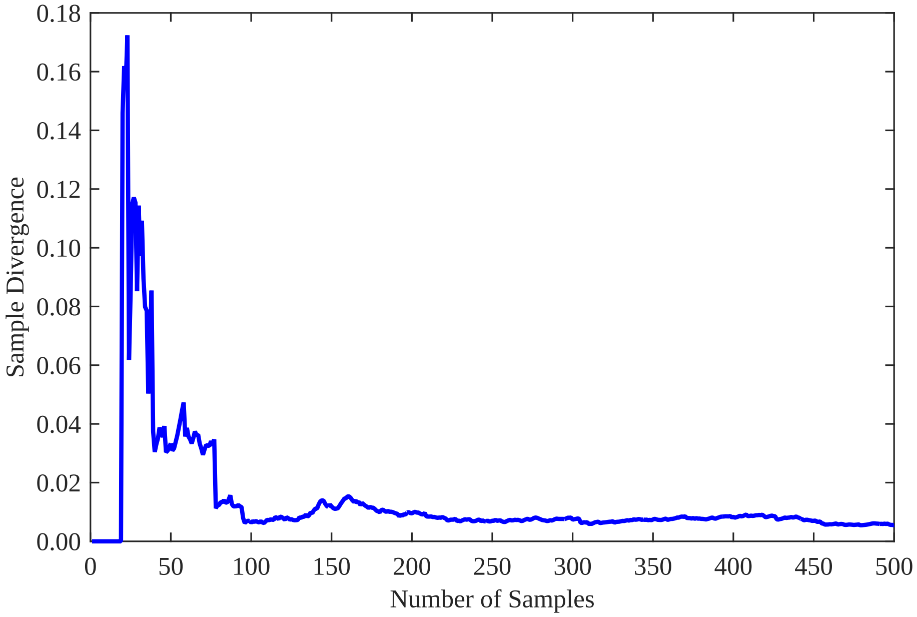

**Figure 76:** Convergence of the statistical distance with the number of samples

Of course, the number of samples to adequately characterize a distribution will be heavily dependent on the system under investigation. Therefore, using an adaptive scheme is a straight forward way of improving this methodology.

### 9.1.5 Evaluate Simulation and Build Database of Results

The full factorial design resulted in 1296 design points. Summary statistics for this design are shown in Table 24. Using 150 replications at each design point, there were 194,000 runs of the simulation. The Distributed-MASS simulation was executed at the required design points for the specified number of replications. This large number of runs was very computationally expensive requiring a cumulative run-time of almost 2 years! The ability to use parallel processing made this task feasible; however, it is clear that this is a very computationally intensive process. The results from all of the runs were compiled for further processing during the data analysis phase of the SEED methodology.

**Table 24:** Full factorial design summary

| Summary Statistic | Value |
|---|---|
| Number of design points | 1296 |
| Number of design point replications | 150 |
| Total number of simulation runs | 194,400 |
| Average simulation run time (seconds) | 322.7 |
| Total simulation run time (days) | 726.1 |

## *9.2 Characterization of the System Measures*

The system level metrics of interest, Base Target Track Count Percentage (BTTCP) and Network Efficiency, were captured for each run of the simulation. The distributions for these metrics as a function of the design variables are shown in Figures 77 and 78 for BTTCP and Network Efficiency, respectively. These plots can be interpreted as showing the *global sensitivity* of the response at each setting of the design variable. The most obvious observation is that the system exhibits a wide range of possible performance across almost the entire range of variable settings. Looking at the BTTCP metric, the general trends (i.e., mean response) of increasing performance with increasing number of agents, increasing communication range, decreasing number of targets, and increased agent velocity, make intuitive sense. At first glance, it is surprising that the number of targets and the velocity factor did not have a bigger impact on the system performance. However, this can be explained by examining the network efficiency as a function of these variables. Although the system performance does improve for fewer number of targets and faster agents, the network efficiency actually degrades. Since the agents give priority to following agents rather than maintaining network connectivity, the larger number of targets will tend to cause lower connectivity. The higher agent velocity tends to break communication links as the agents generally spend less time within the communication range of other nearby agents.

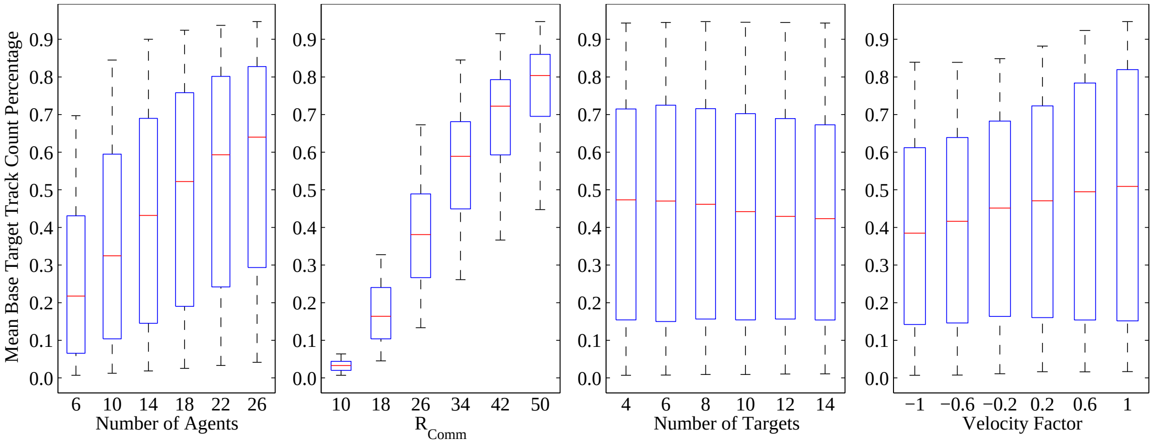

**Figure 77:** Box plots showing the variation of the mean of the Base Target Track Count Percentage (BTTCP) versus design variable parameters

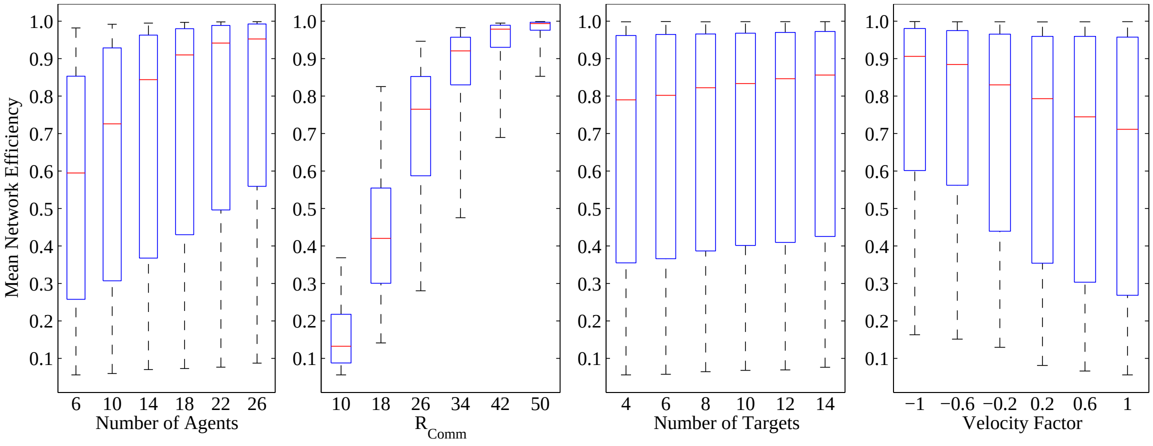

**Figure 78:** Box plots showing the variation of the mean of the Network Efficiency versus design variable parameters

One of the foundational assumptions in this thesis is that the statistical distribution—and not just the mean—is important when characterizing the behavior of the system. Although using a single statistic like standard deviation is not likely to capture all important features in a nonparametric distribution, it does provide a straightforward way of understanding how the spread of the distribution changes as a function of the design variables. The standard deviation of BTTCP and of the Network Efficiency is shown in Figure 79 and 80, respectively. Typically, smaller standard deviations are desirable since they are indicative of robust designs that are less affected by inherent stochasticity in the problem. However, one region of interest that becomes clear when examining these figures is the dramatic change in the standard deviation as a function of the communication range. It is clear that an $R_{COMM}$ value between 18-26 not only produced significant changes to the mean of both BTTCP and Network Efficiency, but also significant changes to the standard deviation. These types of shifts are reminiscent of the concept of "edge of chaos" discussed earlier, where there exists a critical transition between types of behaviors and is a possible indicator for emergent behavior.

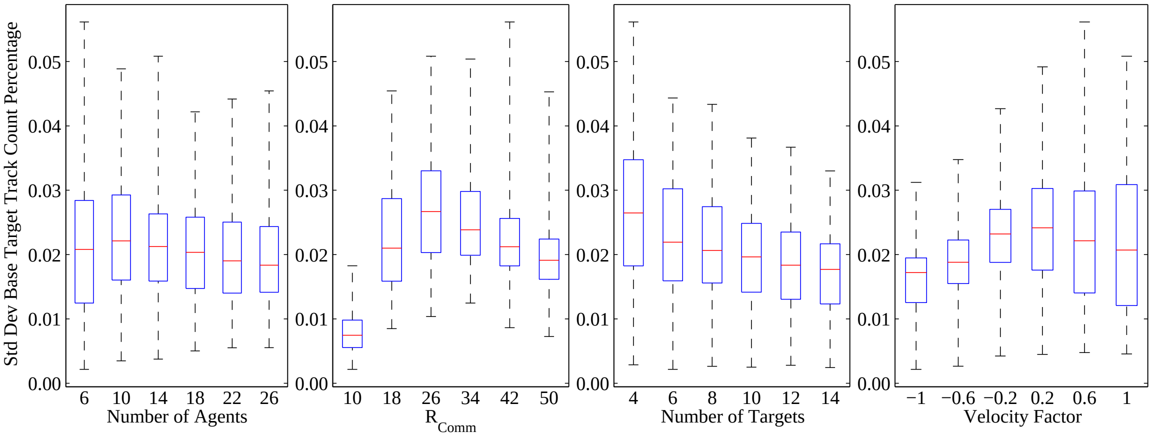

**Figure 79:** Box plots showing the variation of the standard deviation of Base Target Track Count Percentage (BTTCP) versus design variable parameters

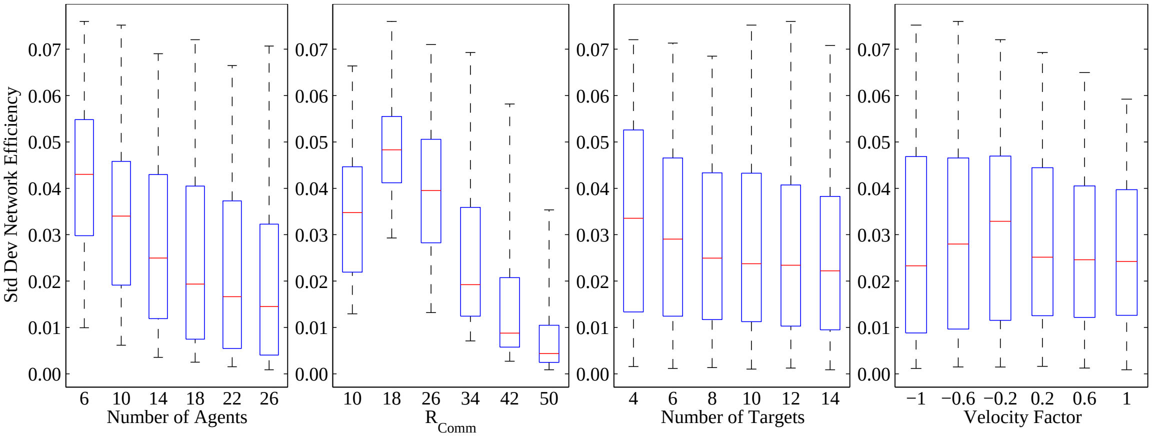

**Figure 80:** Box plots showing the variation of the standard deviation of Network Efficiency versus design variable parameters

This visualization does provide insight into the behavior of the system and general trends; however, since the data is projected onto a single dimension, higher-dimensional structure is lost and makes characterization and parameter tuning difficult. The next step toward improved visualization of the landscape of the system response is to use contour plots. Contour plots are useful in visualizing the structure of the data projected onto a two-dimensional plane. In this case study, there are four design variables which results in 6 combinations of pairs of variables. The downside of using contour plots is that they are still a projection onto a lower-dimension, which means that higher-dimensional structure is lost. Additionally, the number of combinations of pairs of variables grows quickly; in a 10 dimensional problem, there are 45 combinations.

The contour plots for every pair of variables for the variation of the mean BTTCP and network efficiency in Figure 81 and Figure 82, respectively. These contour plots show the communication range and number of agents dominate the structure of the mean response. Both BTTCP and network efficiency often show the same trends in the mean response; however, as was seen earlier, a smaller number of targets and increased agent velocity tends to improve BTTCP but simultaneously degrades the network efficiency.

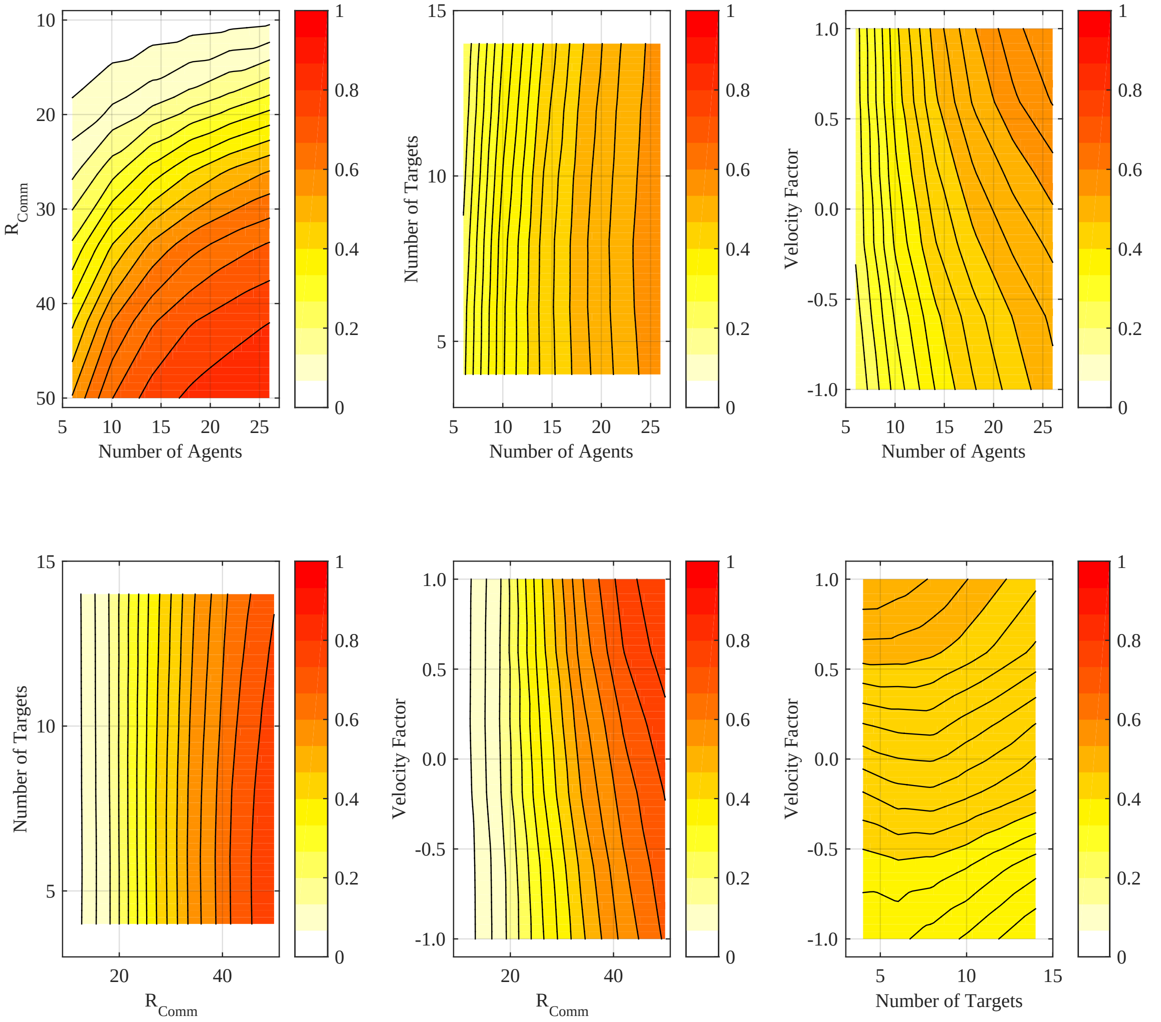

**Figure 81:** Contour plots showing the variation of the mean of Base Target Track Count Percentage (BTTCP) for design variable pairs

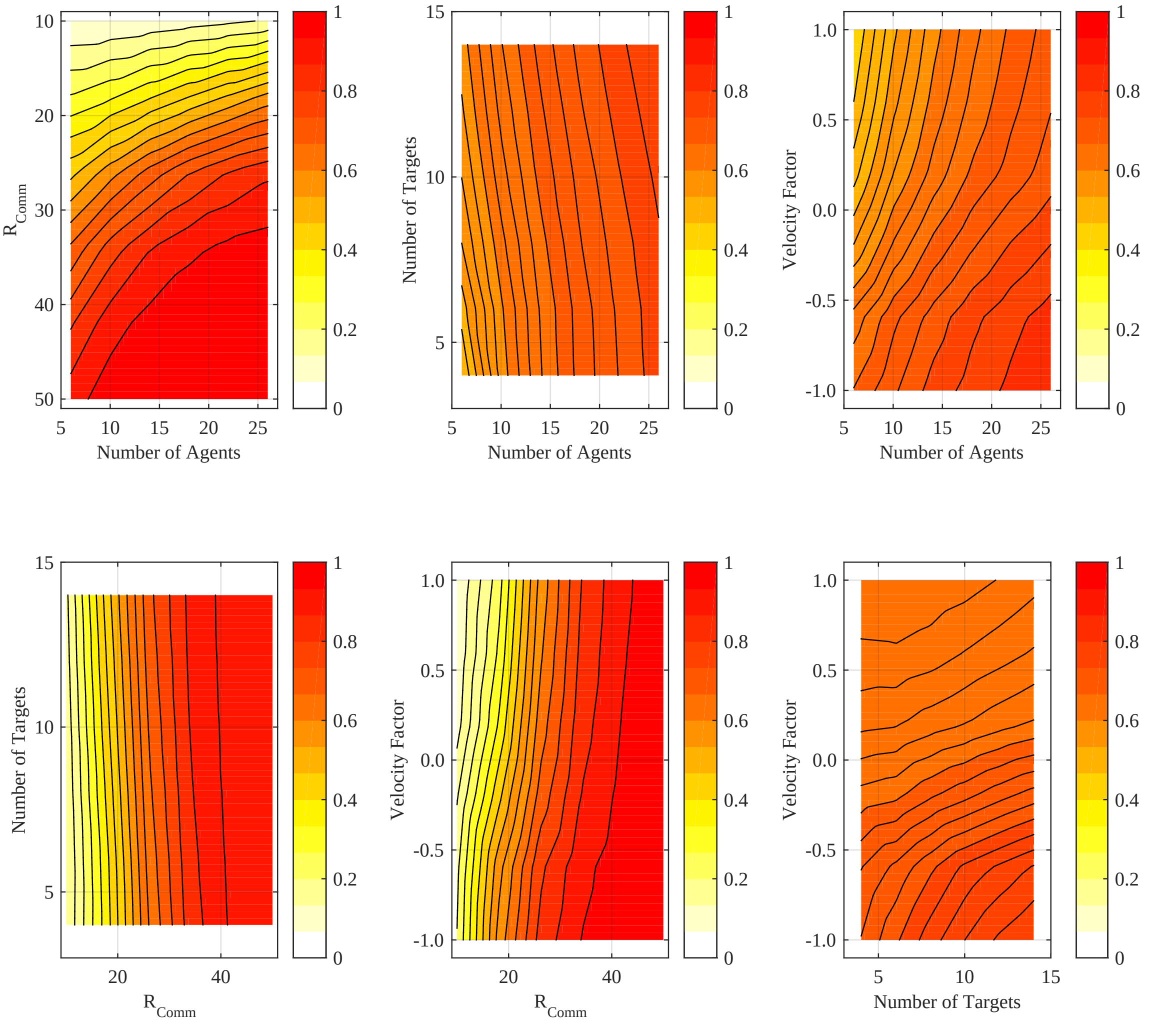

**Figure 82:** Contour plots showing the variation of the mean of network efficiency for design variable pairs

Once again, the importance of the entire probability distribution is emphasized here. Understanding how the design variables influences the standard deviation of the response variables is an important part of understanding the system behavior. The variation of the standard deviation of BTTCP and network efficiency is shown in Figure 83 and Figure 84, respectively. It is interesting to see that the structure of the standard deviation is markedly different from the trends seen in the mean response. The standard deviation has local maxima that may be indicative of shifts in behavior and critical transitions. While the number of targets and the velocity factor did not affect the mean response as much as might be expected, these variables have a significant effect on the standard deviation of the responses.

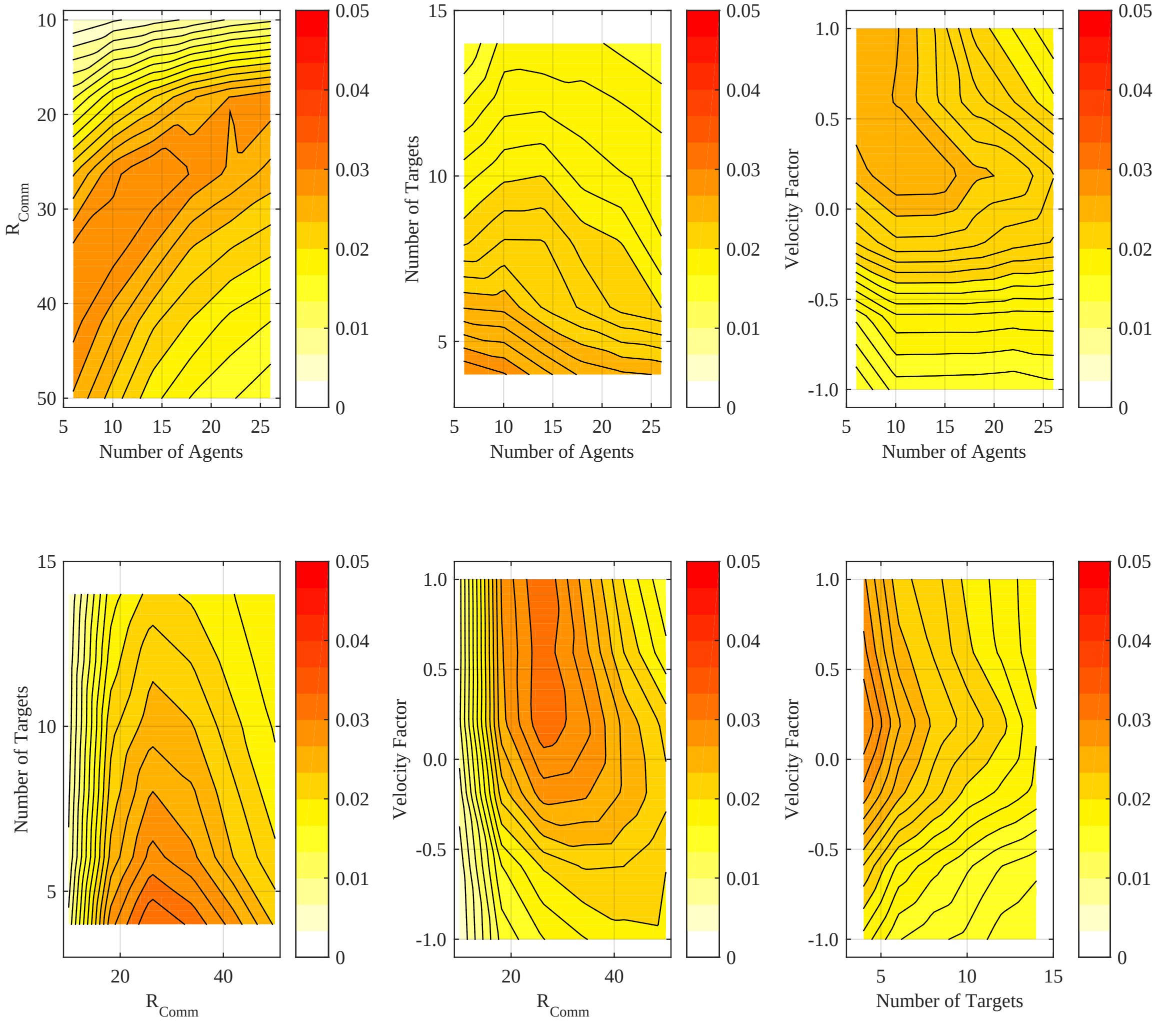

**Figure 83:** Contour plots showing the variation of the standard deviation of Base Target Track Count Percentage (BTTCP) for design variable pairs

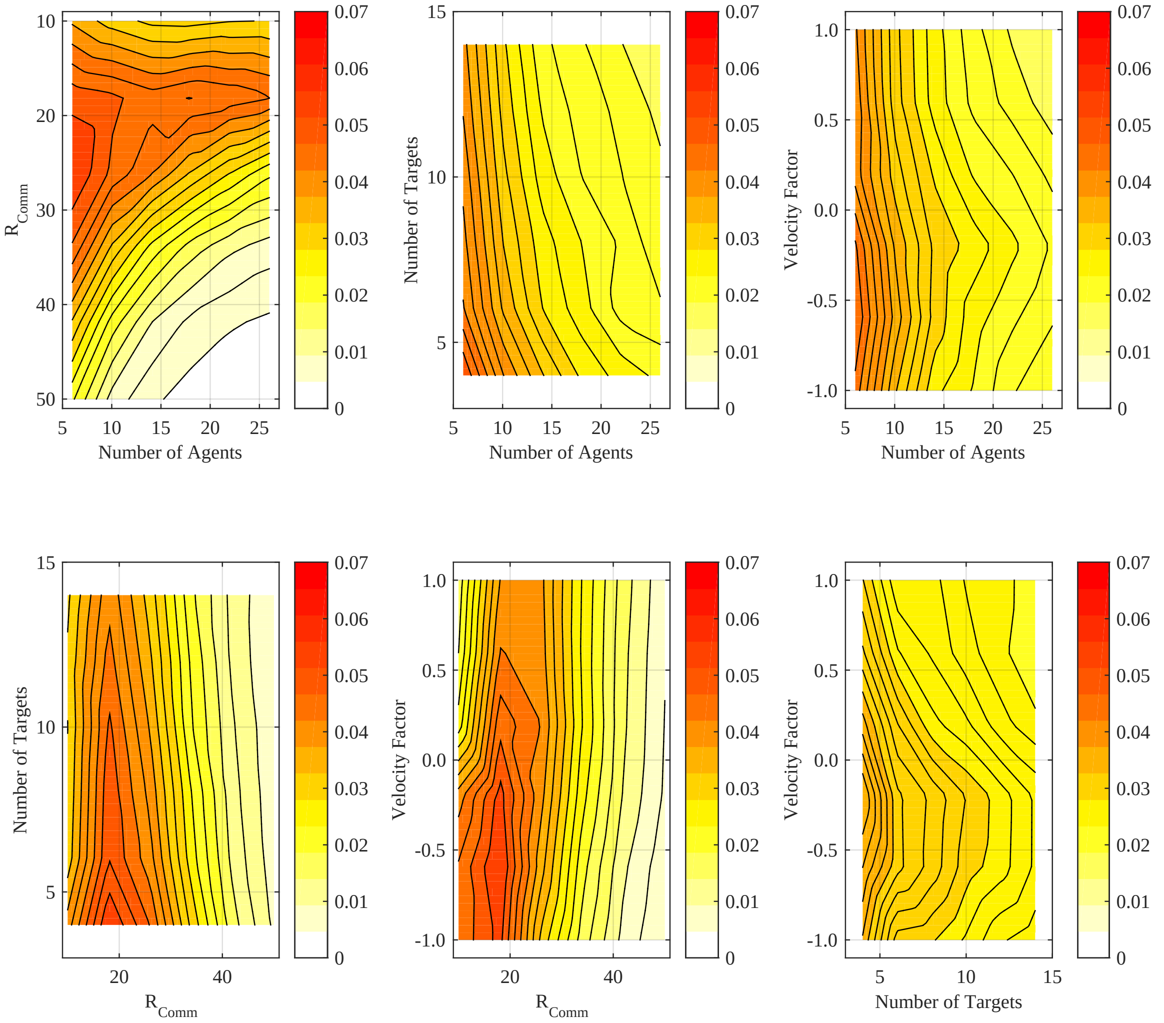

**Figure 84:** Contour plots showing the variation of the standard deviation of network efficiency for design variable pairs

## *9.3 SEED Methodology – Phase 2: Data Analysis*

The data analysis phase as implemented in the SEED methodology is largely an automated process. Once the various alternatives for each step in the process are selected for appropriateness for the problem, the data analysis phase, as implemented in Chapter 7.3, can proceed. Only several of the steps in the data analysis phase will require any further discussion for this case study.

### 9.3.1 Select Neighbor Distance Measure and Perform Neighbor Search

One of the decisions to make before performing the neighbor search is to select the distance measure to be used. As discussed in the methodology chapter, two of the most likely options are the Euclidean ($L^2$) and Chebyshev ($L^\infty$) distances. The effect of this choice will determine which nearby points are selected as neighbors. As discussed earlier, the Chebyshev distance will tend to capture "corner" points. This results in neighbors which have multiple variables changing simultaneously and therefore will include interaction effects between variables. On the other hand, the Euclidean measure will not capture the corner points and each neighbor will tend to be along a single dimension only. This has the effect of measuring one-at-a-time effects and neglects interactions. Since emergent behavior may be brought about by simultaneous changes of variables, it is suggested that the Chebyshev distance be used for selecting neighbors. The downside of selecting Chebyshev over Euclidean is the large increase in the number of neighbors selected. As the dimensionality of the problem increases, the number of neighbors will grow very quickly. As a demonstration, the total number of neighbors in the case study using the Euclidean and Chebyshev distances is shown in Table 25. Even in only 4 dimensions, allowing corner points to be selected as neighbors resulting in nearly an 8-fold increase in the total number of points to evaluate. This evaluation is relatively inexpensive; however, it is clear that for large dimensional problems, it is likely to become infeasible. Switching to using a Euclidean distance measure for neighbor selection will likely be an acceptable solution—allowing the search of a higher dimensional space while higher-order interaction effects (using the Chebyshev distance, all higher-order interactions would be identical in distance in a regular grid).

**Table 25:** Number of total neighbors for 4D full factorial design

| Distance Measure | Number of Total Neighbors |
|---|---|
| Euclidean | 8,640 |
| Chebyshev | 64,240 |

In addition to selecting the distance measure, a corresponding search distance must be defined. Any neighbor within that distance according to the distance measure selected will be considered a neighbor. In this case study, a full factorial design is being used with an equal number of levels per dimension. This creates a regular grid where all neighbors are equally spaced from each other. The use of 6 levels per variable results in a normalized distance of 0.2 along each dimension. Using a search distance of 0.2 with a Chebyshev measure results in the adjacent design points along each dimension plus all of the corner points being selected.

### 9.3.2 Density Estimation

The density estimation was performed using a univariate kernel method. As discussed in Chapter 7.2.8.5, once a kernel function provides the required amount of smoothness, the exact choice of kernel function is not critical. Therefore, a Gaussian kernel function was used due to its ubiquity in kernel estimation algorithms. The bandwidth parameter was automatically selected using Equation 20. As demonstrated in **Experiment C**, this approach produced good estimates of the sampled data.

### 9.3.3 Evaluate Design Space Divergence

Evaluating the Design Space Divergence requires the selection of two options: the statistical distance measure being used and the design space distance measure. The numerator of the DSD measure uses the selected statistical distance measure. **Experiment B** demonstrated that the [symmetric] Kullback-Leibler divergence ($\text{KL}_2$) provides good results for problem. Since the sampling design is relatively coarse, the $\text{KL}_2$ divergence is appropriate for distributions which are likely to be very different due to the larger distances between neighboring design points. The denominator of the DSD measure is the design space distance between the pair of neighbors. In this case study, the Euclidean distance is used to define the distance between two points. Even though the Chebyshev distance was used to *select* the points, the distance used to evaluate the DSD measure can be independent of the measure used to select the points. The Euclidean distance was selected due to its ubiquity and general acceptance as a distance measure.

## *9.4 SEED Methodology – Phase 3: Data Exploration*

The data exploration phase of the SEED methodology can be carried out on the data generated from Phase 2. That phase generated a very large data set (i.e., 64,240 values) to explore. Since this is a large data set, a combination of data mining techniques and visual analytics will be used to identify trends in the data.

It is clear from Figure 77 that a communication range value of 10 results in poor performance over the entire design space; therefore, any design region including this communication range value are filtered so that they are not included in the results. Since it is obvious that increasing the communication range above a value of 10, the filtering will help to focus on other portions of the design space that may yield interesting results.

The design space divergence measure is evaluated according to the methodology presented in this thesis. The 2D contour plots of the design space divergence measure for every pairwise combination of design variables is shown in Figure 85. Since the contour plots are 2D projections from the higher-dimensional space, an aggregating function is used in order to collapse the remaining dimensions. In Figure 85, the design space divergence measure is averaged with respect to the remaining dimensions for each plot.

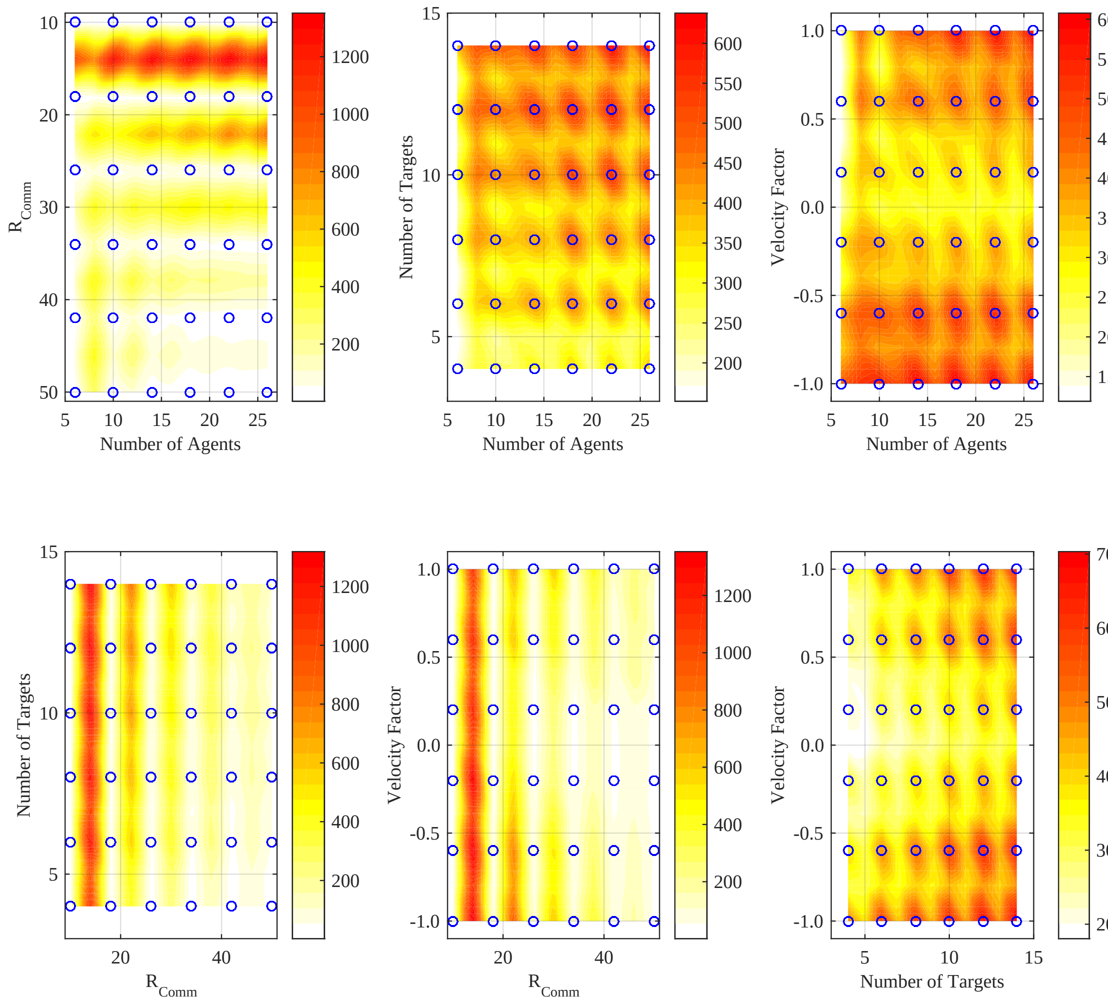

**Figure 85:** Design space divergence measure for Base Target Track Count Percentage (BTTCP) system metric of the Distributed Multi-Agent Surveillance Simulation

This collapsing of higher dimensional data onto lower-dimensional projections in order to visual the results shows one of the major flaws in the visual analytics approach to locating design points of interest. In order to assist the analyst, techniques are used to highlight high-dimensional structure that may be lost due to projections. Clustering techniques can help to identify regions of interest instead of relying on the analyst's judgment. This will also help improve the traceability of the process and provides a handful of candidate solutions for further investigation. For this case study, k-means clustering was used to help identify regions with high design space divergence measures. Silhouette plots are used to identify how many clusters should be used in the analysis and the goodness of the results. Figure 86 (left) shows the variation of the mean silhouette value as a function of the number of clusters. This figure shows that 7 clusters results in the highest mean value and will therefore be the number of clusters selected in this analysis. The silhouette plot on the right side of 86 shows the resulting silhouette plot for 7 clusters. These figures show that there is a moderate amount of structure captured using these clusters.

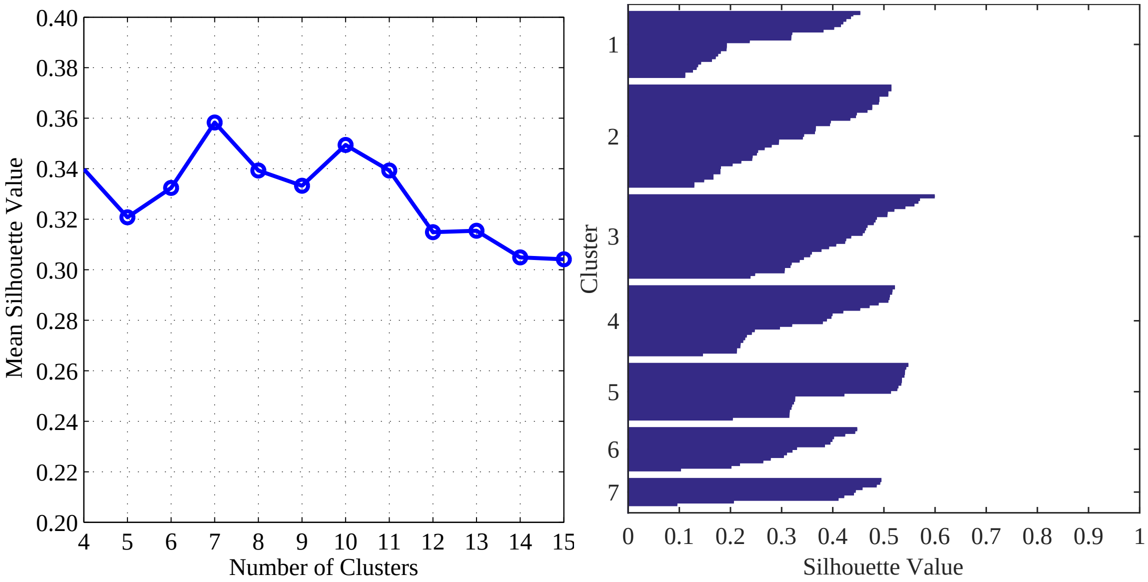

**Figure 86:** (Left) Mean of silhouette values for various number of clusters; (Right) silhouette plot for 7 clusters (maximum value from plot on left)

Figure 87 shows one resulting cluster from the preceding cluster analysis. The nearest point to the cluster centroid is shown as a red circle; this point will be used as the prototypical solution for the cluster.

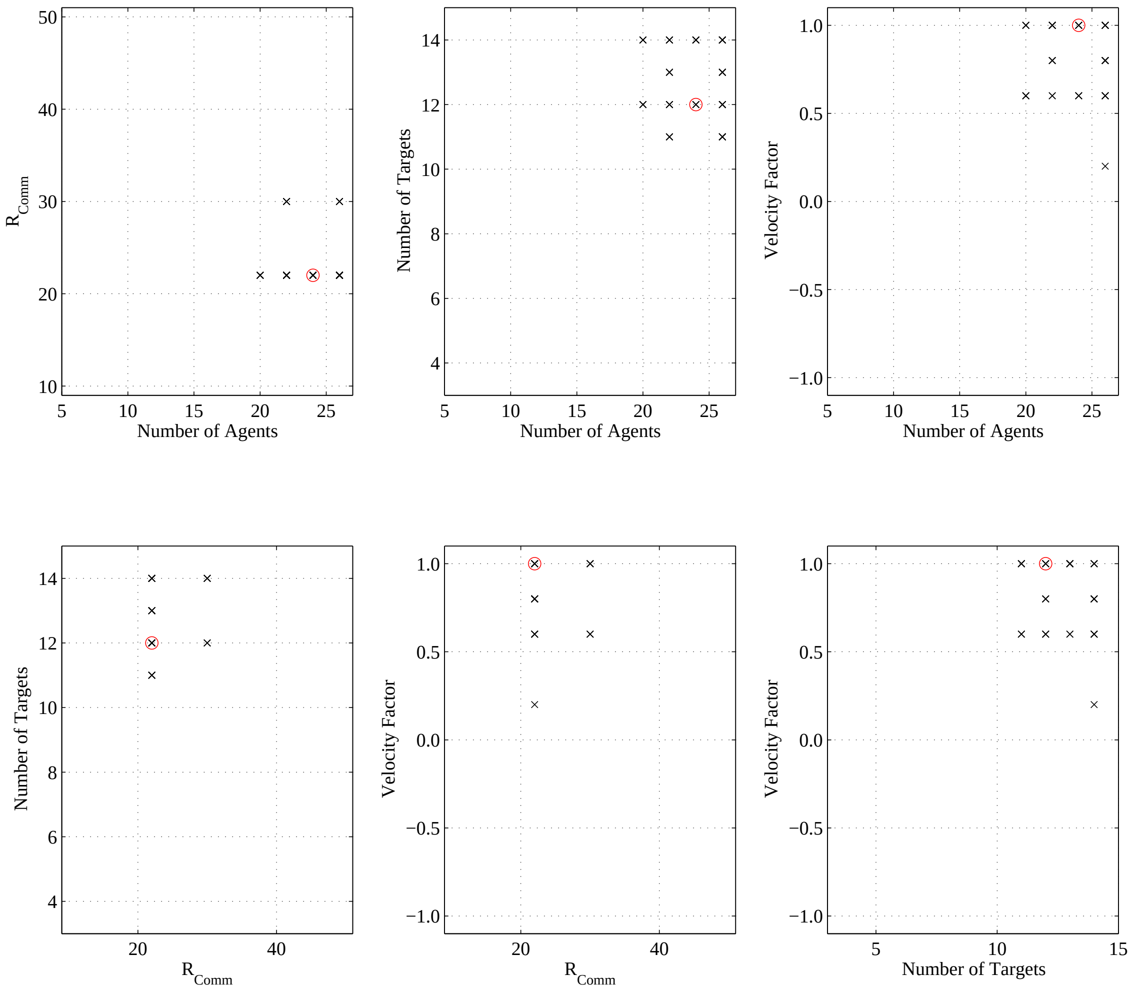

**Figure 87:** A single cluster of high design space divergence values with nearest point to cluster centroid shown as red circle

Figure 88 shows all of the 7 clusters selected from the cluster analysis.

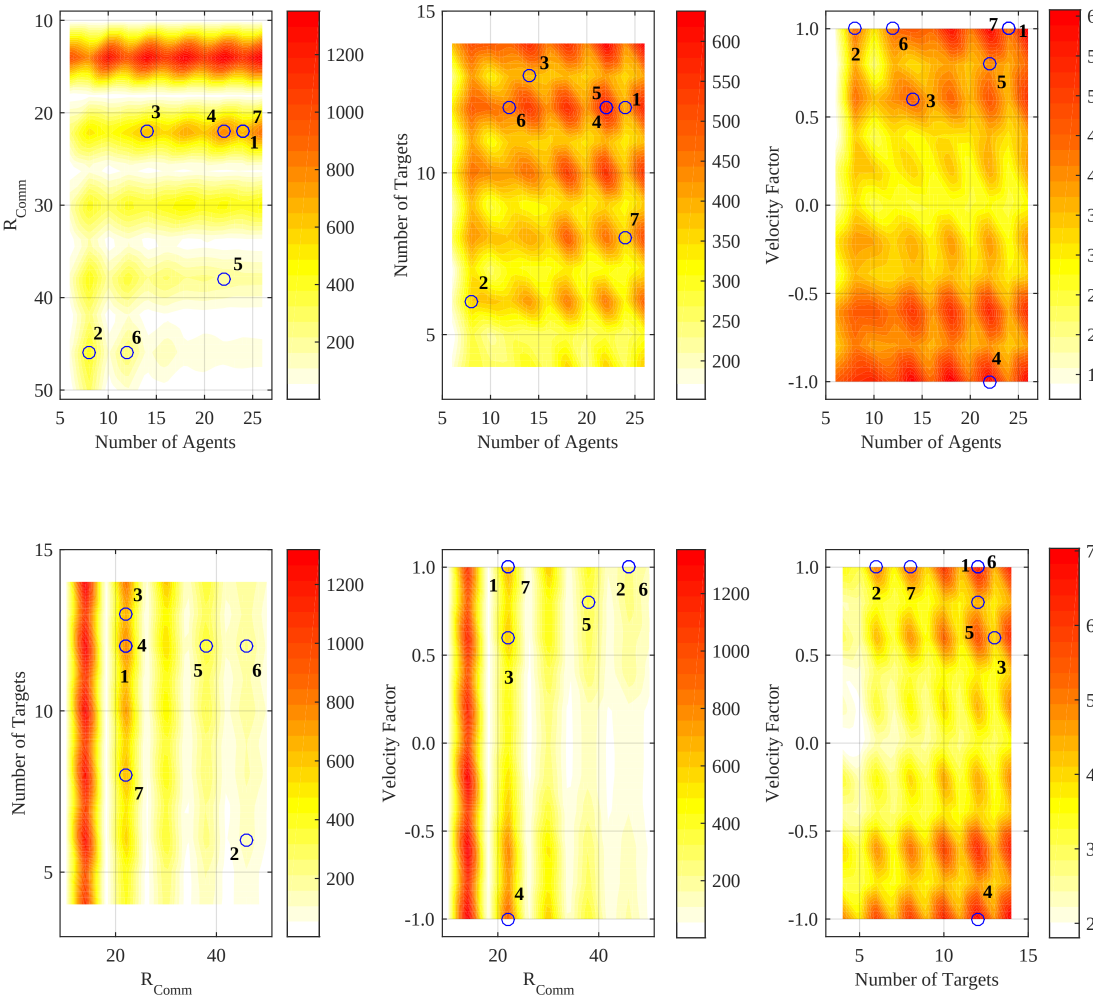

**Figure 88:** Cluster centroidal points

The clusters in Figure 88 are shown overlaid the design space divergence measure; however, the measure is averaged over the other dimensions. Because of this averaging, the visual identification of some of the clusters is difficult to accomplish if the cluster is a local effect that varies significantly from the average. In order to illustrate this issue, cluster #2 in Figure 88 is difficult to justify based on the contour plots: it is not clear that cluster #2 represents a location with one of the highest design space divergence measures. To demonstrate that this is due to high-dimensional structure projected onto a lower dimensional space, instead of using the averaged contour plots, the contours are taken at a slices through the cluster location. As shown in Figure 89, the distribution of the design space divergence measure is significantly different from the averaged distribution shown in Figure 88. It becomes much more clear how the local structure represents a location of high design space divergence.

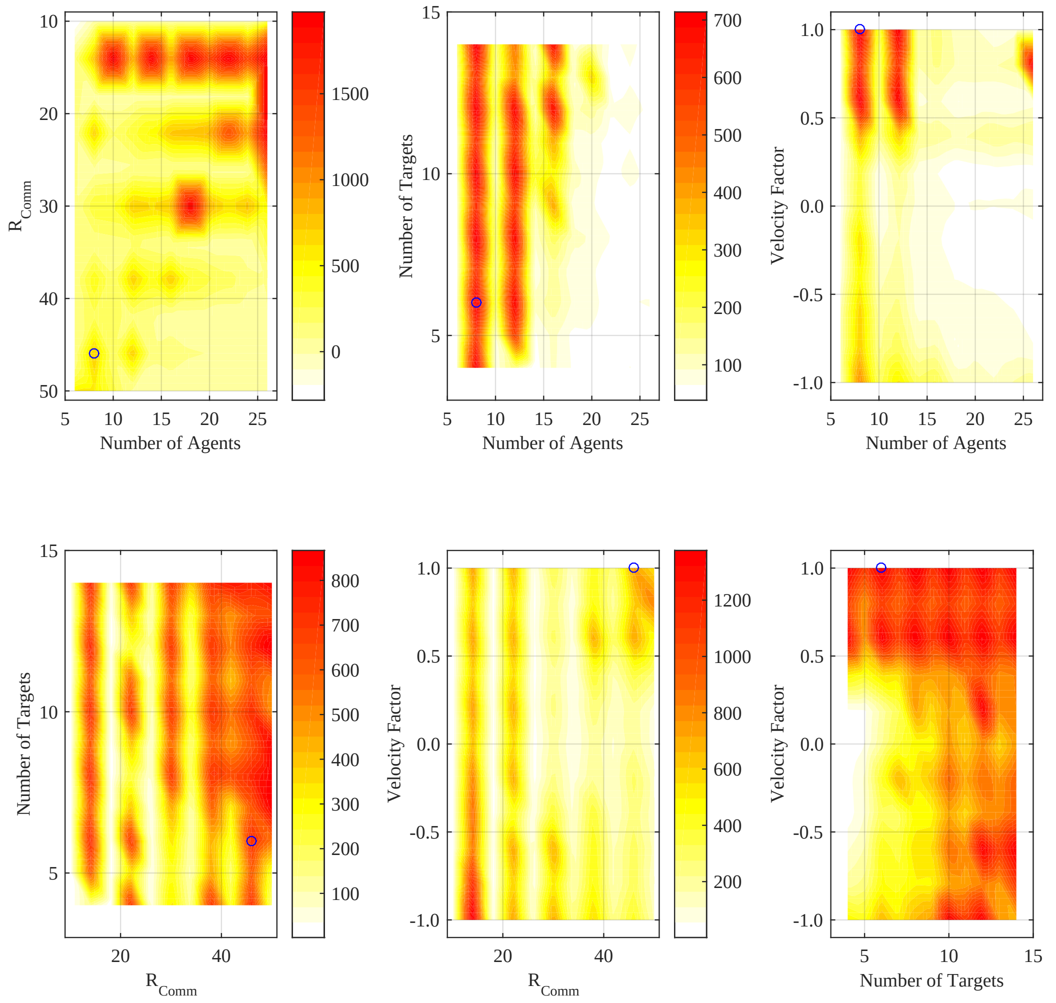

**Figure 89:** Design space divergence measure contour plot taken at slices through cluster #2

Although all of the clusters are potential emergent design points, this case study will focus on two of the clusters identified above as candidates for emergence. Clusters #1 and #2 are selected for further investigation. These two candidates are far away from each other in the design space so they represent two distinct solutions. They also correspond to the two types of interesting solutions initially identified in the chapter on the Distributed

Multi-Agent Surveillance System (Chapter 6.5.3). Cluster #1 will be labeled as Scenario A, a solution using a large number of agents. Cluster #2 will be labeled as Scenario B in which a small number of agents are used. Both scenarios will be investigated in depth below.

#### *9.4.0.1 Scenario A*

Scenario A represents a critical transition in system performance using a large number of agents. In this scenario, the number of agents is increased from 22 to 26 and the communication range is increased from 18 to 26. The design point values at this transition are shown in Table 26.

**Table 26:** Scenario A design space variables

| **Design Variable** | **Point 1** | **Point 2** |
|---|---|---|
| Number of Agents | 22 | 26 |
| Communication Range | 18 | 26 |
| Detection Range* | 9 | 13 |
| Number of Targets | 12 | 12 |
| Agent-Target Velocity Factor | 1 | 1 |

* assumed to be 1/2 of the Communication Range

Figure 90 shows the variation of the system performance as a function of the design variables including the design points corresponding to Scenario A. The green and blue dots correspond to the design points in this scenario and illustrate the significant change in system performance between the two points.

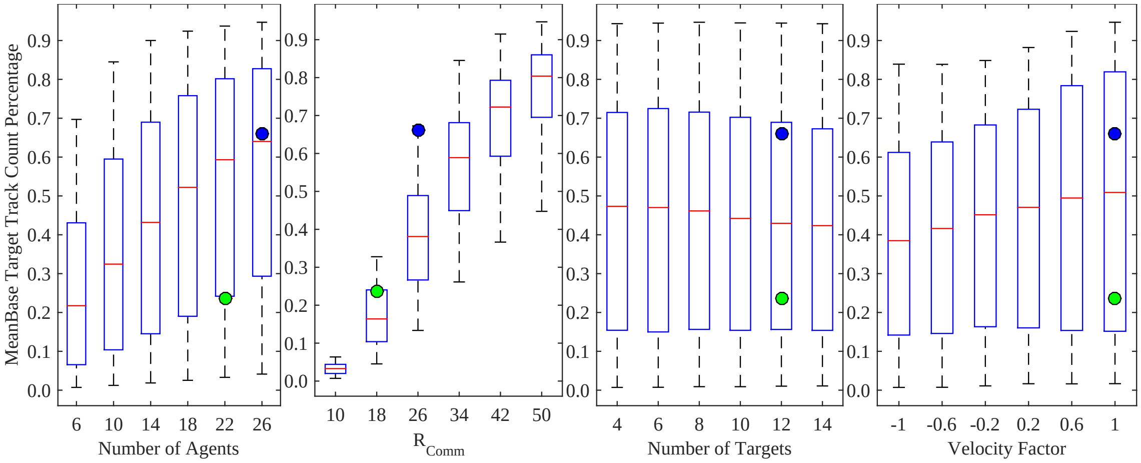

**Figure 90:** Box plots showing the variation of the mean of the Base Target Track Count Percentage (BTTCP) versus design variable parameters including design points at Scenario A

Figure 91 shows the contour plots of the system performance as a function of the design variables including the design points corresponding to Scenario A. The green and blue dots correspond to the design points in this scenario and illustrate the significant change in system performance between the two points. The use of contour plots helps to illustrate how this scenario is related to rapidly changing regions in the design space (i.e., large gradients).

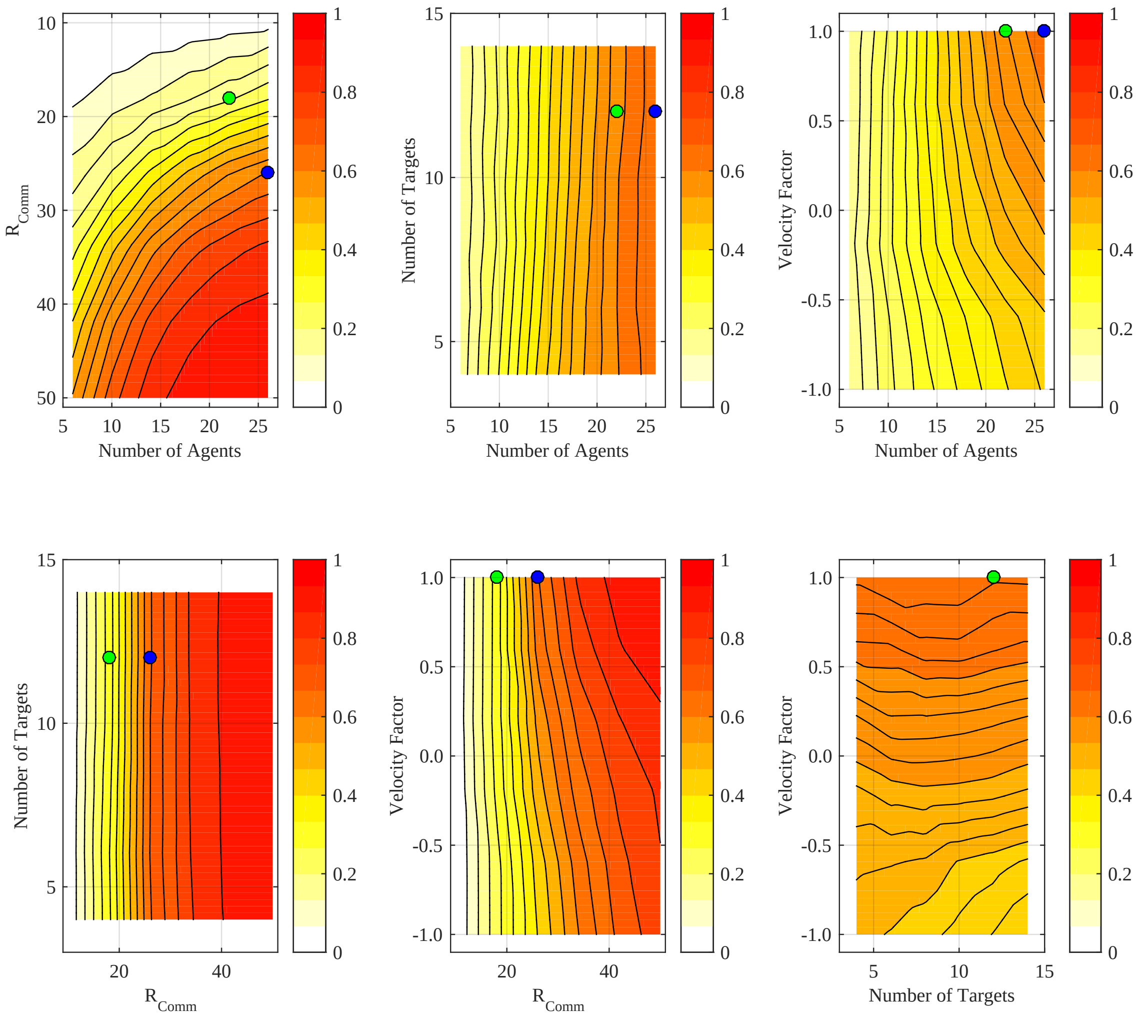

**Figure 91:** Contour plots showing the variation of the mean of Base Target Track Count Percentage (BTTCP) for design variable pairs including design points at Scenario A

Figure 92 illustrates the variation of the standard deviation of the system performance. Since the design space divergence measure considers not only shifts in the mean but also changes in the spread of the underlying probability distribution, it is interesting to see that this identified scenario also represents a location where there is a significant shift in variance from one design point to the next. It is clear that the number of agents and communication range dominate the response; however, by examining the variation of the spread of the probability distribution as a function of the design variables. It can be seen how the values for the other design variables (i.e., number of targets and velocity factor) are selected due to their influence on increasing the change in the variance from one design point to the next. Using just the mean response, the effect of the number of targets is very small; in fact, it is tempting to say that the number of targets is an insignificant variable. However, the analysis shows that the number of targets is important in determining the variance of the system performance. It can be seen that the scenario involving 12 targets represents a local maximum in the standard deviation of the system response.

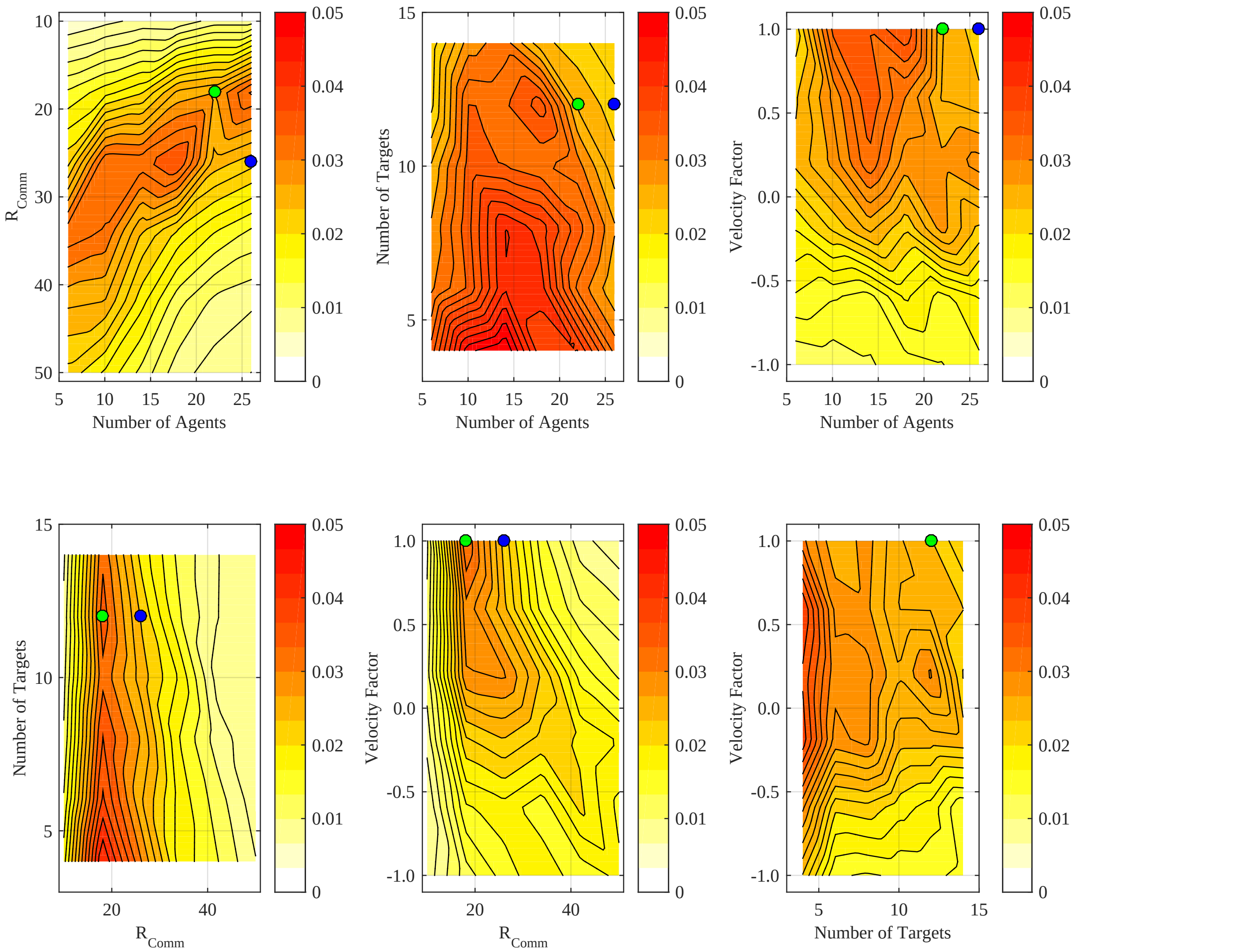

**Figure 92:** Contour plots showing the variation of the standard deviation of Base Target Track Count Percentage (BTTCP) for design variable pairs including design points at Scenario A

While the above figures are useful in visualizing the design space, it is useful to see the underlying probability distributions for this scenario. The probability distribution for the Base Target Track Count Percentage (BTTCP) measure for both design points is shown in Figure 93. This figure makes clear the significant shift in system performance from one design point to the next: the change in design variables corresponds to over a doubling of the system performance measure.

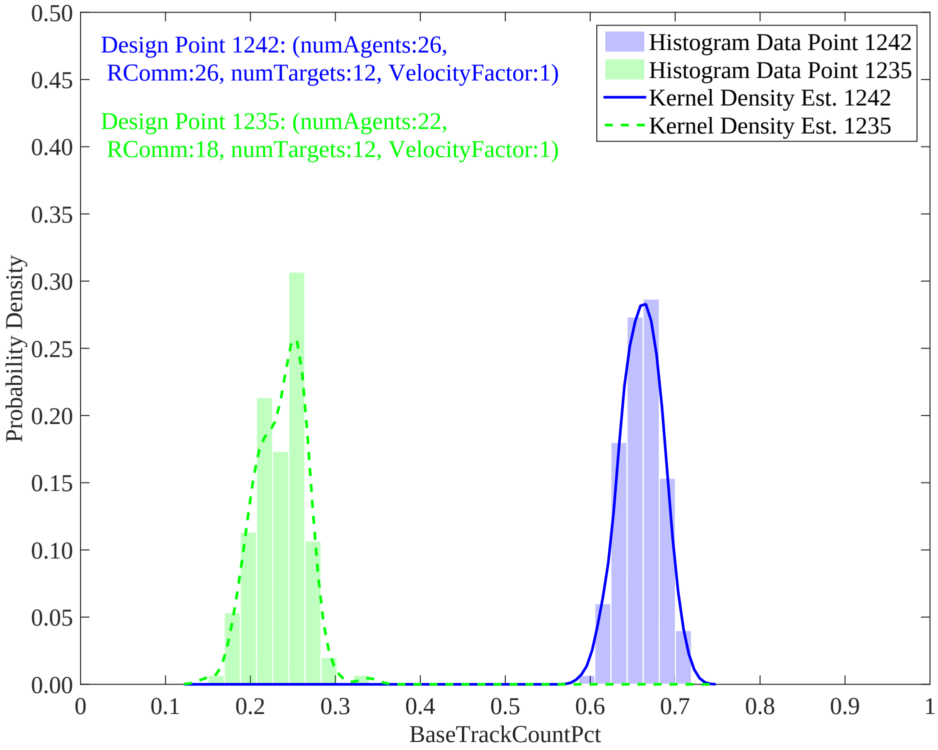

**Figure 93:** Probability distributions for design points at Scenario A

To further substantiate that this design point corresponds to an emergent design, the corresponding network degree over the course of the simulation for both design points in this scenario are shown in Figure 94. Each line in the figure represents the average of 150 replications. Moncion et al. [190, p. 5] argue that the mean degree of the network graph should increase as a complex system organizes. The degree of a node is the number of connected edges; therefore, the average network degree is a measure of the average number of links for all of the nodes. This analysis shows that there is a significant difference in the system connectivity, with Point 2 having a steady-state degree of roughly 6.6 while Point 1 has a steady-state degree of approximately 3.1. The oscillations in the network degree is due to the agents returning to base to refuel. It is interesting to note that these oscillations die out for Design Point 2 while they continue with only very slight damping in Design Point 1. This demonstrates that the system undergoes a significant transition from disorganized behavior to organized behavior in which it takes advantage of the mechanisms built into the system to produce a system which goes beyond the capabilities of any single agent to create an effective, coherent system.

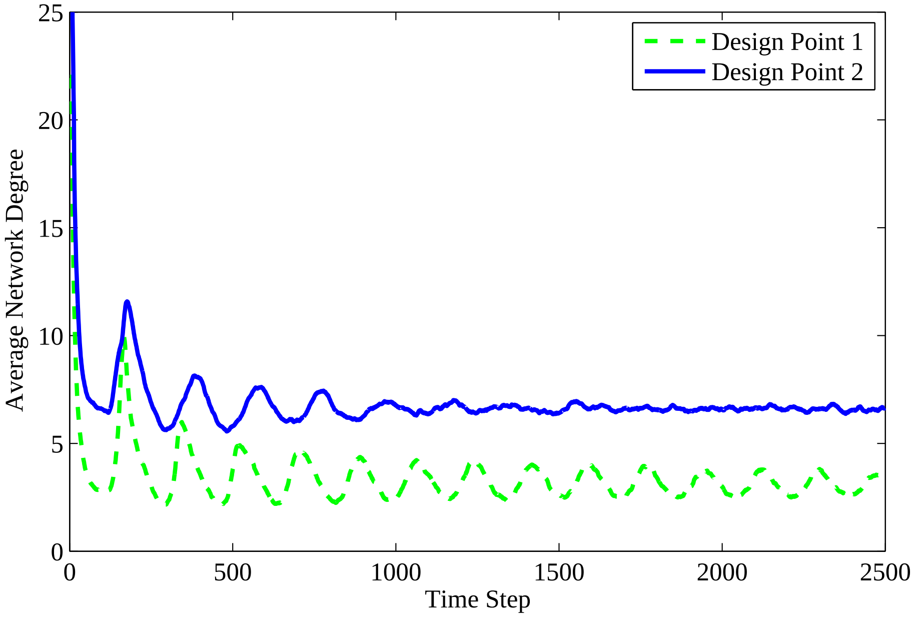

**Figure 94:** Average network degree vs. time for Scenario A design points

### *9.4.0.2 Scenario B*

Scenario B represents a critical transition in system performance using a small number of agents. In this scenario, the number of agents is increased from 6 to 10 and the communication range is increased from 42 to 50. The design point values at this transition are shown in Table 27.

**Table 27:** Scenario A design space variables

| **Design Variable** | **Point 1** | **Point 2** |
|---|---|---|
| Number of Agents | 6 | 10 |
| Communication Range | 42 | 50 |
| Detection Range* | 21 | 25 |
| Number of Targets | 6 | 6 |
| Agent-Target Velocity Factor | 1 | 1 |

* assumed to be 1/2 of the Communication Range

Figure 95 shows the variation of the system performance as a function of the design variables including the design points corresponding to Scenario A. The green and blue dots correspond to the design points in this scenario and illustrate the significant change in system performance between the two points.

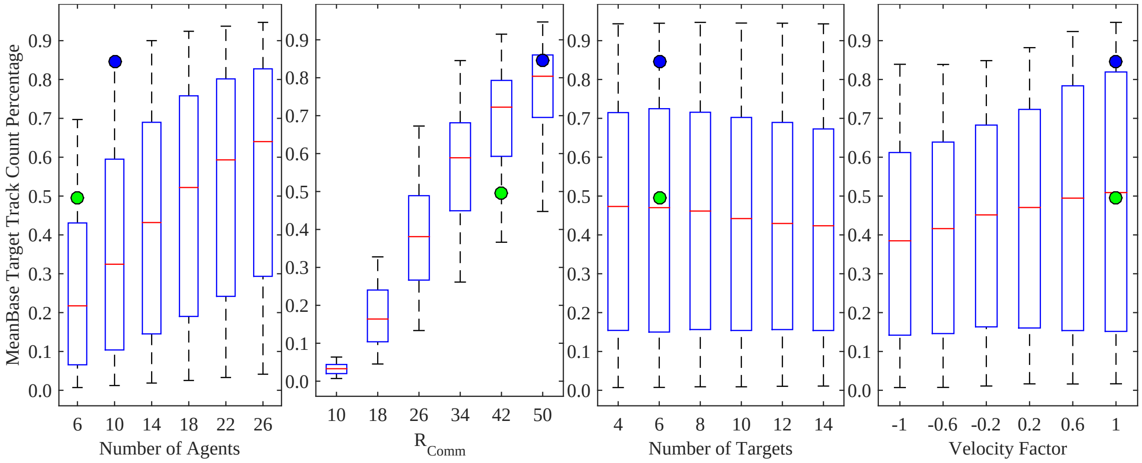

**Figure 95:** Box plots showing the variation of the mean of the Base Target Track Count Percentage (BTTCP) versus design variable parameters including design points at Scenario B

Figure 96 shows the contour plots of the system performance as a function of the design variables including the design points corresponding to Scenario B. The green and blue dots correspond to the design points in this scenario and illustrate the significant change in system performance between the two points. The use of contour plots helps to illustrate how this scenario is related to rapidly changing regions in the design space (i.e., large gradients).

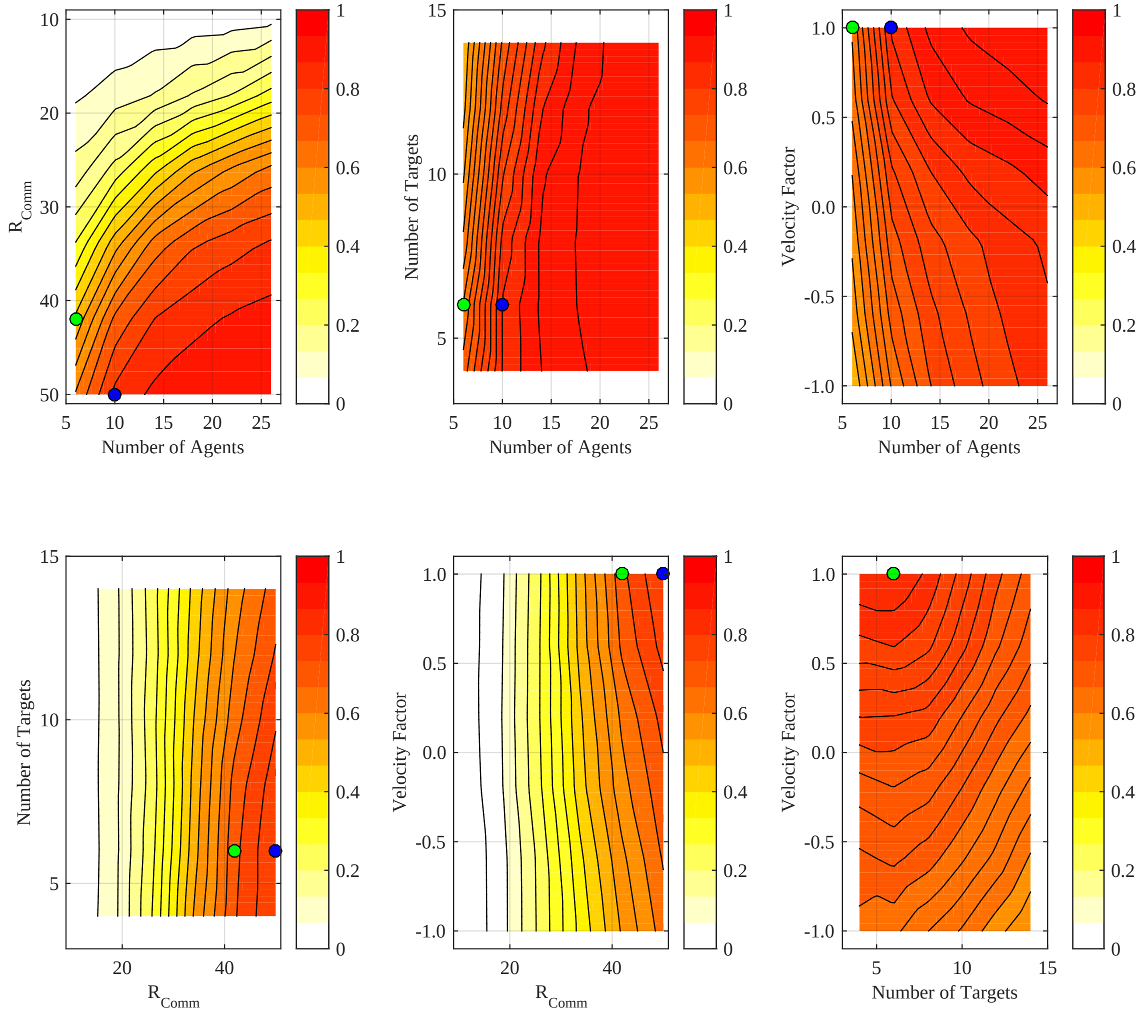

**Figure 96:** Contour plots showing the variation of the mean of Base Target Track Count Percentage (BTTCP) for design variable pairs including design points at Scenario B

Figure 97 illustrates the variation of the standard deviation of the system performance. Since the design space divergence measure considers not only shifts in the mean but also changes in the spread of the underlying probability distribution, it is interesting to see that this identified scenario also represents a location where there is a significant shift in variance from one design point to the next. It is clear that the number of agents and communication range dominate the response; however, by examining the variation of the spread of the probability distribution as a function of the design variables. It can be seen how the values for the other design variables (i.e., number of targets and velocity factor) are selected due to their influence on increasing the change in the variance from one design point to the next. Using just the mean response, the effect of the number of targets is very small; in fact, it is tempting to say that the number of targets is an insignificant variable. However, the analysis shows that the number of targets is important in determining the variance of the system performance. It can be seen that the scenario involving 6 targets represents a local maximum in the standard deviation of the system response.

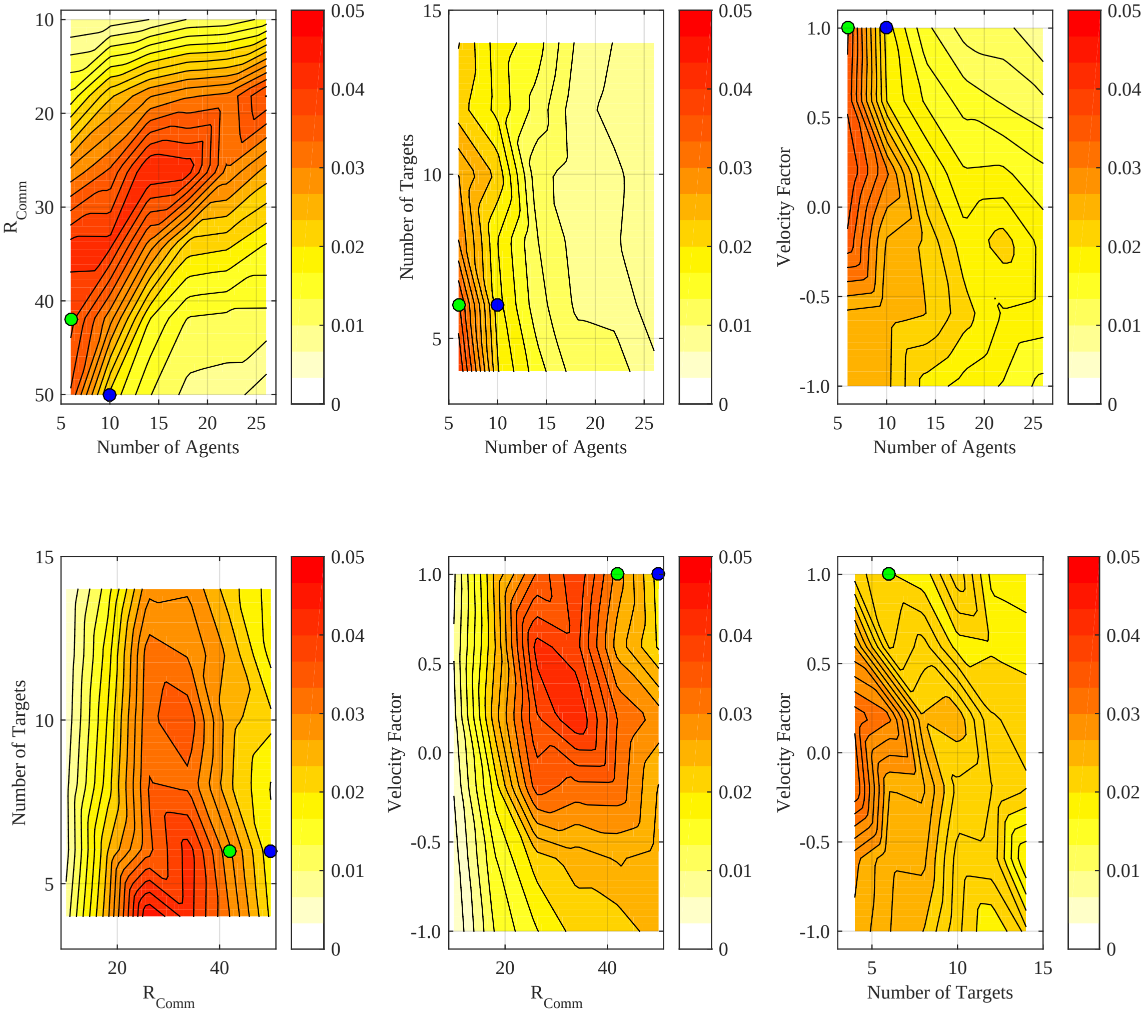

**Figure 97:** Contour plots showing the variation of the standard deviation of Base Target Track Count Percentage (BTTCP) for design variable pairs including design points at Scenario B

While the above figures are useful in visualizing the design space, it is useful to see the underlying probability distributions for this scenario. The probability distribution for the Base Target Track Count Percentage (BTTCP) measure for both design points is shown in Figure 98. This figure makes clear the significant shift in system performance from one design point to the next including both shifts in mean and variance.

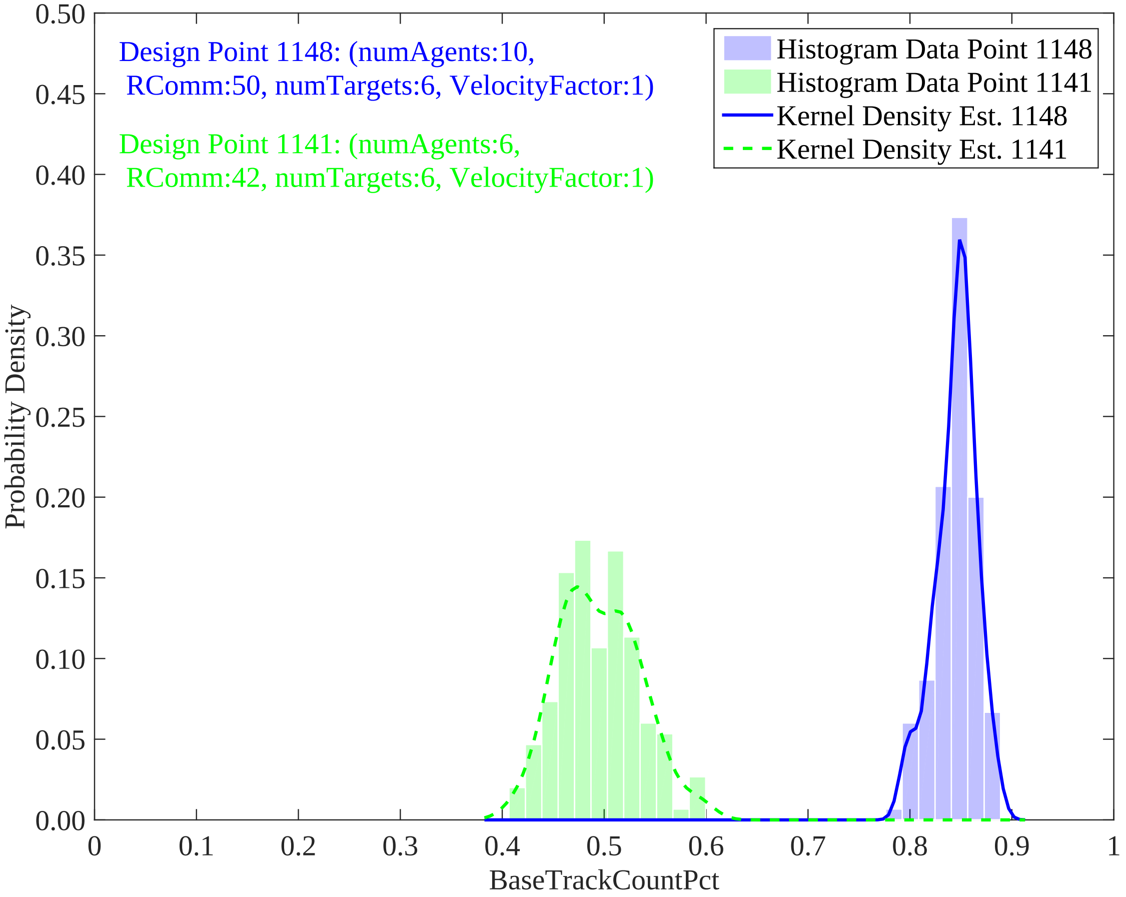

**Figure 98:** Probability distributions for design points at Scenario B

# CHAPTER X

# SUMMARY AND CONCLUSIONS

## *10.1 Summary*

The objective for this research was to develop a new methodology which is capable of systematically exploring the design space of a distributed multi-agent system and identifying parameters that may yield emergent behavior. A new measure for emergence, Design Space Divergence, was proposed as part of this research. It was demonstrated that the new measure of emergence corresponds to regions in the design space where the probability distributions of the system effectiveness measures are rapidly changing as a function of the system parameters. This measure of emergence allows us to identify critical transitions in behavior. Identifying critical regions is potentially more useful because it allows us to see situations where our system can just as easily change behavior towards a less effective state. This gives the system designer the ability to tailor the use of emergence for various purposes. Keeping the system near the critical transition region will make the system less robust; however, it may make it more adaptable and able to quickly adjust to changing conditions.

**Research Objective**: The objective of this research is to develop a method for identifying emergent behavior, both beneficial and detrimental, in complex systems.

**Research Goal**: The proposed methodology should be able to: 1) identify design points (i.e., parameter settings) which are candidates for emergent behavior, 2) present the candidate points in a manner which allows the designer to characterize the design space and make proper inferences in coming up with strategies to exploit or avoid behavior. The success of this research will be judged based on the ability of the proposed approach to rigorously and robustly identify the design space parameters which lead to emergent effects.

A case study using a distributed multi-agent system was used to test the methodology and emergence measure. The results from this case study show that the SEED methodology presented in this research has successfully completed the research goal. The methodology did provide design points which correspond to interesting and potentially emergent designs. Furthermore, the methodology is able to provide views about the behavior of the system across the design space which gives the designer confidence that the results are not spurious results but rather due to a fundamental underlying behavior.

### 10.1.1 Summary of Research Questions

During the development of the emergence measure and the SEED methodology, a number of research questions were examined. The first research question (**RQ1**) examined the concept of emergence in order to guide the process of developing a definition that is useful for engineering design but also consistent with the literature and with our understanding of distributed multi-agent and other complex systems.

- **(RQ1)**: What is emergence?
  - **(RQ1.1)**: What are the characteristics of emergence and what makes it difficult to understand and predict?
  - **(RQ1.2)**: How can emergence be defined?
  - **(RQ1.3)**: How can emergence be detected or measured?
  - **(RQ1.4)**: How can emergence be understood in the context of engineering?

A number of definitions of emergence were found in literature and were considered for their suitability in describing emergent collective behavior in a distributed multi-agent system. Since an existing definition did not capture emergence in objective and actionable way, a new definition was proposed that allowed it to be used for engineering. Various techniques for detecting or measuring emergence were reviewed for their applicability to the distributed multi-agent problem. Divergence-based approaches were deemed to be most appropriate and general enough to be applied to a range of multi-agent systems. In fact, the probabilistic nature of the divergence-based approach makes it applicable to most stochastic problems.

Finally, a new emergence measure—Design Space Divergence—was proposed as a way to measure emergence through design space exploration.

The second research question (**RQ2**) examined ways of analyzing and designing complex systems. The purpose of this research question was to understand the underlying problem to make sure that the proposed solution captures the important features of complex behavior and emergence.

- **(RQ2)**: How do we analyze and design complex systems?
  - **(RQ2.1)**: What characterizes a complex system?
  - **(RQ2.2)**: What causes a system to be complex?
  - **(RQ2.3)**: How can the complexity of a system be measured?
  - **(RQ2.4)**: How should a complex system be analyzed?

The subject of complexity was explored in order to gain insight into the distributed multi-agent problem. By understanding the mechanisms that give rise to complexity, they can be harnessed to generate "good" complexity—complexity that leads to emergent behaviors that helps to create an effective system. Although complexity and emergence are often related, a review of the literature showed that most of the measures of complexity were not appropriate for measuring emergence for the multi-agent distributed problem. Many complexity measures were focused on measures of organization, complexity of description, or amount of information processing. However, all of these approaches are difficult to justify as effective measures when designing systems. The final part of this research question sought to answer what the right approach is for analyzing complex systems. Various design methodologies and approaches were reviewed to help understand how complexity helps shape the design process. It became clear that complexity makes design harder by obfuscating how inputs lead to certain outputs. This necessitates a rigorous, traceable, and objective methodology for systematically exploring the input space and then using insights from the results to help guide future iterations of the design process.

Finally, the main research question (**RQ3**) drove the development of the SEED methodology. The experiments that were performed in this research were done to answer portions of this research question. The results from the experiments helped to select from the various alternatives available when developing the methodology. Not only did they help to make decisions, they also illustrated the consequences for making the incorrect selection when developing and using the methodology.

- **(RQ3)**: What is the appropriate methodology for engineering complex systems?
  - **(RQ3.1)**: What are the required features of a design methodology for emergence?
  - **(RQ3.2)**: How can the design space of a complex system be systematically explored in order to identify emergent behavior?
    - **(RQ3.2.1)**: How should the design points be distributed in the design space?
  - **(RQ3.3)**: What is the appropriate way to perform design space exploration on a complex, stochastic space?
  - **(RQ3.4)**: What is the appropriate way to represent and evaluate probability distributions?
    - **(RQ3.4.1)**: Should probability distributions be modeled as parametric or non-parametric distributions?
    - **(RQ3.4.2)**: What is the appropriate statistical distance measure to use when evaluating probability distribution similarity?
  - **(RQ3.5)**: How can emergent behavior be identified in higher-dimensional systems?
    - **(RQ3.5.1)**: How can the design space be visualized (especially in high-dimensional spaces) in order to facilitate emergent behavior identification and system characterization?

A number of these research questions lead to hypotheses and corresponding experiments to these them. These hypotheses and experiments will be reviewed in the next section. In summary, all of the research questions that were introduced at the beginning of the thesis have been addressed either through a literature review or argumentation or through an experiment.

## *10.2 Summary of Hypotheses and Experiments*

The hypotheses developed in the course of this research are shown below. Although a number of hypotheses were either partially or completely falsified, all served a useful purpose. The hypotheses related to developing the methodology helped to identify whether there is a consequence or not for making the incorrect selection within the methodology. The two most important hypotheses were **HYP1** and **HYP5**, which tested the proposed emergence measure and overall methodology, respectively. Both of these hypotheses were supported through the experiments and case study.

### 10.2.1 Hypothesis 1

- **Hypothesis 1 (HYP1)**: Assuming a desired range of system behavior has been identified, large divergence measures of the distributions in the response variables reflects possible emergent conditions near critical conditions while low divergence measures reflect robust design points.

**Hypothesis 1** represents one of the fundamental hypotheses of this research. This hypothesis posits that the transition to emergent behavior can be identified by a high Design Space Divergence measure. This hypothesis was tested through **Experiment D** (i.e., the case study). The data exploration phase identified a number of points with high DSD measures. Several of these points were investigated in depth in the case study. Scenario A showed a marked transition in behavior between the corresponding neighboring points. An investigation into the average network degree (which has been argued [190] that this measure would increase as the system self-organizes) versus time for both of these points show how there is a significant difference between the average network degree before versus after the critical transition. This seems to be strong evidence of self-organizing, emergent behavior in the system. Since this behavior also corresponded with a high DSD measure, this supports **Hypothesis 1**.

### 10.2.2 Hypothesis 2

- **Hypothesis 2** (**HYP2**): If there is a similar sampling density, there will be no significant difference in discriminability between sampling methods (i.e., both full factorial and space filling designs will yield the same similar divergence measures).

**Hypothesis 2** is used to help develop the SEED methodology. The methodology requires a thorough sampling of the design space. However, since orthogonality is not required for the evaluation of the Design Space Divergence measure, there is no strong a priori justification for saying with certainty that one sampling method will yield better results than the another. Since sampling density initially appears to be important, the hypothesis is essentially a null hypothesis that there will be no significant difference in results between the sampling methods given that the sampling density is similar. **Experiment A** was used to test **Hypothesis 2** by testing various sampling designs and number of points per design. The results from this experiment showed that not only was sampling density important, but having a regularly spaced design yielded much better results. This result partially supported **Hypothesis 2** but also demonstrated that sampling density was not sufficient since uniformity was also required for good discriminability.

### 10.2.3 Hypothesis 3

- **Hypothesis 3** (**HYP3**): There will be no qualitative difference in discriminability between statistical distance measures.

**Hypothesis 3** is also used as part of the development of the SEED methodology. An important component of the methodology is the selection of the statistical distance measure used. Although the analytical properties of each statistical distance could be used to gain insight into how its use would affect the Design Space Divergence measure, it is also dependent on the system being tested. Therefore, the more straightforward way to understand the behavior of the statistical distance measures is through an experiment. **Hypothesis 3** was akin to a null hypothesis that stated there will be significant difference in discriminability between the various measures. **Hypothesis 3** was tested via **Experiment B**. The results from that experiment showed that demonstrated how the probability distribution affects the statistical distance measure. This experiment was used to develop guidelines about the nature of the underlying probability distribution and the more appropriate statistical distance measure to use in that scenario. This experiment demonstrated that there was a significant difference between the various statistical distance measures and therefore the hypothesis is rejected. However, this result was not an issue since the goal was to help develop the guidelines and the success of the methodology is not dependent on the support or rejection of this hypothesis.

### 10.2.4 Hypothesis 4

- **Hypothesis 4** (**HYP4**): In situations where probability distributions are constrained or when a large degree of flexibility is needed, a nonparametric probability distribution provides a better model for capturing probabilistic features.

**Hypothesis 4** is also used to help develop the steps in the SEED methodology. Although the methodology could be developed with either a parametric or nonparametric probability distribution, I have argued that a nonparametric approach is more appropriate for complex systems. By using the more general approach towards modeling probability distributions, the methodology is more flexible and is better able to handle the variety of distributions that might be seen when analyzing and designing engineered systems. However, the nonparametric approach is not without costs—it requires more fine-tuning when performing the estimation and the convergence rates are much slower compared to parametric estimation techniques. This hypothesis is tested using **Experiment C**. This experiment investigated whether a parametric approach is possible for the Distributed-MASS problem. The results showed that a parametric approach was not possible and that the nonparametric approach provided a better solution for density estimation using the approach laid out as part of the SEED methodology. Therefore, **Hypothesis 4** was supported.

### 10.2.5 Hypothesis 5

- **Hypothesis 5** (**HYP5**): The proposed methodology is capable of identifying conditions for candidate emergent behavior in a complex system model.

Finally, **Hypothesis 5** sought to test the overall SEED methodology and whether it was capable of attaining the research goal of identifying candidates for emergent behavior within the design space. This hypothesis is related to **Hypothesis 1**, which was primarily testing whether the proposed Design Space Divergence measure corresponds with emergent behavior; however, **Hypothesis 5** takes this to the next step and is testing whether the proposed methodology is capable of identifying regions in the design space with high Design Space Divergence measures. Not falsifying **Hypothesis 1** is a prerequisite for **Hypothesis 5**. This hypothesis is tested using **Experiment D**. By carrying out the SEED methodology on the Distributed-MASS problem, it was demonstrated that a number of candidate design points exhibiting emergent behavior were identified. Therefore, the hypothesis was supported.

## *10.3 Contributions*

Two significant contributions from this research are the a new emergence measure and the SEED methodology. Both of these contributions have been successfully demonstrated on the case study to show the value that they provide to a designer when studying distributed multi-agent systems. Both of these contributions are solutions to the two main gaps that were identified early within this dissertation. These two contributions have helped to close the identified gaps.

**Identified Gap 1**: The design process for distributed multi-agent systems is ad hoc and heavily based on designer intuition. The goal is to create a methodology for systematically exploring the design space in order to make the design process more thorough and traceable.

**Identified Gap 2**: Few of the measures of emergence would be appropriate for use in a design space exploration methodology. Divergence-measures have been applied to identify emergence in time-series data but there exists a need to extend the approach to design space exploration.

The Distributed-MASS model was another contribution of this research. This model expanded the multi-agent surveillance simulation developed by Aksaray [1] to create a simulation that is more capable, flexible, and easier to use. Aksaray's simulation was completely refactored with new data structures that emphasize the agent-based nature of the simulation. The Distributed-MASS simulation easily allows for parallel processing, necessitated by the large amount of runs necessary needed to probabilistically characterize the design space.

Most importantly, the SEED methodology provides the capability to generate new *insights* into the distributed multi-agent problem. Exploratory analysis seeks to characterize the problem space and identify potential solutions as starting points in the design process. It has been demonstrated that the SEED methodology is capable of providing candidate solutions which exhibit emergent behavior. More so, these design points are effective *levers* [169], solutions that demonstrate ways of affecting change in the system behavior. These design points help identify interesting regions of the design space and help in the refinement of the parameter space as the design process progresses through a number of iterations. In addition to identifying particular solutions, the SEED methodology also generates data which can be used to characterize the design space and to gain confidence about how the system behavior changes throughout the design space. This not only helps to reduce the risk of re-work due to unforeseen behavior but also starts down the path of behavior assurance and system certification.

## *10.4 Areas of Future Research*

There are a few areas of future research that could be pursued to gain more insights into the challenges and opportunities when designing distributed multi-agent systems. The first area of additional research could continue the exploration of the multi-agent persistent surveillance simulation via improvements to the simulation.

### 10.4.1 Improvements to Distributed-MASS

For the purpose of this research, the Distributed-MASS simulation was purposefully kept relatively simple in order to allow a thorough analysis and explanation of the system behavior. However, further improvements will improve the fidelity of the simulation and to allow explorations into additional behaviors enabled by new mechanisms for agent coordination and adaptation.

To improve the simulation fidelity, a number of assumptions that were used during the development can be replaced with more realistic models. In the Distributed-MASS simulation, the agents were assumed to be point masses and were able to maneuver instantaneously throughout the 2D environment. A higher fidelity three-degree of freedom (3DOF) flight dynamics model would allow for additional investigations into the maneuverability requirements for the individual vehicles and how they affect the system effectiveness. The 3DOF flight dynamics model would require the addition of the third dimension (i.e., altitude). Additional collision avoidance rules can be added to the simulation and to help deconflict multiple UAVs in close proximity. Communications and sensor behavior was assumed to be perfect; however, a more realistic model would include losses and other degradations.

In addition to the simulation fidelity, improvements to the simulation implementation are also possible. The required runtime for all of the simulations was clearly prohibitive. Although runtime issues were considered during the development of Distributed-MASS, it is clear that even more significant steps must be taken to help alleviate this issue. Potential improvements include trading simulation memory for processing power. A number of computationally expensive network properties were calculated at each time step in the simulation. It seems possible that building in a memory into the simulation will allow expensive computations to be performed only if the connectedness changes between time steps. Although the network connectedness does change often, it seems likely that there is a significant savings possible since the number of times the connectedness changes is still smaller than the total number of time steps in the simulation.

The effects of adding additional intelligence to either the agents or the targets might have interesting effects on the behavior of the system. Adding more intelligence to the system along with more parameters to control the behavior of the agent itself would make the design space more interesting and even harder to explore; however, that may potentially make the proposed methodology even more compelling as a solution. As more mechanisms are allowed, such as adaptation, the potential for truly surprising behavior is unlocked.

### 10.4.2 Improvements to the SEED Methodology

Potential additional research on the methodology was briefly mentioned within the methodology chapter: adaptive sampling strategies, for both design point selection and replication, could improve the efficiency of the methodology. The multi-agent surveillance problem demonstrated that there are often large regions of the design space that are rather uniform and uninteresting. For very low communication ranges, the system performed poorly everywhere. On the other hand, for very large communication ranges, the system was almost always near maximum performance. Design points in either of these regions would not need a large amount of replications to characterize the probability distribution at each point. Similarly, design points would not need to be densely sampled in this space since the performance is not expected to change rapidly. The goal of an adaptive method is to locate points in more interesting regions of the design space—where the Design Space Divergence measure is expected to be large due to a transition in underlying behavior.

Exploring even higher dimensional problems would also be important to show that the methodology remains useful as the problem grows in size. A number of steps in the methodology work best for relatively small dimensional problems but will need adjustment to handle larger problems. For example, the $k$-d tree partitioning method works is efficient compared to the brute-force approach only when $N \gg 2^k$. This means that as $k$ approaches on the order of 10 dimensions, the $k$-d tree partitioning method is likely to be no better than brute force, which itself will be very computationally expensive. Another issue with high dimensions is that the design space distance measure may become less significant. As the number of dimensions increases, points tend to get equally far away from each other when using the Euclidean distance, which results in a less significant distance measure. Continuing to adapt the methodology to handle larger dimensions will improve its applicability and usefulness.

# APPENDIX A

# PHILOSOPHY OF EMERGENCE

**Chapter Road Map**: The discussion in this chapter is primarily from the philosophy of science viewpoint and addresses the many—sometimes conflicting—properties of emergence that have made it a pervasive, captivating, yet controversial topic.

## *A.1 A Brief History of Emergence*

This section will examine the historical context of emergence and how the definition and philosophy of emergence has changed through time. This has lead some to believe that emergence is a current fad and a "buzz-word" that will eventually go away. While there have been legitimate issues in creating rigorous and coherent definitions for emergence, the concept is compelling enough that it has continued to attract attention for over a hundred years—and its popularity today is as high as ever.

### A.1.1 The Emergence of Emergence: British Emergentist Movement

Although the intuitive notion of emergence can be traced back to Aristotle's famous quote of the whole being greater than the sum of its parts, a more complete foundation for emergence was laid during the British Emergentist movement in the mid-1800s to the early part of the 20th century. It was this movement that formed the first major push to advance the concept of emergence as a principle of scientific understanding. McLaughlin [183] discusses the history of emergence during the British Emergentist period between the years 1843–1925. This period spans from J.S. Mill's System of Logic (1843) to C.D. Broad's The Mind and Its Place in Nature (1925). This movement was a philosophical movement that included the first coinage of the term emergent. Since this period coincided (and was therefore influenced by) with great advances made in physics, chemistry, and biology, this movement sought to understand the relationship between properties and the nature of matter and science. This period focuses heavily on developing a principle of composition—how properties combine among constituent parts. The essential question driving this debate is whether or not the properties or behavior of an object can be determined from its constituent parts. It is also insightful to keep in mind the problems that the British Emergentists were facing. A large portion of the British Emergentist movement was devoted to trying to understand the nature of Life. The main question being asked was whether Life could be reduced to nothing but physico-chemical processes. What gave rise to life, the mind, and biological evolution? Could these topics be answered by appealing to basic physical and chemical properties and without appealing to any supernatural forces? As Broad writes, "in a philosophy based on the procedure sanctioned by progress in scientific research and thought, the advent of novelty of any kind is loyally to be accepted wherever it is found, without invoking any extra-natural Power (Force, Entelechy, Elan, or God) through the efficient Activity of which the observed facts may be explained" [193, p. 2].

While the British Emergentist movement was important because it established emergence as a concept in science and philosophy, it also posed a number of problems. Positions taken by Emergentists could be considered by some as anti-science. A philosophy of emergence that claims some phenomena admit "no explanation" and rely on the investigator to accept it with "natural piety" [5, pp. 46–47] would be inconsistent with the analytic methods of science require breaking down the object studied into its constituent components. It was debated whether or not chemistry, biology, and even psychology could be explained by examining the principles of composition for elementary particles. The physics and chemistry of atomic elements were a common subject of Emergentist thought. During this period, it appeared that chemical phenomena resulted in emergent properties as compared to the constituent elements. Emergentists claimed that it was impossible to predict chemical many properties; however, as McLaughlin explains, the quantum mechanical revolution would go on to discredit Emergentist thought by providing a law governing many atomic properties [183]. Similarly, in the field of biology, "far more damaging to the cause of emergent evolution was the rise of the science of genetics in the 1920s and 1930s and the triumph of an analytical, experimental approach to biology" [76, p. 20]. The history of British Emergentism provides an interesting lesson. It is clear that many of the arguments used to support Emergentism were incorrect; however, the arguments were based on the understanding of the science at the time. As physics underwent a major paradigm shift in the first half of the 20th century with the improved atomic models and the introduction of quantum theory, so did our understanding of the governing laws. As the possibility of emergent configurational forces faded, so did the Emergentist movement. According to some, the lesson is that it is quite possible that phenomena which are currently regarded as being emergent could be easily deducible in the future. This gave rise to the idea that emergence is nothing but a stop-gap measure that is a product of incomplete knowledge and is theory-dependent. By the end of the Emergentist movement, the popularity and relevance of emergence was diminished, replaced by the great advances made by reductionist scientific approaches in the field of chemistry, biology, and physics. For the time being, it seemed that the reductionist approach had won the war against the Emergentists.

### A.1.2 Re-Emergence of Emergence: The Limits of Reductionism

The central battle in the philosophy of emergence is commonly framed as the reductionists versus the emergentists. The reductionists believe that everything in the universe can be explained by an underlying micro-level behavior. Emergentists claim that some effects cannot be explained by any underlying mechanisms. However, both viewpoints are important to solving problems in science and engineering. Wimsatt explains how both reductionists and emergentists are neglecting the contributions that the other side brings to the understanding of complex systems:

> "Both sides here conflict with most scientists' intuitions about when something is emergent. Discussions of emergent properties in nonlinear dynamics, connectionist modeling, chaos, artificial life, and elsewhere give no support for traditional antireductionism or woolly-headed antiscientism. Emergent phenomena like those discussed here are often subject to surprising and revealing reductionistic explanations. But reductionists often misunderstand the consequences

> of their explanatory successes. Giving such explanations does not deny their importance or make them any less emergent—quite the contrary: it explains why and how they are important, and ineliminably so." [254]

The new question is not whether emergence or reductionism is correct; rather, how can both of these approaches be used to advance science? The success of reductionism in science cannot be ignored. The advances made in understanding how the world works using a reductionist approach are clear. Any attempt to frame emergence as a superior and exclusive explanatory approach in science is doomed to fail. As Anderson acknowledges, "the reductionist hypothesis may still be a topic for controversy among philosophers, but among the great majority of active scientists I think it is accepted without question" [9]. Clearly, a framework for emergence has to allow for and utilize the benefits of reductionism. However, reductionism has its limits. Laughlin summarizes this dilemma, "We have succeeded in reducing all of ordinary physical behavior to a simple, correct Theory of Everything only to discover that it has revealed exactly nothing about many things of great importance" [168]. Instead, we should look to see where the reductionist approach has failed and how we can overcome those failures. As Goldstein notes, "it wasn't emergence itself that was the problem but the examples that the emergentists used to exemplify it" [128]. Explaining the resurgence of emergence as scientific principle, Kim writes:

> "but the idea of emergence refused to die, continuing to attract a small but steady stream of advocates from both the philosophical and the scientific ranks, and it now appears to be making a strong comeback. This turn of events is not surprising, given the nearly total collapse of positivistic reductionism and the ideal of unified science which was well underway by the early '70s. The lowly fortunes of reductionism have continued to this day, providing a fertile soil for the reemergence of emergentism." [158]

One of the well-known champions of the concept of emergence that came out of this period is P. W. Anderson, a Nobel Laureate known for his work in condensed matter physics. He is a contemporary champion of the concept of emergence in science. His paper "More is Different"

[9] is considered a seminal article advocating for emergence. Similarly, R. Laughlin is another Nobel Laureate who advocates for emergent phenomena in his well-known paper “The Theory of Everything” [168] in which he argues that reducing physics to simple equations fails to reveal anything about important phenomena in solid-state physics. Anderson continues this line of thinking, “the main fallacy in this kind of thinking is that the reductionist hypothesis does not by any means imply a ‘constructionist’ one: The ability to reduce everything to simple fundamental laws does not imply the ability to start from those laws and reconstruct the universe” [9]. More generally, this is known at the fallacy of composition: the mistaken belief in a property of a part belonging to that of the whole. “The constructionist hypothesis breaks down when confronted with the twin difficulties of scale and complexity. The behavior of large and complex aggregates of elementary particles, it turns out, is not to be understood in terms of a simple extrapolation of the properties of a few particles. Instead, at each level of complexity entirely new properties appear, and the understanding of the new behaviors requires research which I think is as fundamental in its nature as any other” [9]. Bringing us full circle to the central idea of emergence, Anderson “the whole becomes not only more than but very different from the sum of its parts.” [9]

This reincarnated version of emergence is more nuanced and less strictly enforced than some of those seen during the British Emergentist period. Another Nobel Laureate, F. Crick accepts the scientific validity of emergence in this new form. He rejects the early type of thinking that said that “emergent behavior cannot in any way, even in principle, be understood as the combined behavior of its separate parts” [78, p. 11]. Instead, “the scientific meaning of emergent . . . assumes that, while the whole may not be the simple sum of its separate parts, its behavior can, at least in principle, be *understood* from the nature and behavior of its parts *plus* the knowledge of how all these parts interact” [78, p. 11].

## *A.2 Characterizing Emergence*

### A.2.1 Nonlinearity

The concept of an emergent, as opposed to a resultant, was common to most of the thinkers of the British Emergentist movement. This aspect sought to characterize how properties combine among constituent parts. While this distinction is no longer present in the contemporary discussion of emergence, it will be presented here for historical context. Furthermore, while the distinction between emergents and resultants no longer exists, they did provide some ideas that remain central to emergence today.

J.S. Mill was a British empiricist philosopher that proposed a framework using the principle of *composition of causes*, “the principle which is exemplified in all cases in which the joint effect of several causes is identical with the sum of their separate effects” [187, p. 267]. Although he did not use the term emergent, he called effects of causes that failed to obey the principle of additivity as “heteropathic” [187, p. 269]. For example, in Newtonian mechanics, the kinematics of a body with two simultaneously applied forces is equivalent to the linear superposition of the kinematics of the body with the forces acting separately. This system would obey the composition of causes. However, it should be made clear that even the composition of causes should not be taken for granted. As Broad says about the result of vector-sum additivity in Newtonian mechanics, “there is not the least possibility of deducing this law of composition from the laws of each force taken separately” [55, p. 62]. The principle of additivity should not be regarded as a default or most common principle—even simple additive laws must be determined through induction.

English philosopher G.H. Lewes is credited with coining the term emergent. Lewes’ “emergent” effects are similar to Mill’s “heteropathic” effects. Lewes advanced the idea that emergents and resultants were two classes of effects separated by the modes of their combination. In contrasting resultants with emergents, “each agent, indestructible and independent, has its own individual value; and the effect or combination of agents has two modes: in the one case we have an addition or mixture; in the other a combination, with an emergent” [171, p. 368]. “Every resultant is either a sum or a difference of the co-operant forces: their sum, when their directions are the same; their difference, when their directions are contrary” [171, p. 369]. The major claim here is that emergents are a result of nonadditive effects. Lewes also made a further distinction about emergents, “every resultant is clearly traceable in its components, because these are homogeneous and commensurable” [171, p. 369]. “It is otherwise with emergents, when, instead of adding measurable motion to measurable motion, or things of one kind to other individuals of their kind, there is a co-operation of things of unlike kinds. ... The emergent is unlike its components in so far as these are incommensurable, and it cannot be reduced either to their sum or their difference" [171, p. 369]. Emergents are characterized by nonlinear behavior and differences in scale of the underlying components. Nonlinearity means that the effects of causes are nonlinear with respect to their combination. The consequence of this leads means that "although each effect is the resultant of its components, the product of its factors, we cannot always trace the steps of the process, so as to see in the product the mode of operation of each factor. ... It arises out of the combined agencies, but in a form which does not display the agents in action" [171, p. 368]. This characterization of emergence is consistent with the Bedau's viewpoint of weak emergence [34], discussed later in this chapter. This view of emergence, although well ahead of its time, bears resemblance to the view of emergence as seen from the field of nonlinear dynamics and chaotic systems.

"The essential feature of ... a mechanistic interpretation is that it is in terms of resultant effects only, calculable by algebraical summation. ... Against such a mechanical interpretation such a mechanistic dogma emergent evolution rises in protest. The gist of its contention is that such an interpretation is quite inadequate. Resultants there are; but there is emergence also" [193, p. 8]. While Morgan was not advocating for an enlarged view for what it means for something to be mechanistically reducible, his argument makes it clear that if we limit ourselves to systems that can be described using only "algebraical summation" we are left with very few systems that meet this requirement. It should be noted that this distinction between resultants and emergents is quite simplistic and should not be regarded as the key distinguishing feature of emergence. While nonlinearity and multiscale behavior are indeed important properties of emergence, they should not be regarded as *the* defining feature. The goal of this section is to provide historical context. As aforementioned, the linear additivity property is by no means common and should not be considered the default principle of composition. Relatively few systems exhibit this type of additivity and the only way to establish this is through induction. If we regard anything that is not the result of summation as emergent, then we have not captured the essence of emergence.

A modern interpretation of the idea of a resultant effect is based on W.C. Wimsatt's aggregativity heuristics. An approach advocated by Wimsatt, known for his work in the field of philosophy of biology, is a formalization based on the idea of aggregativity. An aggregative property is similar to British Emergentists' concept of a "resultant"—an outcome that is simply an aggregation of its constituent parts. Wimsatt's aggregativity approach [254] evaluates decompositions and alternative descriptions of a system seeking to find whether descriptions are aggregative or not. Those systems which fail to be aggregative are considered emergent. A system property is an aggregative with respect to a decomposition and description if all of the following conditions are met [254]:

- **Intersubstitution**: Invariance of the system property under operations rearranging the parts in the system or interchanging any number of parts with a corresponding numbers of parts from a relevant equivalence class of parts.
- **Qualitative Similarity**: Qualitative similarity of the system property under addition or subtraction of parts.
- **Decomposition and Re-aggregation**: Invariance of the system property under operations involving decomposition and re-aggregation of parts.
- **Linearity**: There are no cooperative or inhibitory interactions among the parts of the system for this property.

Wimsatt proposes that these heuristics are used to guide the process when developing a theory that describes the system behavior. The system is manipulated by the transformations implied by the criteria in search of invariances. These invariances, Wimsatt claims, imply a system that is more 'natural' "because they provide simpler and less context-dependent regularities, theory, and mathematical models involving these aspects of their behavior" [254]. While Wimsatt advocates this approach for identifying emergent behavior, its usefulness towards that goal is much more limited and subtle. The suggested approach finds the complement of emergence—this is, systems that are not emergent. The implication is that the arrangements that failed to be aggregative are therefore emergent. However, as Wimsatt himself asserts, “very few system properties are aggregative, suggesting that emergence, defined as failure of aggregativity, is extremely common—the rule, rather than the exception” [254]. This means that the number of systems and decompositions of a system that this approach identifies is vastly overwhelmed by the remainder of the space of possibilities. Despite this limitation, I include this approach here as a possible stepping stone towards helping construct a theory about an emergent property. This desire is typical of the approach of science. An unexplainable or surprising phenomenon is identified, then a reductionist approach is taken to identify the underlying mechanisms. Once a candidate emergent property has been identified, it is natural to prove its emergent status and develop a theory which explicates the phenomenon. In order to prove that something emergent, it must be shown that each of the constituent components is necessary and that the system composition is both necessary and sufficient to result in a particular property. The aggregativity approach can be used to substantiate an emergent property by showing its dependence with respect to various transformations and decompositions of the system.

Contemporary definitions of emergence no longer rely on such a simplistic distinction. More generally, emergent properties are a result of *nonlinear* combination of effects. The nonlinearity enables *multiscale effects* where microlevel effects can affect higher level properties. Nonlinearity is a necessary condition for emergence [96] [220, p. 70].

### A.2.2 Irreducibility

One of the main reasons for the existence of the concept of emergence is for *scientific explanation*. Explanation seeks to identify theories that elucidate an observation. One of the primary methods of explanation is through the use of *reduction*, where an phenomenon is explained by referencing the behavior of the underlying constitutive components. An irreducible property is a one that cannot be deduced from the properties of its constituent parts. Irreducibility implies the existence of ontologically real properties and objects in the world. After all, if an object cannot be reduced to something else, then it must be real in its own right. This is the central philosophical battle concerning the relevance of emergence.

One of the core definitions of emergence deals with the irreducibility between levels that are part of the hierarchical structure of reality. Irreducibility and hierarchical levels are nonseparable concepts. If something is reducible, then it can be equivalently represented using a lower-level representation. On the other hand, if it is irreducible, then it cannot be explained or represented in terms of the lower-level features. This idea was common during the Emergentist movement, as well as in definitions of emergence used today. Understanding reducibility is the first step to understanding irreducibility. “A reductive explanation of a behavior or a property of a system is one showing it to be mechanistically explicable in terms of the properties of and interactions among the parts of the system” [254]. Furthermore, “the explanations are causal, but need not be deductive or involve laws” [254]. In a rigorous way, “a theory $T_h$ reduces to a lower-level theory $T_l$ if all the nomic claims made by $T_h$ can be explained using only the resources of $T_l$ and necessary truths” [29, p. 198]. This is very similar to the reduction framework proposed by Nagel [202] where “bridge laws” are used as supplementary premises to connect terms between adjacent levels when not all of the claims in $T_h$ can be explained using only the terms in $T_l$. The use of bridge laws necessarily implies irreducibility. Broad’s use of irreducible “trans-ordinal laws” to connect emergent properties between adjacent levels are a form of Nagel’s bridge laws. Alexander describes the irreducible relationship as a “higher quality emerges from the lower level of existence and has its roots therein, but it emerges therefrom, and it does not belong to that lower level, but constitutes its possessor a new order of existent with its special laws of behaviour. The existence of emergent qualities thus described is something to be noted . . . under the compulsion of brute empirical fact, or . . . to be accepted with the ‘natural piety’ of the investigator. It admits no explanation” [5, pp. 46–47].

During the Emergentist movement, Broad describes two theories of explanation. The first, the “Theory of Emergence,” describes a theory in which “the characteristic behaviour of the whole could not, even in theory, be deduced from the most complete knowledge of the behaviour of its components, taken separately or in other combinations, and of their proportions and arrangements in this whole” [55, p. 59]. The second, the “Mechanistic Theory,” in which “the characteristic behaviour of the whole is not only completely determined by the nature and arrangement of its components; in addition to this it is held that the behaviour of the whole could, in theory at least, be deduced from a sufficient knowledge of how the components behave in isolation or in other wholes of a simpler kind" [55, p. 59]. An important concept when determining if something is emergent or not is whether the properties of the higher level can be deduced from the properties of the lower level. In Broad's conception, the various aggregates of orders represent the hierarchical and multiscale structure of complex systems. Broad distinguishes between "intra-ordinal" and "trans-ordinal" laws that describe the behavior of each aggregation level and the relationships between adjacent orders, respectively. "A trans-ordinal law would be one which connects the properties of aggregates of adjacent orders" [55, pp. 77-78]. "An intra-ordinal law would be one which connects the properties of aggregates of the same order" [55, p. 78]. Properties of an order "which could in theory be deduced from the structure of the aggregate, the properties of its constituents, and certain laws of composition which have manifested themselves in lower orders" are called "reducible characteristics" [55, p. 78]. On the other hand, those properties which could not have been likewise deduced represent emergent properties which are described by emergent trans-ordinal laws.

One of the outcomes of this irreducibility is that emergents could only be identified through induction. Experimentation and observation are the fundamental method for gaining knowledge about a system—emergent properties must be discovered by empirical methods. Logically, this must be so: if emergents are defined in a way that makes deducibility impossible, experimentation is the *only* way to know the properties given a certain situation. "Were all effects simple resultants ... our deductive power would be almost absolute... It is precisely because effects are mostly emergents that Deduction is insecure, and Experience is requisite to confirm even the most plausible deductions" [171, p. 370]. Broad claims "we do not know all the properties of any element, and that there is always the possibility of their manifesting unpredictable properties when put into new situations" [55, p. 66]. As Mill explains regarding the chemical reaction of hydrogen and oxygen producing water (a canonical example used during this period), "no experimentation on oxygen and hydrogen separately, no knowledge of their laws, could have enabled us deductively to infer that they would produce water. We require a specific experiment on the two combined" [187, pp.

315–316].

The criterion for the impossibility of deduction is obviously troublesome. If new scientific laws and mathematical principles are discovered that allow us to deduce previously emergent phenomena, the concept of emergence loses its credibility. To get around this issue, contemporary interpretations of emergence, such as Bedau's weak emergence, do not require a strict notion of irreducibility. Rather, the irreducibility is interpreted as a computational incompressibility. Huneman [148] argues that the computational interpretation of emergence can be made consistent with the lack of deducibility in the sense of incompressible deduction.

### A.2.3 Novelty

The idea of *novelty* is a defining characteristic of emergence. However, there are different types of novelty, all of which are used in the discussion on emergence. According to Bunge, there are three types of novelty: conceptual, empirical, and ontological [57, p. 127]. I would add a fourth type of novelty that is relevant to the discussion of emergence: *observer-dependent novelty*. Conceptual novelty, the weakest form of emergence, is a type of irreducibility where "a natural kind in a higher-level theory is conceptually novel if there is no kind in any potential reducing theory that captures the same set of phenomena" [185, p. 204]. Conceptual novelty is the overcoming of irreducibility through the introduction of new theories and explanations. "The conceptual approach maintains that a system that has reached a critical level of complexity can be described effectively only by introducing a conceptual or descriptive apparatus that is new compared to what is used for more basic phenomena. This conceptual novelty can range from the invention of a new term to the introduction of an entirely new theory" [35, p. 10]. Conceptual novelty also has a range of interpretations. One end is rigorous and based on the sets of axioms needed to deduce the system's behavior. On the other end of the spectrum is the novelty due to a change in language or description of a system. While this is an intuitive concept and one that is common in the literature, it is also one fraught with problems. A discourse-based theory of emergence would be subjective and would greatly minimize the utility of emergence as a scientific concept. Empirical novelty is the first realization of a concept. For example, a new atomic element may be theorized; however, it is not until it is discovered in nature or created in a laboratory environment that it has attained empirical novelty. This interpretation of novelty is closest to the everyday usage of something being "new." Finally, there is ontological novelty. This type of novelty is concerned not with new concepts entering into scientific knowledge, but rather on whether a new physical entity or property has come into existence. The existence of examples of ontological novelty are debated; therefore, I will leave it to the philosophers of science. Finally, many discussions in emergence focus on observer-dependent novelty. The system acts under its governing dynamics without knowledge or concern that its behavior is emergent; however, its behavior is interpreted to be novel by an outside observer. "The 'newness' in each case is only heightened by the fact that the emergent feature stands in direct opposition to the systems' defining character: complete determinism underlies chaos and near-complete stochasticity, the orderliness of self-similarity" [82]. While this interpretation can be found in the literature on emergence, it is not one that I am advocating. Observer-dependence brings in a host of issues, none of which are necessary for emergence to stand on its own. This issue will be discussed in more detail later in this chapter.

As Bedau acknowledged about his conception of weak emergence, one vital piece missing from his formulation is the desire to identify interesting emergent phenomena. A useful and interesting definition of emergence must incorporate the concept that an emergent property represents a *qualitative change* in behavior. British biologist C.L. Morgan approaches emergence from the viewpoint of biological evolution. "The orderly sequence, historically viewed, appears to present, from time to time, something genuinely new" [193, p. 1]. This interpretation of emergence captures the more intuitive sense that emergence is not just nonlinear or irreducible behavior, but also represents *qualitative novelty.* "There may often be resultants without emergence; but there are no emergents that do not involve resultant effects also. Resultants give quantitative continuity which underlies new constitutive steps in emergence. And the emergent step, though it may seem more or less saltatory, is best regarded as a qualitative change of direction, or critical turning-point, in the course of events" [193, p. 5]. What we are really interested in identifying when studying emergence are those critical turning points that lead to new behaviors. Corning includes both of these conditions, “coherence” (Morgan’s quantitative continuity) and “radical novelty” (Morgan’s “something genuinely new”) as characteristics of emergence [76].

### A.2.4 Hierarchy

The idea of a hierarchy is common to the philosophy of emergence. Since emergence was initially used for scientific explanation (e.g., how chemical properties can be predicted from their constituent elements; or how life can be explained by chemical processes), the structure of emergence mirrors the structure of the sciences. There is a level that contains fundamental particles and is governed by fundamental laws. Aggregates of the fundamental particles form the next level higher of increasing complexity. Each level also includes a “special science” (e.g., physics, chemistry, biology, psychology) that governs the behavior of that level.

The two most important questions when dealing with hierarchies are 1) how levels are defined (i.e., how to establish boundaries), and 2) how levels are ordered (i.e., what does it mean for one level to be higher than another?). Bedau [35, pp. 5-6] provides three general characteristics of this hierarchical ordering of levels:

- the hierarchy of levels has no precisely defined order, but instead is determined implicitly by the organizational complexity of objects,
- each level is assumed to contain at least one kind of object and one kind of property that is not found below that level,
- at each level kinds exist that have novel causal powers that emerge from the organizational structure of material components.

This conception of the hierarchy is based on emergence—each level in the hierarchy is defined by the presence of emergent properties. Others, like Checkland have a similar definition, “there exists a hierarchy of levels of organization, each more complex than the one below, a level being characterized by emergent properties which do not exist at the lower level” [68]. This increasing level of complexity at each level is common in the understanding of the structure of the sciences. As Anderson writes, “with increasing complication at each stage, we go on up the hierarchy of the sciences" where each new level is characterized by "new types of behavior" [9]. Since each level is itself defined by emergent properties, there can be multiple levels of emergence [33].

#### *A.2.4.1 Supervenience*

Supervenience is a term of art used in the philosophy of emergence, but it has an intuitive and accepted interpretation in the sciences. It is a way of expressing determinism between levels in a hierarchy. Supervenience means that all properties of a system are only determined by its micro level [159, 158]. This has several consequences. The first is that every property is due to only its microlevel structure; there is nothing but the physical. Another consequence is that there cannot be a change in a higher level property without a corresponding change in the lower level. Supervenience establishes the causal determinism from the micro level to the macro level. Supervenience is required for reduction [184]. While supervenience may pose problems for some formulations of emergent properties [147], supervenience is quite accepted as the underlying mechanism for how micro-level properties give rise to higher-level emergent properties [33].

#### *A.2.4.2 Downward Causation*

While the micro-to-macro causal determinism specified by supervenience is generally accepted, the macro-to-micro causal effects, known as *downward causation*, are much more controversial. As Bedau writes, "strong emergence has a number of failings, all of which can be traced to strong downward causation" [34, p. 377]. This process, coined by Campbell [58], refers to macro-level phenomena having causal influence over the micro-level from which they arose [158]. Why is downward causation necessary? Emergent properties "are supposed to represent novel additions to the ontology of the world, and this could be so only if they bring with them genuinely new causal powers" [158]. Bedau similarly explains that "emergent phenomena without causal powers would be mere epiphenomena" [33]. O'Connor provides us with a distinction between macro-level and micro-level causal influence, "an emergent's causal influence is irreducible to that of the micro-properties on which it supervenes: it bears its influence in a direct 'downward' fashion, in contrast to the operation of a simple structural macro-property, whose causal influence occurs via the activity of the micro-properties which constitute it" [205].

If we accept supervenience, then downward causation implies that higher-level emergent phenomena would necessarily have to influence the micro-level properties that determine all of the properties in the world. Downward causation appears to be necessary for emergence. However, downward causation poses one important problem of logical coherence, known as the *exclusion argument*: "if an emergent, $M$, emerges from basal condition $P$, why can't $P$ displace $M$ as a cause of any putative effect of $M$?" [158]. In other words, "the micro-level explanation of the micro-level effects will preempt the macro-level explanation" [33]. Another problem caused by downward causation is that it "makes a difference only if it violates micro causal laws" [33]. If the exclusion argument is accepted, the only way for macro- and micro-level causal powers to coexist is if "the macro and micro causes conflict because micro causation would have brought about an incompatible micro effect so the downward causation would violate the fundamental micro laws" [33]. There are some that argue that this is, in fact, the case. Anderson argues "that matter will undergo mathematically sharp, singular 'phase transitions' to states in which the microscopic symmetries, and even the microscopic equations of motion, are in a sense violated" [9]. Obviously, this talk of violation of fundamental micro laws is troublesome.

I would argue that instead of claiming that micro laws are violated, rather the micro states take on a trajectory that is significantly different due to the presence of the macro-level cause. This brings about another problem, "how could these higher-level properties causally influence and alter the conditions from which they arise? [158, p. 25]. Kim addresses this concern by introducing the notion of a diachronic downward causation, meaning that cause and effects are separated by an increment in time that breaks the circular dependency. While this might be necessary to address the philosophical concerns, I believe that this simply describes the natural feedback loops that exist in nature. As Deacon explains, "emergent phenomena grow out of an amplification dynamic that can spontaneously develop in very large ensembles of interacting elements by virtue of the continuing circulation of interaction constraints and biases, which become expressed as system-wide characteristics" [93, p. 124].

Furthermore, in terms of causality, there is a "circular connectivity of causal dynamics, not a special form of causality" [93, p. 124]. This helps to alleviate concerns about strong emergence's requirement of new forms of causality. Circular connectivity means that there is simultaneous upward and downward causal influence. The idea of feedback is well known. If we look at the system from the viewpoint of feedback, then there is nothing inconsistent there—the state of a property can be a function of itself (e.g., the partial differential equations that describe many physical systems). If the emergent property exists then having a feedback mechanism poses no incoherence. The alternative is a degenerate case: a system in which property causes the elimination of the underlying emergence condition would not be emergent and would therefore not be instantiated. The properties that are observed are the ones that maintain the emergence condition during any downward causation; if the causal condition is inconsistent with the emergent property, then the emergent property is either transient or is never instantiated to begin with.

Boschetti and Gray [51] propose a "Turing test" for emergence that seeks to identify emergence based on the presence of causal downward influence. However, here causality is severely diminished to mean simply whether or not "by acting upon it, we can change the effects it produces" [51]. This test for emergence asks "whether a process empowered with autonomous causal emergent properties (a human) can discriminate between another causal emergent process and a computer program" [51, p. 358]. They propose three types of emergence, with increasing downward causal influence: pattern formation, intrinsic emergence, and causal emergence. Boschetti and Gray claim that pattern formation is not emergent because it is not possible to manipulate the patterns without directly acting on the constituent parts [51]. However, this claim that pattern formation is not emergent is not accepted. As I shall show later in this chapter, Bedau and others claim that pattern formation does constitute an autonomous effect with causal influence. In the intermediate form of emergence, intrinsic emergence, is borrowed from Crutchfield [82], where emergent effects confer additional functionality in the system. In the strongest form of emergence, "we can exert direct control without manipulating, nor concerning ourselves with, the lower-level constituents" [51]. This framework by Boschetti and Gray is clearly subject to interpretation, something that we would want to avoid. But like the Turing test that this framework takes inspiration from, their goal is to test whether or not an intelligent observer can distinguish between simulated causal influence and our conception of real causal influence: "what matters is the perception by which we believe we can exert causal control over the higher-level emergent features. Does it look as if those features possess causal control? Does it look like they do more than the limited number of behaviours purposely encoded in the local rules? Do system entities behave as if they were autonomously interacting with external processes and respond accordingly?" [51].

#### *A.2.4.3 Alternative to the hierarchical approach*

Ryan proposes a method based on scope, resolution, and state instead of one based on a hierarchical structure [220]. In Ryan's framework, the modeling of emergence can be described by the equation $M = \mathcal{C}(\mu)$, where $M$ is the macrostate, $\mu$ is the microstate, and $\mathcal{C}$ is the hidden mapping from microstate to macrostate. $\mathcal{C} : M_{\mathrm{\mu}} \rightarrow M_M$, the map between coarse-grained macrostate $M$ and the microstate $\mu$. If an emergent property$\rho_i$ exists at the macrostate, it also exists at the microstate via the mapping $\mathcal{C}$. In this formulation, "a property is a novel emergent property iff it is present in a macrostate but it is not present in any microstate, where the microstates differ from the macrostate only in scope" [220, p. 72]. The difference in scope can be established by determining the minimal macrostate $M^*$, which is the smallest possible scope that still exhibits an emergent property. This definition implies the existence of a critical scope that demarcates emergence. Scope defines a temporal-spatial boundary, while resolution is the finest temporal-spatial distinction between two alternative system configurations [219, p. 121]. A higher resolution distinguishes between a larger number of possible states. The scope defines the boundary between the system and its environment [220, p. 69]. There are a few problems with this approach. The first is that, by eliminating the concept of hierarchical levels, Ryan has thrown out many of the concepts that have been used for centuries and that are common to the understanding of the structure of the sciences. In its place, Ryan introduced an observer-dependent property, "resolution." Furthermore, scope and resolution are not strictly independent.

### A.2.5 Strong versus Weak Emergence

It should be clear at this point that it is difficult to define emergence in a logically coherent manner. This has caused many to propose a “weak” sense of emergence that helps alleviate some of the incoherence—however, the existence of a “weak” form also implies that there is a “strong” form. The distinction between these

#### *A.2.5.1 Strong Emergence*

Strong emergence is the strictest conception of emergence. It is the type of emergence conceived by applying the definition of emergence including “the requirement that emergent properties are supervenient properties with irreducible causal powers” [33]. The inclusion of irreducible downward causal influence is the defining feature of strong emergence. However, it should be noted that the lack of a universally accepted definition of emergence also means that the conception of weak and strong emergence lie on a spectrum of interpretations and strictness. Also, it is clear from the literature that some writers use the weak/strong distinction to refer to a different condition: the degree of difficulty in the reducibility between the macro and micro levels effects. In this alternate form of emergence, strong emergence refers to processes where irreducibility is impossible even in principle, while a weak form of emergence would refer to processes that are straightforward to reduce the macro-level effects to the micro-level effects with hindsight. There is similarity in both of these types of weak/strong emergence in that they both imply degrees of irreducibility, but the former type emphasizes the irreducibility of the causal effects, while the latter emphasizes the relationship between macro and microlevel effects without necessarily establishing causality. In this section, I’ll primarily focus on the first type of strong emergence.

O’Connor, coming from the perspective of analytic philosophy, provides a definition of strong emergence: “Property $P$ is an emergent property of a (mereologically-complex) object $O$ iff $P$ supervenes on properties of the parts of $O$, $P$ is not had by any of the object’s parts, $P$ is distinct from any structural property of $O$, and $P$ has a direct (‘downward’) determinative influence on the pattern of behavior involving $O$’s parts” [205]. Bar-Yam writes of strong emergence, “properties that are unique to the collective—cannot be identified through any observations of the parts, and is counter to the conventional perspective that parts determine the behavior of the whole. In the case of strong emergence, it is possible that the properties of the whole determine the behavior of the parts" [24].

"Although strong emergence is logically possible, it is uncomfortably like magic. How does an irreducible but supervenient downward causal power arise, since by definition it cannot be due to the aggregation of the micro-level potentialities? Such causal powers would be quite unlike anything within our scientific ken. This not only indicates how they will discomfort reasonable forms of materialism. Their mysteriousness will only heighten the traditional worry that emergence entails illegitimately getting something from nothing" [34, p. 377]. More so, "the most disappointing aspect of strong emergence is its apparent scientific irrelevance" [34, p. 377]. Despite all of the philosophical literature on the subject of emergence, there is little scientific evidence that uses strong emergence as the basis of understanding. Consciousness represents one of the most compelling and often cited arguments for downward causation. Chalmers argues that consciousness is the only clear case of strong emergence [65, p. 246] and quantum mechanics is the best example of strong downward causation [65, p. 249]. Sperry, a Nobel laureate who studied neurobiology and neuropsychology writes, "the conscious properties of cerebral patterns are directly dependent on the action of the component neural elements. Thus, a mutual interdependence is recognized between the sustaining physico-chemical processes and the enveloping conscious qualities. The neuro-physiology, in other words, controls the mental effects, and the mental properties in turn control the neurophysiology" [236, p. 534]. However, Bedau criticizes the use of an outdated Polanyi and Sperry paper in most claims of evidence of strong emergence. I would say that we should avoid the use of consciousness as a scientific example of strong emergence since, at this time, we know little about how consciousness is related to brain function. I think it is flawed to use a concept that we understand so little of to use it as an example of a type of behavior. As was the case with quantum mechanics explaining "emergent" chemical properties, it is quite possible that a new theory of neurobiology can lead to improved theories of consciousness that does not rely on strong emergence and downward causation. Emergence would be better served by examples that are better understood so that we can understand what emergence means. On the other hand, consciousness is one of the holy grails of science and philosophy and seems to be the ideal candidate for what constitutes a complex system.

But strong emergence with downward causation is hardly a discredited philosophy. There are many that argue for its scientific relevance in explaining behavior. Bar-Yam advocates for the existence and importance of strong emergence and claims that weaker forms of emergence do not capture all collective behavior [24]. Bar-Yam explains how downward causation might work, "when a system is faced with global constraints, the properties of an entire system may determine the properties of a part, without the properties of a part determining the properties of the whole system. ... Only when constraints are defined that act on collectives and not on components does strong emergence occur" [24]. Batterman similarly argues that phase transitions in materials are examples of genuine physical discontinuities in nature that are "ontologically irreducible to any theory of its parts" [29, p. 214]. Menon and Callender [185] explain the difficulties of explaining transition phenomenon using statistical mechanics—the sine qua non of studying the thermodynamics of physical systems. Developing a theory for phase transitions using statistical mechanics, including renormalization group theory, relies on the assumption of infinite sized systems [185, p. 197]. Once finite systems are considered, the theories break down; the result of this is that many people believe that phase transitions are one of the best examples of truly emergent behavior [185, p. 198].

#### *A.2.5.2 Weak Emergence*

Bedau proposes a concept he called *weak emergence* that attempts to get around the metaphysical conundrum of downward causation in traditional theories of emergence [34, 33]. Why does weak downward causation not violate causality like strong downward causation does? A weak macro cause is nothing more than the aggregative effect of micro causes, therefore there is no violation of micro causal laws. For the same reason, the "exclusion" argument does not apply since the micro and macrolevel causes are equivalent. Weak downward causation is diachronic, which eliminates the circular causality argument between levels. Bedau gives two necessary (but not sufficient) conditions for emergence that he calls “hallmarks of emergent phenomena” [34, p. 375]. The first condition is roughly in line with the concept of supervenience. The second condition is a weak form of downward causation; instead of claiming causality, it simply calls the phenomena autonomous. Together, these can be used to define any form emergence:

1. Emergent phenomena are constituted by, and generated from, underlying processes

2. Emergent phenomena are autonomous from underlying processes

In weak emergence, all of the information we would need to describe the collective behavior is present in the microstate: the state of each element and their interactions with each other and the environment—no macrolevel information is needed to determine the outcome of the system. As Bedau explains, “the system’s global behavior derives just from the operation of micro-level processes, but the micro-level interactions are interwoven in such a complicated network that the global behavior has no simple explanation. The central idea behind weak emergence is that emergent causal powers can be derived from micro-level information but only in a certain complex way” [33]. This “certain complex way” refers to “the complex consequences of myriad non-linear and context-dependent micro-level interactions” [33]. Bedau believes that weak emergence is sufficient to explain virtually all emergent behavior, so his definition is more general. On the other hand, Bar-Yam argues that even complete knowledge of microstates is not always sufficient to determine collective behavior, leading to the claim that strong emergence is a possibility for systems that have macro-level constraints. Bar-Yam takes a slightly different perspective when defining weak emergence. For him, weak emergence describes a system in which “collective behaviors of the system cannot be readily recognized because it is difficult to extract them from the large amount of information present in the fine scale microscopic view” [24]. In other words, “the relationship of microscopic and macroscopic views of a system that differ only in precision” [24, p. 17]. For Bar-Yam, weak emergence happens in systems with overwhelming intricate behavior at the microlevel but there does not exert downward causal influence from the macrolevel back to the microlevel.

According to Bedau, all weak emergence at least have an epistemic portion to macro explanatory autonomy. This is due to the large number of aggregate interactions and effects that we could only calculate using simulation. The more important question is whether weak emergence also has sufficient autonomy such that the underlying explanation is not merely the aggregation of the micro effects. In order to have macro-level explanatory autonomy, there would need to be "distinctive objective structures in reality" [33]. To demonstrate that it would be possible for macro-level autonomy, Bedau argues that multiply-realizable elements (e.g., glider guns in Conway's Game of Life) are autonomous from the micro-level. Every macro-level effect (a glider stream) are explained by macro-level causes (a glider gun). Furthermore, the macro-level structure is multiply-realizable since any number of micro-histories can lead to the macro-level glider gun. "But weak emergent phenomena that would be realized in an indefinite variety of different micro contingencies can instantiate robust macro regularities that can be described and explained only at the macro level. The point is not just that macro explanation and description is irreducible, but that this irreducibility signals the existence of an objective macro structure. This kind of robust weak emergence reveals something about reality, not just about how we describe or explain it. So the autonomy of this robust weak emergence is ontological, not merely epistemological" [33]. Furthermore, "if those micro histories had been different, the macro explanation could still have been true. The macro explanation is autonomous from the aggregate micro explanation" [33].

Bedau gives waves, vortices, and traffic jams as three examples of systems that exhibit weak emergence including weak downward causation [33]. Taking the example of a water wave, we intuitively recognize a wave as a coherent macro-level structure. We also understand that a wave (macro-level) comprises molecules of water (micro-level), so we know that the aggregative behavior of the individual water molecules constitute the wave. The wave has autonomy in that the wave retains its coherence regardless of which water molecules enter or leave the system. It also has weak downward causation in that the wave will entrain water molecules that constitute the wave as it moves. Bedau claims that weak emergence qualifies to be emergent since it meets both "hallmarks of emergence" criteria given earlier [33].

Darley proposed a definition for emergence, "a true emergent phenomenon is one for which the optimal means of prediction is simulation" [86]. Bedau's definition of weak emergence closely mirrors Darley's definition. A macrostate of a system with a microdynamic is weakly emergent if and only if the macrostate can be derived from the microdynamics and the system's external conditions only by simulation [34, p. 378]. However, a more general sense of "simulation" is meant here than simply the execution of a computer model. "A derivation by simulation involves the temporal iteration of the spatial aggregation of local causal interactions among micro elements" [33]. For practical purposes, certainly the use of a algorithm on a digital computer is implied; however, to avoid practical limitations in storage and computational power, a simulation using a Laplacian supercomputer (based on the idea of Laplace's demon, a perfect observer with infinite computing capabilities) is also sometimes used for thought-experiments. "The behavior of weakly emergent systems cannot be determined by any computation that is essentially simpler than the intrinsic natural computational process by which the system's behavior is generated" [33]. Weak emergent systems are computationally irreducible by definition, a concept which will be explored in more depth in the chapter on simulation.

Simon writes that weak emergence allows "reductionism in principle even though it is not easy ... to infer rigorously the properties of the whole from knowledge of the properties of the parts" [234]. If reductionism is possible in principle, it would seem that reversing the reductive relations would enable prediction. Bedau argues that "since weakly emergent properties can be derived (via simulation) from complete knowledge of micro-level information, from that information they can be predicted, at least in principle. If we have been observing a simulation of some system $S$ and at time $t$ we saw that $S$ was in state $P$, then we know that there is an appropriate derivation that $S$ will be in macrostate $P$ at $t$" [34, p. 393]. "In principle we can derive the system's behavior because we can simulate the system and observe its behavior for as long as necessary. And if we can derive how the system will behave, we can predict its future behavior with complete certainty" [34, p. 393]. However, to me, it seems that this conception of "prediction" is not really prediction at all. A prediction is a forecast of a future event that has not been observed. Therefore, we cannot simulate the system up to time $t$ and claim that this process is identical to a prediction. The same mechanism that made it impossible for us to predict any outcome before the start of the simulation still exists between the current time step and the next one at which we seek to make a prediction.

The second (and the more interesting and practical) concern with Bedau's weak emergence is that it does not point to any methodology (other than the use of simulation) of how to find and understand the phenomena that we're most interested in. Bedau acknowledges that "not all weak emergence is metaphysically or scientifically significant. In some quarters emergence per se is treated as a metaphysically significant category that signals a qualitative difference in the world" [33]. Emergence is "not a special, intrinsically interesting property; rather, it is widespread, the rule rather than the exception" [34, p. 394]. Chalmers attempts to combat this problem by defining weak emergence as "an interesting property that is unexpected, given the underlying principles governing the system" [65, p. 253]. However, this definition is dependent on an observer for whom the property is interesting. Therefore, according to Chalmer's view, an observer would use two criteria for classifying a property as emergent: 1) how interesting the property is, 2) how difficult it is to deduce (obviousness). Bedau recognizes that a "central challenge in complexity science is to identify and study those exceptional, especially interesting weak emergent macrostates that reflect fundamental aspects of complex systems and are amenable to empirical investigation" [34, p. 394]. This is the motivation for my work in this thesis. The goal is to find the phenomena that are both emergent and scientifically significant and interesting. To do this, we will have to include a definition of emergence that includes a signal of "a qualitative difference in the world." Overall, I would agree with Bedau that weak emergence is the most appropriate conception of emergence to adopt. Even without strong emergence, weak emergence (with the addition of looking for interesting examples) is compelling enough and provides enough of a framework for studying emergence.

### A.2.6 Ontological versus Epistemological Emergence

One of the most divisive and irreconcilable differences in the viewpoints of emergence is a fundamental question about whether emergence represents ontological novelty or is a result of an imperfect observer. While it appears to be a rather subtle question, it is central to the debate about the existence and consequences of emergence. Stated in philosophical terms, the fundamental question is whether emergence is an ontological or epistemological issue. This leads to two main questions, 1) what does it mean for something to be an ontological or epistemological problem, and 2) what are the consequences if the problem is ontological versus epistemological? Some might argue that this debate is of no consequence outside of the philosophy of science. What are the consequences if emergence is ontological instead of epistemological? I think this debate is important because it can help us understand how to approach solving the problem of emergence. If emergence is epistemological, then emergence has to be understood as a consequence of observer dependence and limited information. The introduction of an observer necessarily leads to the problem of observations and subjectivity. On the other hand, if emergence is ontological, then it is not clear that new phenomenological laws and more accurate knowledge of the state of a system will always help us to address emergent phenomena.

This idea was also present during the Emergentist period. Most of the Emergentists had an ontological view of emergence; however, others had a weakened sense or even an epistemological view of emergence. Early Emergentists, especially Mill and Broad, had a fairly strong ontological notion, wherein emergence had primitive causal powers that exist in addition to those of the lower-level [206]. Broad makes it explicit that the unpredictability of emergence is not a result of perceptual limitations—even a perfect observer would be unable to deduce the resulting properties. “If the emergent theory of chemical compounds be true, a mathematical archangel, gifted with the further power of perceiving the microscopic structure of atoms as easily as we can perceive hay-stacks, could no more predict the behaviour of silver or of chlorine or the properties of silver-chloride without having observed samples of those substances than we can at present. And he could no more deduce the rest of the properties of a chemical element or compound from a selection of its properties than we can” [55, pp. 70–71]. Alexander had a weaker ontological sense in which emergent qualities were metaphysically primitive but did not alter the fundamental physical dynamics [206]. At the other end of the spectrum, Lewes implies emergents can be temporary and epistemological: “To fill up this gap in our knowledge by the word ‘power,’ or ‘causal link,’ is illusory. Some day, perhaps, we shall be able to express the unseen process in a mathematical formula; till then we must regard the [combination of hydrogen and oxygen to form] water as an emergent" [171, p. 370, bracketed mine]. Although this was not a major claim by Lewes, this view of emergence was one of the first in the line of claims that emergence is theory-based and subjective. This provisional view is neither necessary nor productive, as I will explain later in this chapter.

Hempel and Oppenheim argue that "emergence of a characteristic is not an ontological trait inherent in some phenomena; rather it is indicative of the scope of our knowledge at a given time; thus it has no absolute, but a relative character; and what is emergent with respect to the theories available today may lose its emergent status tomorrow" [140]. It then seems that the determining factor of whether something is emergent or not is whether we have a working micro-theory. The history of physics, chemistry, and biology shows that certain phenomena that were once considered emergent became not once the underlying micro-theory was discovered. This supports Hempel's assertion that emergence is relative and will eventually be eliminated as a concept. Is this the end of emergence then? Is every system headed towards being deducible, not just in principle, but also in practice? I want to argue that unpredictability is, in fact, an ontological issue when dealing with complex systems. Bedau argues that the epistemological issues associated with emergence are a byproduct of the ontological structure of emergence. "Some people view the form of emergence associated with the denial of reduction in practice as essentially limited to this epistemological import, reflecting just the need for an epistemological crutch when explaining derivative phenomena. Others view this epistemological feature of scientific emergence as at least sometimes the consequence of a distinctive ontological structure in derivative phenomena, and therefore as reflecting a distinctive ontology for those phenomena" [35, p. 214]. The latter viewpoint means that epistemological limitations are the consequence of a deeper ontological cause.

An important distinction should always be remembered: unpredictability is a consequence of emergence and not vice-versa. While some conceptions of unpredictability, such as the notion of surprise or unexpectedness, are subjective (and rely on an underlying model that establishes an expectation), there are other definitions of unpredictability that are objective and rigorous. These latter concepts are much more useful tools because they eliminate the inherent subjectivity of the former concepts. In particular, concepts like chaos and computation irreducibility are potentially inherent properties of certain complex systems. In describing how chaotic behavior in deterministic systems leads to unpredictability, Crutchfield says, "the effective dynamic, which maps from initial conditions to states at a later time, becomes so complicated that an observer can neither measure the system accurately enough nor compute with sufficient power to predict the future behavior when given an initial condition. The emergence of disorder here is the product of both the complicated behavior of nonlinear dynamical systems and the limitations of the observer" [79]. It seems that emergence has elements of both ontological and epistemological characteristics. It should be emphasized that even though Crutchfield says that unpredictability is partly due to observer limitations, this should not be interpreted as supporting an exclusively subjective view of emergence. This type of observer dependence is due to both practical limitations and due to fundamental limits, e.g. the limits given by the Heisenberg Uncertainty Principle.

Ronald et al. [216] argue that emergence is observer dependent. They propose an "emergence test" wherein there exists a designer and observer (who could be the same entity as the designer). The designer constructs a system in language $L1$ describing simple and local interactions between components. The observer is aware of the system rules written in $L1$; however, the role of the observer is to describe the behavior of the system in a language $L2$. The goal of the observer is to reconcile the "cognitive dissonance between the observer's mental image of the system's design stated in $L1$ and his contemporaneous observation of the system's behavior stated in $L2$" [216, p. 228]. The harder it is for the observer to bridge the gap between $L2$ and $L1$, the more surprising the observation and the more likely it is emergent. It is easy to understand why an approach like this would be advocated; simple rules leading to intricate patterns and behavior is one of the motivating examples of emergent behavior. This especially true since Ronald et al. were approaching the problem from the Artificial Life (ALife) field. ALife seems to be particularly prone to the definition that emergence is observer-dependent. On the other hand, emergence from the world of physics and biology tends to be much more ontological in nature.

I, along with many others, feel that this observer- and language-dependent approach is troublesome and unnecessary for the study of emergence. This approach has numerous problems. The first is that it trivializes emergence to a lack of “sophistication of the observer” [216, p. 230–231]. This makes emergence a useless concept. It makes any example of emergence as theory-laden and introduces an infinite number of unknowable assumptions. We have to infer the sophistication of the designer, their intent, the internal model of the final working system, and their experiences in order to determine if there is a difference between the observed and intended effects. For scientific work, this approach is a non-starter. Other problems with Ronald et al.'s approach include that they believe that “artificial [should be] treated differently from the natural” [216, p. 227]. They argue that their test for emergence should only apply to artificial systems. I think that emergence should be the same phenomena in both artificial and natural domains. I see no reason to branch the study of emergence into natural and artificial systems. The concept and most examples are born out of natural systems. The goal should be to unify the concept. Surely, Ronald's approach does not apply to natural systems, since that would imply the need to apply teleological arguments. Teleology implies that every entity has a *designed* purpose, which is a position that can not be rigorously sustained outside of philosophy. Regardless of the difficulty of applying Ronald's approach to natural systems, if we are to motivate our study of emergence with examples in natural systems, then clearly we should develop a theory of emergence that is equally valid to natural and artificial systems. Furthermore, Ronald et al. try to justify the existence of the observer by appealing to the language used by other authors [216, p. 235]. For instance, Holland's use of the term “direct inspection” or Gell-Mann's use of the term “apparently complex.” I am not convinced that the original authors intended so much to be read into these use of these words. These are not scientific terms. Inspection is often used in the mathematical field; however, there is no subjectivity implied in this context: something can be solved by inspection if it fits a pattern for which a solution is known. Similarly, I think Ronald et al. are misreading Holland's words. Certainly some of the words that Holland uses to describe emergence can be interpreted to imply the action of a person. But there is no mistaking Holland's position that he does not believe that emergence is observer dependent: "It is tempting to take the inability to anticipate—surprise—as a critical aspect of emergence. It is true that surprise, occasioned by the antics of a rule-based system, is often a useful psychological guide, directing attention to emergent phenomena. However, I do not look upon surprise as an essential element in staking out the territory. In short, I do not think emergence is an 'eye-of-the-beholder' phenomenon that goes away once it is understood" [143, p. 5]. Clearly, Ronald et. al are using Holland's quote out of context and without a thorough reading of his position! In addition to these problems, even one of the authors of this paper has distanced himself from approach advocated [35, p. 302]. Kubik also criticizes Ronald et al.'s idea about surprise, "We think that judging the behavior of complex systems on the basis of our subjective feeling of surprise is misleading and obscures better explanations" [165]. In a more general rejection of observer dependence, Corning writes, "Must the synergies be perceived/observed in order to qualify as emergent effects, as some theorists claim? Most emphatically not. The synergies associated with emergence are real and measurable, even if nobody is there to observe them" [76].

Ryan proposes a modified definition of weak emergence, "a property is weakly emergent iff it is present in a macrostate but it is not apparent in the microstate, and this macrostate differs from the microstate only in resolution. A weak emergent property is a limitation of the observer, not a property of the system" [220, p. 71]. Unlike Bedau, Ryan claims that weak emergence is epistemic. Ryan claims that once the mapping between macro- and micro-states has been discovered, a property can no longer be considered emergent [220, p. 71]. I do not agree with this conclusion. If the only reason the mapping is known is through a computationally irreducible process, then clearly the mapping is identically equal to the simulation. If this is the case, then the mapping provides no new information. In information theoretic terms, the mapping contains no new information (i.e., reduction uncertainty). If uncertainty due to observational limitations is the source of uncertainty about the outcome, then the emergence is epistemological. However, Bedau makes it explicitly clear that weak emergence is not an epistemological notion. "Underivability without simulation is a purely formal notion concerning the existence and nonexistence of certain kinds of derivations of macrostates from a system's underlying dynamic" [34, p. 379]. Bedau claims that even a Laplacian supercomputer could not derive weakly emergent properties except by simulation [33]. Ryan claims that "if resolution is the only difference between a macrostate and microstate, no property of the macrostate can be emergent from the microstate" [220, p. 71]. Resolution is an observer dependent property and naturally leads to all of the complications that observer-dependence implies. I believe the goal should be to formulate emergence in a way that avoids unnecessary references to observer-dependent properties.

While establishing whether emergence is ontological or epistemological is important to build a coherent definition of emergence, for practical purposes, the distinction may not be critical. As Broad writes, "it matters little whether we ascribe this to the existence of innumerable 'latent' properties in each element, each of which is manifested only in the presence of a certain other element; or to the lack of any general principle of composition" [55, pp. 66–67]. "Whatever the ultimate truth of the matter may be, both the chemist and the physiologist are forced in practice to behave as if the complexes with which they deal had emergent properties" [55, p. 76].

### A.2.7 Nonseparability

The idea of nonseparability goes back to the core ideas of emergence as defined by the British Emergentists. Nonseparability is often used to describe the relationships between entities within the same level. Nonseparability means that spatiotemporally separating the constituent parts in any way would lead to a qualitatively different system (see Healy [139] for more discussion on separability). While nonseparability seems to imply a binary distinction, in practice, there can often exist a spectrum of separability. Separability describes the amount of coupling between two entities and the degree of uncertainty when the coupling is broken. In the field of aerospace engineering, the coupling between the structural and aerodynamic effects is known as aeroelasticity. However, while the combined effects are important, engineering practice has shown that the structure and aerodynamics can be analyzed separately and the design can be converged through an iterative approach. In this case, since the coupling effects do not dominate the solution and an iterative approach can be used, this example of aeroelasticity would not be considered strictly nonseparable. In other cases, the state of one entity is dominated by the state of another entity in the system. One well known example of nonseparable states is quantum entanglement. In a quantum entanglement "the composite system can be in a pure state when the component systems are not, and the state of one component cannot be completely specified without reference to the state of the other component. Furthermore, the state of the compound system determines the states of the constituents, but not vice versa" [147]. An example of this type of nonseparability in quantum states that leads to directly observable phenomena are phase transitions that lead to superconductivity and superfluidity in helium [147]. This central issue of separability is the degree of interactions with other components in the system. Systems with minimal or no interaction between components are easily analyzed separately. On the other hand, some systems are *dominated* by the interactions. Corning recognizes that modern ideas about "reductionism" place a lot more emphasis on the interactions in a system instead of simply looking at the individual constituent components. He goes on to argue that the line between systems science and reductionism have blurred. Indeed, it seems that reductionism and emergentism are converging towards a more singular concept that recognizes the importance of each approach. Corning recognizes that "the interactions among the parts may be far more important to the understanding of how a system works than the nature of the parts alone" [76]. In other words, we can say that emergence is brought about in systems where the behavior is dominated by interaction effects.

## *A.3 Concluding Remarks on the Philosophy of Emergence*

As Goldstein points out, perhaps we should not look to emergence as an "explanation but rather as a descriptive term pointing to the patterns, structures or properties that are exhibited on the macro-scale" [128, p. 58]. It is clear that attempting to attribute novel causal powers to emergent effects poses irreconcilable theories of explanation. Furthermore, emergence should not be viewed as strict distinction. There exists a wide range of systems with various degrees of emergence. Darley argues that there is no discontinuous separation between emergence and non-emergence [86]. Emergent systems should be considered as a class of systems at one end of a spectrum. At one end of the spectrum are systems where the state of the system can be predicted for all time (e.g., two-bodies subject to Newton's laws) while the other end of the spectrum has systems where prediction beyond one increment of time or space is exceedingly difficult.

While it is possible that some phenomena that were once regarded as emergent less mysterious as science produces laws and models that can be used for predictions, there is the likely possibility that it will never be possible or practical to deduce systems that are primarily governed by human and social behavior. Similarly, it may remain impossible or impractical to deduce systems that are dominated by chaos or stochasticity. In other words, while something may be deducible *in principle*, an emergence viewpoint might serve as a practical framework to understand the system. As long as there are practical limits to understanding system behavior, these techniques will remain useful.

Another important philosophical question is whether emergence is rare or ubiquitous. The answer will often depend on the viewpoint used when defining emergence. The consequence of this choice will dictate the relevance and usefulness of the entire concept of emergence. If you accept the view of weak emergence, then emergence is common with many examples of systems exhibiting emergence in the weak sense. However, if you are of the position that emergence only exists in the strong form then there will be likely few, if any, examples of emergent systems. At the other extreme, if emergence is extremely common then it is loses its utility and interestingness as a concept. We should aim to find a middle ground, one where the ideas of emergence can be used to help us understand a class of systems that is of interest to the scientific community. Bedau argues that "the most important goal should be to show that emergent properties are useful in empirical science… A defense of emergence will be secure only if emergence is more than merely a philosophical curiosity; it must be shown to be a central and constructive player in our understanding of the natural world" [34, p. 376]. Towards this goal, I hope to show that emergence is a useful concept not in just understanding the natural world, but also engineering complex systems.

# APPENDIX B

# INFORMATION THEORY

**Chapter Road Map**: The goal of this chapter is to provide background on the use of information theory within the study of complex systems. The background and results in this chapter will be used throughout this dissertation.

## *B.1 Information Theory*

The use of information theory when analyzing complex systems is not new. Ashby laid out the connections between cybernetic systems and information theory in his 1956 work An Introduction to Cybernetics [11]. Since then, other researchers have continued in this tradition in advocating or using information theory as a natural application of information theory in analyzing complex systems. Krippendorff argues that "entropy measures provide access to a rich source of data for the construction of theories in which variety, diversity, and differentiation are the target of generalizations" [163]. Entropy is the "irreducible complexity" of a random process [77, p. 1]. We can see the overlap in the language used to describe emergence and complexity and the language of information theory. The connection between information theory and statistical mechanics is remarkably strong. Shortly after Shannon formalized information theory, Brillouin showed the equivalence of information theory and statistical mechanics [241]. Jaynes [152, 153] showed that information theory could be used to derive the laws of statistical mechanics.

Information theory remains one of the dominant approaches in studying complex systems. This include the work of Shalizi [229], Ryan [219], Prokopenko et al. [212], and many others. Many use information theory to *measure the complexity* in a system. However, this is not my goal in this thesis. There does not seem to be a clear connection between complexity measures and emergence. There is neither a "critical" level of complexity that yields emergence nor is there a problem-independent case to be made that higher measures of complexity yield emergence. Instead, information theory is used here because it gives us the methods to measure changes in probabilistic outcomes. Information theory provides a much more robust measure of the variability of a distribution than simpler statistical measures like variance [175].

Information theory is also a useful tool because it is a universal prediction method that requires no model to make predictions [229, pp. 23–24]. This model-free approach can be applied to a large range of systems without having to validate a model for the system first. Even more fundamentally, the lack of a model to predict or explain system behavior is one of the common features of emergence and complex systems.

## *B.2 Uncertainty, Information, and Entropy*

Information and uncertainty are fundamentally linked concepts. In the simplest definition, information is the change in uncertainty regarding a set of outcomes. A message that does not change our uncertainty about an outcome contains no information. At the other extreme, the most amount of information comes about when all uncertainty is removed and there exists a single certain outcome. As Wiener wrote, "the transmission of information is impossible save as a transmission of alternatives" [253, p. 10]. While much of information theory is couched in communication theory and transmitted messages, the concept of a message can be generalized to include any process where there is a transmission of information, including the performing of experiments. In fact, in 1953, Leon Brillouin argued that information theory applies equally to an observer in the laboratory receiving messages from his instruments about the outcome of the experiment [241]. Cover and Thomas define a communication channel as "a system in which the output depends probabilistically on its input" [77, p. 7]—a definition which can apply to almost any system we wish to study. Regardless if we are studying real world systems or systems in a controlled experiment, information theory applies. According to Nicolis [204, p. 186], information requires two fundamental conditions:

1. A sharp symmetry breaking in space, owing to which other possible issues of the reading process are continuously eliminated.

2. An element of unpredictability, associated with the revealing of an object or a message that the reader could not infer to begin with.

Information theory is typically formulated in discrete terms. While information theory can be generalized to a continuous form [149], I will formulate my thesis in the discrete form for several reasons. The first is that the discrete form is better behaved with many desirable properties (discussed later in this chapter). Additionally, any measurements of the real world, and especially computer simulations, are inherently discrete. Crutchfield argues that "due to the inherent limitations of scientific instruments, all an observer can know of a process in nature is a discrete-time, discrete-space series of measurements" [80]. Similarly, Badii argues that discretization (i.e., symbolic representation) of physical systems does not pose any problems; in fact, many natural systems can be easily and usefully represented discretely. Similarly, Nicolis and Prigogine argue that most physical processes are fundamentally discrete [204, pp. 159–160].

### B.2.1 Discrete Uncertainty and Information

Uncertainty is a measure of the logical variety in a system. Uncertainty $U$ is defined by the base-2 logarithm of the number $N$ of options available with reference to some variable $\mathbf{X}$.

$$U\left(\mathbf{X}\right) = \log_2 N_{\mathbf{X}}$$

The amount of *information* of message $x$ out of the set of all possible messages $\mathbf{X}$ is the difference in uncertainty before and after the message.

$$I\left(x \in \mathbf{X}\right) = U\left(\mathbf{X}\right) - U\left(x\right) = \log_2 N_{\mathbf{X}} - \log_2 N_x = -\log_2 \frac{N_x}{N_{\mathbf{X}}} = -\log_2 p\left(x\right)$$

### B.2.2 Entropy

Entropy is a measure of the uncertainty in a probability distribution. It is a measure of "how much freedom one is given in the selection of an event, or how uncertain the outcome is, or how difficult to predict the outcome" [149, p. 5]. More concretely, entropy measures the number of bits on average required to describe the random variable [77, p. 5].

The entropy $H$ of a random variable $\mathbf{X}$ with the probability mass distribution $p\left(x\right) = \Pr\left(\mathbf{X} = x\right)$ for $x \in \mathbf{X}$ is defined by Equation 30. The probability distribution of $\mathbf{X}$ is given by $p(x) = (p_1, ..., p_n)$ with where $p_i$ represents the probability of each of the $n$ outcomes; however, to simplify the notation, the limits of summation will be simply noted as $x$, indicating a summation over the domain of the corresponding variable.

$$H(\mathbf{X}) = -\sum_x p(x) \log_2 p(x) \tag{30}$$

The entropy function satisfies three conditions: 1) $H$ is continuous function of the probabilities, 2) if all probabilities are equal, the entropy is a monotonically increasing function of the number of probabilities considered, and 3) the information measure must be the same regardless of composition of the individual probabilities. Shannon proved that the only equation that satisfied these conditions is given by $H(p_i) = -K \sum_i p_i \ln p_i$, where $K$ is a positive constant.

In the discrete case, the entropy is always non-negative and has a maximal value of $\ln |N|$, where $N$ is the number of possible states, if $p$ is uniform. This is intuitive since a uniform distribution represents the highest uncertainty with no outcome having a higher probability than any other. When interpreted with respect to a discrete stochastic system, where $p$ represents the probability of each state of the system, the entropy is a measure of the unpredictability a future state of any system.

Information theory and entropy provide us with not only a theoretical basis for the analysis of systems, but also a highly practical one. Information theory is a nonparametric approach since it does not assume a functional form of the probability distribution. This gives us flexibility in the types of systems we can apply it to. Also, entropy is a "measure of observational variety or of actual (as opposed to logically possible) diversity" [163]. Unobserved outcomes or any outcomes with probability zero do not enter the measure with the convention that $0 \log 0 = 0$.

#### *B.2.2.1 Continuous formulation of entropy*

Information theory can also be formulated for random variables with continuous distributions. The difference between discrete and continuous formulations are straight forward, with the summations of the discrete form replaced with integrals over the continuous domain. However, the continuous form loses many of the desirable properties of the discrete form. Unlike the discrete form, which measures entropy is an *absolute* measure, the continuous entropy is a measure of *relative* entropy [149, p. 17]. Continuous entropy is not invariant under coordinate transformation [149, p. 20], and is therefore dependent on the coordinate system used. Because of these differences, continuous entropy can be negative or infinite, if it exists at all.

## *B.3 Mutual Information*

Conditional entropy, relative entropy, and mutual information are all closely related concepts. In fact, their relationship to each other can be defined by Equation 31. Each of these quantities will be discussed in this section.

$$I\left(\mathbf{X},\mathbf{Y}\right)=H\left(\mathbf{X}\right)-H\left(\mathbf{X}|\mathbf{Y}\right) \tag{31}$$

### B.3.1 Conditional Entropy

Considering two discrete random variables $\mathbf{X}$ and $\mathbf{Y}$, a *conditional* entropy can be defined as Equation 32. Conditional entropy is the entropy of a random variable conditional on the knowledge of another random variable. The joint distribution is given by $p\left(x,y\right)$ and the marginal distributions of $\mathbf{X}$ and $\mathbf{Y}$ are given by $p_{\mathbf{X}}\left(x\right)$ and $p_{\mathbf{Y}}\left(y\right)$, respectively.

$$H\left(\mathbf{X}|\mathbf{Y}\right)=-\sum_{x}\sum_{y}p\left(x,y\right)\log\frac{p\left(x,y\right)}{p_{\mathbf{Y}}\left(y\right)} \tag{32}$$

### B.3.2 Relative Entropy

Relative entropy is a measure of the distance between two distributions. The Kullback-Leibler (KL) divergence between two probability distributions $p\left(x\right)$ and $q\left(x\right)$ is defined as Equation 33.

$$D\left(p\|q\right)\equiv\sum_{x}p\left(x\right)\log\frac{p\left(x\right)}{q\left(x\right)} \tag{33}$$

Other terms for relative entropy include Kullback-Leibler information number, information divergence, information for discrimination, information gain, and cross entropy. Kullback-Leibler divergence is a type of Bregman divergence. Like all Bregman divergences, they are defined on a convex function; therefore, relative entropy has the property of convexity.

Relative entropy has the property of being non-negative and equal to zero if and only if $p = q$. However, this quantity is not a true measure since it is not symmetric and does not satisfy the triangle inequality [77, p. 19]. One simple method, as was done by Fisch et al. [114], to make it symmetric is to define a new metric, $KL_2$, as the average between the relative entropy between $p$ and $q$, and the relative entropy between $q$ and $p$ as shown in Equation 34.

$$KL_2 = \frac{1}{2} D\left(p \| q\right) + \frac{1}{2} D\left(q \| p\right) \tag{34}$$

Jensen-Shannon divergence is a further improvement, with the property of being symmetric and finite-valued, as defined in Equation 35, where $r = \frac{1}{2}\left(p + q\right)$.

$$JSD\left(p \| q\right) = \frac{1}{2} D\left(p \| r\right) + \frac{1}{2} D\left(q \| r\right) \tag{35}$$

### B.3.3 Mutual Information

If two random variables are dependent, knowledge of one variables leads to a reduction in uncertainty in the other variable. This is called *mutual information*. Mutual information is therefore a measure of dependence between $\mathbf{X}$ and $\mathbf{Y}$. The mutual information between $\mathbf{X}$ and $\mathbf{Y}$, $I\left(\mathbf{X}, \mathbf{Y}\right)$, is the relative entropy between the joint probability distribution $p\left(x, y\right)$ and the product of the marginal distributions $p_{\mathbf{X}}\left(x\right)$ and $p_{\mathbf{Y}}\left(y\right)$ of $\mathbf{X}$ and $\mathbf{Y}$, respectively. This is shown in Equation 36.

$$I\left(\mathbf{X}, \mathbf{Y}\right) \equiv D\left(p\left(x, y\right) \| p_{\mathbf{X}}\left(x\right) p_{\mathbf{Y}}\left(y\right)\right) = \sum_{x} \sum_{y} p\left(x, y\right) \log \frac{p\left(x, y\right)}{p_{\mathbf{X}}\left(x\right) p_{\mathbf{Y}}\left(y\right)} \tag{36}$$

If the random variables are independent, the mutual information is equal to zero. If the random variables are dependent, then the mutual information is non-negative and symmetric [77, p. 7]. The symmetry implies that knowledge of $\mathbf{X}$ provides as much information about $\mathbf{Y}$ as knowledge of $\mathbf{Y}$ provides about $\mathbf{X}$.

# REFERENCES

[1] Aksaray, D., "Formulation of Control Strategies for Requirement Definition of Multi-Agent Surveillance Systems," PhD thesis, Georgia Institute of Technology, Dec. 2014.

[2] Aksaray, D., Griendling, K., and Mavris, D., "UAVs for Law Enforcement: A Case Study for Connectivity and Fuel Management," *14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014, DOI: 10.2514/6.2014-2415.

[3] Alcazar, J. A. and Garcia, E., "Parametric Analysis for Modeling and Simulation of Stochastic Behavior in the Predator-Prey Pursuit Domain," *SIMULATION*, Vol. 82, No. 12, 2006, pp. 827–840, DOI: 10.1177/0037549707075490.

[4] Alderson, D. L. and Doyle, J. C., "Can Complexity Science Support the Engineering of Critical Network Infrastructures?" *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, Oct. 2007, pp. 44–51, DOI: 10.1109/ICSMC.2007.4414241.

[5] Alexander, S., *Space, Time, and Deity*, Reprinted, 1950 The Humanities Press, New York, Vol. 2, London: Macmillan and Co., Ltd., 1920.

[6] Amaral, L. and Ottino, J., "Complex Networks: Augmenting the Framework for the Study of Complex Systems," *European Physical Journal B: Condensed Matter Physics*, Vol. 38, Mar. 2004, pp. 147–162, DOI: 10.1140/epjb/e2004-00110-5.

[7] Anderson, C., "Creation of Desirable Complexity: Strategies for Designing Self-Organized Systems," *Complex Engineered Systems*, ed. by Braha, D., Minai, A. A., and Bar-Yam, Y., Vol. 14, Understanding Complex Systems, Springer Berlin / Heidelberg, 2006, pp. 101–121, DOI: 10.1007/3-540-32834-3_5.

[8] Anderson, C. and McShea, D. W., "Individual Versus Social Complexity, with Particular Reference to Ant Colonies," *Biological Reviews*, Vol. 76, No. 2, 2001, pp. 211–237, DOI: 10.1017/S1464793101005656.

[9] Anderson, P. W., "More is Different: Broken Symmetry and the Nature of the Hierarchical Structure of Science," *Science*, Vol. 177, No. 4047, Aug. 1972, pp. 393–396, DOI: 10.1126/science.177.4047.393.

[10] Antony, J., *Design of Experiments for Engineers and Scientists*, Elsevier Ltd., 2003.

[11] Ashby, W. R., *An Introduction to Cybernetics*, Chapman & Hall, London, 1956, p. 295.

[12] Assad, A. and Packard, N. H., "Emergent Colonization in an Artificial Ecology," *Towards a Practice of Autonomous Systems: Proceedings of the First European*

*Conference on Artifical Life*, ed. by Varela, F. and Bourgine, P., Cambridge, MA: The MIT Press, 1992, pp. 143–152.

[13] Auer, C., Wüchner, P., and Meer, H. D., "Target-Oriented Self-Structuring in Classifying Cellular Automata," *Journal of Cellular Automata*, Vol. 6, 2010.

[14] Axelrod, R., "Advancing the Art of Simulation in the Social Sciences," *Simulating Social Phenomena*, ed. by Conte, R., Hegselmann, R., and Terna, P., Springer-Verlag, 1997, pp. 21–40.

[15] Ay, N., Der, R., and Prokopenko, M., "Guided Self-Organization: Perception--Action Loops of Embodied Systems," *Theory in Biosciences*, Vol. 131, No. 3, 2011, pp. 125–127, DOI: 10.1007/s12064-011-0140-1.

[16] Ay, N., Olbrich, E., Bertschinger, N., and Jost, J., "A Unifying Framework for Complexity Measures of Finite Systems," *SFI Working Paper*, 06-08-028, 2006.

[17] Badii, R. and Politi, A., *Complexity: Hierarchical Structures and Scaling in Physics*, Cambridge University Press, 1997.

[18] Bak, P., Tang, C., and Wiesenfeld, K., "Self-Organized Criticality," *Physical Review A*, Vol. 38, No. 1, 1988, pp. 364–375.

[19] Balduzzi, D., "Detecting Emergent Processes in Cellular Automata with Excess Information," *Advances in Artificial Life, ECAL 2011*, ed. by Lenaerts, T., Giacobini, M., Bersini, H., Bourgine, P., Dorigo, M., and Doursat, R., MIT Press, 2011, pp. 55–62.

[20] Balestrini Robinson, S., "A Modeling Process to Understand Complex System Architectures," PhD thesis, Georgia Institute of Technology, 2009.

[21] Ball, R. C., Diakonova, M., and MacKay, R. S., "Quantifying Emergence in Terms of Persistent Mutual Information," *Advances In Complex Systems*, Vol. 13, No. 3, June 2010, pp. 327–338, DOI: 10.1142/S021952591000258X.

[22] Barton, R. R. and Meckesheimer, M., "Metamodel-Based Simulation Optimization," *Simulation*, ed. by Henderson, S. G. and Nelson, B. L., Vol. 13, Handbooks in Operations Research and Management Science, Elsevier, 2006, Ch. 18, pp. 535–574, DOI: 10.1016/S0927-0507(06)13018-2.

[23] Bar-Yam, Y., "Multiscale Complexity/Entropy," *Advances in Complex Systems*, Vol. 7, No. 1, Mar. 2004, pp. 47–63, DOI: 10.1142/S0219525904000068.

[24] Bar-Yam, Y., "A Mathematical Theory of Strong Emergence Using Multiscale Variety," *Complexity*, Vol. 9, No. 6, July 2004, pp. 15–24, DOI: 10.1002/cplx.20029.

[25] Bar-Yam, Y., "Engineering Complex Systems: Multiscale Analysis and Evolutionary Engineering," *Complex Engineered Systems*, ed. by Braha, D., Minai, A. A.,

and Bar-Yam, Y., Understanding Complex Systems, Springer Berlin Heidelberg, 2006, pp. 22–39, DOI: 10.1007/3-540-32834-3_2.

[26] Bar-Yam, Y., "Multiscale Variety in Complex Systems," *Complexity*, Vol. 9, No. 4, 2004, pp. 37–45, DOI: 10.1002/cplx.20014.

[27] Bar-Yam, Y., "When Systems Engineering Fails---Toward Complex Systems Engineering," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, Washington, DC, Oct. 2003, pp. 2021–2028, DOI: 10.1109/ICSMC.2003.1244709.

[28] Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P., "Experimental Design and Observation for Large Systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, No. 1, 1996, pp. 77–94.

[29] Batterman, R. W., ed., *The Oxford Handbook of Philosophy of Physics*, Oxford University Press, 2013.

[30] Batterman, R. W., "The Tyranny of Scales," *The Oxford Handbook of Philosophy of Physics*, ed. by Batterman, R. W., Oxford University Press, 2013, Ch. 7, pp. 255–286.

[31] Bedau, M. A., "Artificial Life," *Handbook of the Philosophy of Science*, ed. by Matthen, M. and Stephens, C., Vol. 3 (Philosophy of Biology), Elsevier BV, 2007, pp. 595–613.

[32] Bedau, M. A., "Artificial Life: Organization, Adaptation and Complexity from the Bottom Up," *Trends in Cognitive Sciences*, Vol. 7, No. 11, 2003, pp. 505–512, DOI: 10.1016/j.tics.2003.09.012.

[33] Bedau, M. A., "Downward Causation and Autonomy in Weak Emergence," *Principia*, Vol. 6, 2003, pp. 5–50.

[34] Bedau, M. A., "Weak Emergence," *Noûs*, Vol. 31, No. Supplement 11, Philosophical Perspectives: Mind, Causation, and World, 1997, pp. 375–399, DOI: 10.1111/0029-4624.31.s11.17.

[35] Bedau, M. A. and Humphreys, P., eds., *Emergence*, MIT Press, 2008.

[36] Beekman, M., Sumpter, D. J. T., and Ratnieks, F. L. W., "Phase Transition Between Disordered and Ordered Foraging in Pharaoh's Ants," *Proceedings of the National Academy of Sciences*, Vol. 98, No. 17, 2001, pp. 9703–9706, DOI: 10.1073/pnas.161285298.

[37] Beni, G., "From Swarm Intelligence to Swarm Robotics," *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers*, ed. by Şahin, E. and Spears, W. M., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–9, DOI: 10.1007/978-3-540-30552-1_1.

[38] Bennett, C. H., "Logical Depth and Physical Complexity," *The Universal Turing Machine: A Half-Century Survey*, Berlin, Germany: Oxford University Press, Inc., 1988, pp. 227–257.

[39] Bennett, M., Schatz, M. F., Rockwood, H., and Wiesenfeld, K., "Huygens's Clocks," *Proceedings of the Royal Society London A: Mathematical, Physical & Engineering Sciences*, Vol. 458, 2002, pp. 563–579, DOI: 10.1098/rspa.2001.0888.

[40] Berlekamp, E. R., Conway, J. H., and Guy, R. K., *Winning Ways for Your Mathematical Plays*, Vol. 2, New York, NY: Academic Press, 1982.

[41] Bialek, W., Nemenman, I., and Tishby, N., "Predictability, Complexity, and Learning," *Neural Computation*, Vol. 13, No. 11, Nov. 2001, pp. 2409–2463, DOI: 10.1162/089976601753195969.

[42] Bisig, D. and Pfeifer, R., "Understanding by Design: The Synthetic Approach to Intelligence," *Explorations in Architecture: Teaching, Design, Research*, ed. by Geiser, R. and Kultur, B. F., 2008.

[43] Bollt, E. M., Stanford, T., Lai, Y.-C., and Zyczkowski, K., "Validity of Threshold-Crossing Analysis of Symbolic Dynamics from Chaotic Time Series," *Physical Review Letters*, Vol. 85, No. 16, 2000, pp. 3524–3527.

[44] Bonabeau, E., "Social Insect Colonies as Complex Adaptive Systems," *Ecosystems*, Vol. 1, No. 5, 1998, pp. 437–443, DOI: 10.1007/s100219900038.

[45] Bonabeau, E., Dessalles, J.-L., and Grumbach, A., "Characterizing Emergent Phenomena (2): A Conceptual Framework," *Revue Internationale De Systémique*, Vol. 9, No. 3, 1995, pp. 347–371.

[46] Bonabeau, E., Funes, P., and Orme, B., "Exploratory Design of Swarms," *Proceedings of the 2nd International Workshop on the Mathematics and Algorithms of Social Insects Georgia Institute of Technology, Atlanta, GA, December 15--17, 2003*, ed. by Anderson, C. and Balch, T., 2003, pp. 17–24.

[47] Bonabeau, E. and Théraulaz, G., "Swarm Smarts," *Scientific American*, Mar. 2000, pp. 72–79.

[48] Bonabeau, E., Theraulaz, G., Deneubourg, J.-L., Aron, S., and Camazine, S., "Self-Organization in Social Insects," *Trends in Ecology & Evolution*, Vol. 12, No. 5, 1997, pp. 188–193, DOI: 10.1016/S0169-5347(97)01048-3.

[49] Boschetti, F., "Detecting Qualitatively Different Behaviours in Complex Models," *18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009*, 2009.

[50] Boschetti, F., "Detecting Behaviours in Ecological Models," *Ecological Complexity*, Vol. 7, No. 1, Mar. 2010, pp. 76–85, DOI: 10.1016/j.ecocom.2009.05.014.

[51] Boschetti, F. and Gray, R., "A Turing Test for Emergence," *Advances in Applied Self-Organizing Systems*, ed. by Prokopenko, M., Advanced Information and

Knowledge Processing, Springer London, 2008, pp. 349–364, DOI: 10.1007/978-1-84628-982-8_15.

[52] Boschetti, F., Prokopenko, M., Macreadie, I., and Grisogono, A.-M., "Defining and Detecting Emergence in Complex Networks," *Knowledge-Based Intelligent Information and Engineering Systems*, ed. by Khosla, R., Howlett, R., and Jain, L., Vol. 3684, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2005, pp. 573–580, DOI: 10.1007/11554028_79.

[53] Bouarfa, S., Blom, H. A., Curran, R., and Everdij, M. H., "Agent-Based Modeling and Simulation of Emergent Behavior in Air Transportation," *Complex Adaptive Systems Modeling*, Vol. 1, No. 1, 2013, p. 15, DOI: 10.1186/2194-3206-1-15.

[54] Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M., "Swarm Robotics: A Review from the Swarm Engineering Perspective," *Swarm Intelligence*, Vol. 7, No. 1, 2013, pp. 1–41, DOI: 10.1007/s11721-012-0075-2.

[55] Broad, C. D., *The Mind and Its Place in Nature*, New York: Harcourt, Brace & Company, Inc., 1925.

[56] Buchli, J. and Santini, C. C., "Complexity Engineering: Harnessing Emergent Phenomena as Opportunities for Engineering," *Reports of the Santa Fe Institute's Complex Systems Summer School 2005*, Santa Fe Institute, 2005.

[57] Bunge, M., *Philosophy of Science: From Explanation to Justification*, Revised, Vol. 2, Transaction Publishers, 1998.

[58] Campbell, D. T., "Downward Causation in Hierarchically Organized Biological Systems," *Studies in the Philosophy of Biology*, ed. by Ayala, F. J. and Dobzhansky, T., Macmillan: New York, 1974, pp. 179–186.

[59] Cao, Y. U., Fukunaga, A. S., and Kahng, A., "Cooperative Mobile Robotics: Antecedents and Directions," *Autonomous Robots*, Vol. 4, No. 1, 1997, pp. 7–27, DOI: 10.1023/A:1008855018923.

[60] Carreras, I., Miorandi, D., Saint-Paul, R., and Chlamtac, I., "Bottom-Up Design Patterns and the Energy Web," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 40, No. 4, July 2010, pp. 815–824, DOI: 10.1109/TSMCA.2010.2048025.

[61] Casbeer, D. W., Kingston, D. B., Beard, R. W., and McLain, T. W., "Cooperative Forest Fire Surveillance Using a Team of Small Unmanned Air Vehicles," *International Journal of Systems Science*, Vol. 37, No. 6, 2006, pp. 351–360, DOI: 10.1080/00207720500438480.

[62] Casbeer, D. W., Swindlehurst, A. L., and Beard, R., "Connectivity in a UAV Multi-Static Radar Network," *AIAA Guidance, Navigation, and Control Conference and Exhibit, 21 - 24 August 2006, Keystone, Colorado*, 2006-6209, 2006, DOI: 10.2514/6.2006-6209.

[63] Cavallo, D., “Emergent Design and Learning Environments: Building on Indigenous Knowledge,” *IBM Systems Journal*, Vol. 39, No. 3-4, 2000, pp. 768–781.

[64] Chalmers, D., “The Conscious Mind,” New York, Oxford University Press, 1996, Ch. Supervenience, pp. 32–42.

[65] Chalmers, D. J., “Strong and Weak Emergence,” *The Re-Emergence of Emergence*, ed. by Clayton, P. and Davies, P., Oxford: Oxford University Press, 2006, Ch. 11, pp. 244–254.

[66] Chan, W. K. V., “Interaction Metric of Emergent Behaviors in Agent-Based Simulation,” *Proceedings of the 2011 Winter Simulation Conference*, Dec. 2011, pp. 357–368, DOI: 10.1109/WSC.2011.6147763.

[67] Chan, W. K. V., Son, Y.-J., and Macal, C. M., “Agent-Based Simulation Tutorial - Simulation of Emergent Behavior and Differences Between Agent-Based Simulation and Discrete-Event Simulation,” *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 135–150, DOI: 10.1109/WSC.2010.5679168.

[68] Checkland, P., *Systems Thinking, Systems Practice*, Republished 1999 with a 30-year retrospective, John Wiley and Sons, Chichester UK, 1999.

[69] Chen, C.-C., Clack, C. D., and Nagl, S. B., “Identifying Multi-Level Emergent Behaviors in Agent-Directed Simulations Using Complex Event Type Specifications,” *Simulation: Transactions of the Society for Modeling and Simulation International*, Vol. 86, No. 1, Jan. 2010, pp. 41–51, DOI: 10.1177/0037549709106692.

[70] Chen, C.-C., Nagl, S. B., and Clack, C. D., “A Calculus for Multi-Level Emergent Behaviours in Component-Based Systems and Simulations,” *Proceedings of the Satellite Conference on Emergent Properties in Artificial and Natural Systems (EPNACS)*, ed. by Alaoui, A. M. A., Bertelle, C., Cosaftis, M., and Duchamp, G. H., 2007, pp. 35–51.

[71] Chen, W., “A Robust Concept Exploration Method for Configuring Complex Systems,” PhD thesis, Georgia Institute of Technology, 1995.

[72] Chevaleyre, Y., “Theoretical Analysis of the Multi-Agent Patrolling Problem,” *Intelligent Agent Technology, 2004. (IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on*, Sept. 2004, pp. 302–308, DOI: 10.1109/IAT.2004.1342959.

[73] Chibaya, C. and Bangay, S., “The Relationship Between Emergence of the Shortest Path and Information Value Using Ant-Like Agents,” *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*, SAICSIT ’08, Wilderness, South Africa: ACM, 2008, pp. 27–34, DOI: 10.1145/1456659.1456663.

[74] Clayton, P. and Davies, P., eds., *The Re-Emergence of Emergence: The Emergentist Hypothesis from Science to Religion*, Oxford University Press, 2006.

[75] Committee on CMMP 2010 et al., "What Happens Far from Equilibrium and Why?" *Condensed-Matter and Materials Physics: The Science of the World Around Us*, The National Academies Press, 2007, Ch. 5.

[76] Corning, P. A., "The Re-Emergence of "Emergence": A Venerable Concept in Search of a Theory," *Complexity*, Vol. 7, No. 6, July 2002, pp. 18–30, DOI: 10.1002/cplx.10043.

[77] Cover, T. M. and Thomas, J. A., *Elements of Information Theory*, Second, John Wiley & Sons, Inc., 2006.

[78] Crick, F., *Astonishing Hypothesis: The Scientific Search for the Soul*, First, Touchstone: Simon & Schuster Inc., 1994.

[79] Crutchfield, J. P., Farmer, J. D., Packard, N., and Shaw, R. S., "Chaos," *Scientific American*, Vol. 255, No. 6, Dec. 1986, pp. 46–57.

[80] Crutchfield, J., "Is Anything Ever New? Considering Emergence," *Complexity: Metaphors, Models, and Reality*, ed. by Cowan, G., Pines, D., and Melzner, D., Vol. XIX, SFI Series in the Sciences of Complexity, Addison-Wesley, Redwood City, 1994, pp. 515–537.

[81] Crutchfield, J. P., "Between Order and Chaos," *Nature Physics*, Vol. 8, No. 1, Jan. 2012, pp. 17–24, DOI: 10.1038/nphys2190.

[82] Crutchfield, J. P., "The Calculi of Emergence: Computation, Dynamics, and Induction," *Physica D: Nonlinear Phenomena*, Vol. 75, No. 1-3, Aug. 1994, pp. 11–54, DOI: 10.1016/0167-2789(94)90273-9.

[83] Crutchfield, J. P. and Feldman, D. P., "Regularities Unseen, Randomness Observed: Levels of Entropy Convergence," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 13, No. 1, Mar. 2003, pp. 25–54, DOI: 10.1063/1.1530990.

[84] Crutchfield, J. P. and Shalizi, C. R., "Thermodynamic Depth of Causal States: Objective Complexity Via Minimal Representations," *Phys. Rev. E*, Vol. 59, 1 Jan. 1999, pp. 275–283, DOI: 10.1103/PhysRevE.59.275.

[85] Crutchfield, J. P. and Young, K., "Inferring Statistical Complexity," *Physical Review Letters*, Vol. 63, No. 2, July 1989, pp. 105–108, DOI: 10.1103/PhysRevLett.63.105.

[86] Darley, V., "Emergent Phenomena and Complexity," *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, ed. by Brooks, R. and Maes, P., MIT Press, 1994, pp. 411–416.

[87] Davis, P. K., "New Paradigms and New Challenges," *Proceedings of the Winter Simulation Conference, 2005*, 2005, pp. 1067–1076, DOI: 10.1109/WSC.2005.1574360.

[88] Davis, P. K. and Bigelow, J. H., *Motivated Metamodels: Synthesis of Cause-Effect Reasoning and Statistical Metamodeling*, tech. rep., RAND Corporation, 2003.

[89] Day, M. A., Clement, M. R., Russo, J. D., Davis, D., and Chung, T. H., "Multi-UAV Software Systems and Simulation Architecture," *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, June 2015, pp. 426–435, DOI: 10.1109/ICUAS.2015.7152319.

[90] De Wolf, T. and Holvoet, T., "Emergence and Self-Organisation: a Statement of Similarities and Differences," *Proceedings of the International Workshop on Engineering Self-Organising Applications 2004*, 2004, pp. 96–110.

[91] De Wolf, T. and Holvoet, T., "Emergence Versus Self-Organisation: Different Concepts but Promising When Combined," *Engineering Self-Organising Systems*, ed. by Brueckner, S., Di Marzo Serugendo, G., Karageorgos, A., and Nagpal, R., Vol. 3464, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 1–15, DOI: 10.1007/11494676_1.

[92] De Wolf, T. and Holvoet, T., "Towards a Full Life-Cycle Methodology for Engineering Decentralised Multi-Agent Systems," *Proceedings of the Fourth International Workshop on Agent-Oriented Methodologies*, 2005, pp. 1–12.

[93] Deacon, T. W., "Emergence: The Hole at the Wheel's Hub," *The Re-Emergence of Emergence: The Emergentist Hypothesis from Science to Religion*, ed. by Clayton, P. and Davies, P., Oxford University Press, 2006, Ch. 5, pp. 111–150.

[94] Dennett, D. C., "Real Patterns," *Journal of Philosophy*, Vol. 87, No. 1, Jan. 1991, pp. 27–51, DOI: 10.2307/2027085.

[95] Deshmukh, A. V., Talavage, J. J., and Barash, M. M., "Complexity in Manufacturing Systems, Part 1: Analysis of Static Complexity," *IIE Transactions*, Vol. 30, No. 7, 1998, pp. 645–655, DOI: 10.1080/07408179808966508.

[96] Di Marzo Serugendo, G., Gleizes, M., and Karageorgos, A., "Self-Organisation and Emergence in MAS: an Overview," *Informatica*, Vol. 20, No. 1, 2006, pp. 45–54.

[97] Di Marzo Serugendo, G., Foukia, N., Hassas, S., Karageorgos, A., Mostéfaoui, S., Rana, O., Ulieru, M., Valckenaers, P., and Van Aart, C., "Self-Organisation: Paradigms and Applications," *Engineering Self-Organising Systems*, ed. by Di Marzo Serugendo, G., Karageorgos, A., Rana, O., and Zambonelli, F., Vol. 2977, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2004, pp. 1–19, DOI: 10.1007/978-3-540-24701-2_1.

[98] Diakonova, M. and MacKay, R. S., "Mathematical Examples of Space-Time Phases," *International Journal of Bifurcation and Chaos*, Vol. 21, No. 8, Aug. 2011, pp. 2297–2304, DOI: 10.1142/S0218127411029793.

[99] Dieter, G. E., *Engineering Design: A Materials and Processing Approach*, 3rd ed., McGraw-Hill, 2000.

[100] Doering, D., Benenmann, A., Lerm, R., Freitas, E. P. de, Muller, I., Winter, J. M., and Pereira, C. E., "Design and Optimization of a Heterogeneous Platform for Multiple UAV Use in Precision Agriculture Applications," *19th World Congress The International Federation of Automatic Control Cape Town, South Africa, August 24-29, 2014*, 2014, pp. 12272–12277, DOI: 10.3182/20140824-6-ZA-1003.02261.

[101] Dogaru, R., "Emergence, Locating and Measuring It," *Systematic Design for Emergence in Cellular Nonlinear Networks*, Vol. 95, Studies in Computational Intelligence, Springer Berlin / Heidelberg, 2008, pp. 47–75, DOI: 10.1007/978-3-540-76801-2_4.

[102] Dogaru, R., *Universality and Emergent Computation in Cellular Neural Networks*, ed. by Chua, L. O., Vol. 43, World Scientific Series on Nonlinear Science, World Scientific Publishing, 2003.

[103] Doursat, R., "The Myriads of Alife: Importing Complex Systems and Self-Organization Into Engineering," *Artificial Life (ALIFE), 2011 IEEE Symposium on*, 2011, pp. 1–8, DOI: 10.1109/ALIFE.2011.5954671.

[104] Doursat, R. and Ulieru, M., "Emergent Engineering for the Management of Complex Situations," *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, Autonomics '08, Turin, Italy: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, 14:1–14:10.

[105] Edelstein-Keshet, L., Watmough, J., and Ermentrout, G. B., "Trail Following in Ants: Individual Properties Determine Population Behaviour," *Behavioral Ecology and Sociobiology*, Vol. 36, No. 2, Feb. 1995, pp. 119–133, DOI: 10.1007/BF00170717.

[106] Edmonds, B., "Using the Experimental Method to Produce Reliable Self-Organised Systems," *Engineering Self-Organising Systems*, ed. by Brueckner, S., Serugendo, G., Karageorgos, A., and Nagpal, R., Vol. 3464, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2005, pp. 84–99, DOI: 10.1007/11494676_6.

[107] Efromovich, S., "Orthogonal Series Density Estimation," *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, No. 4, 2010, pp. 467–476, DOI: 10.1002/wics.97.

[108] Enright, J. J., Frazzoli, E., Pavone, M., and Savla, K., "UAV Routing and Coordination in Stochastic, Dynamic Environments," *Handbook of Unmanned Aerial Vehicles*, ed. by Valavanis, K. P. and Vachtsevanos, G. J., Springer Netherlands, 2015, pp. 2079–2109, DOI: 10.1007/978-90-481-9707-1_28.

[109] Érdi, P., *Complexity Explained*, Springer-Verlag Berlin, 2008, DOI: 10.1007/978-3-540-35778-0.

[110] Evans, A., Heppenstall, A., and Birkin, M., "Understanding Simulation Results," *Simulating Social Complexity*, ed. by Edmonds, B. and Meyer, R., Understanding Complex Systems, Springer Berlin Heidelberg, 2013, pp. 173–195, DOI: 10.1007/978-3-540-93813-2_9.

[111] Everdij, M., Scholte, J., Blom, H., and Stroeve, S., "An Investigation of Emergent Behaviour Viewpoints in Literature, and Their Usability in Air Traffic Management," *First Complex World Annual Conference, Seville, Spain, 6- 8 July 2011*, 2011.

[112] Felder, W. N. and Collopy, P., "The Elephant in the Mist: What We Don't Know about the Design, Development, Test and Management of Complex Systems," *Journal of Aerospace Operations*, Vol. 1, No. 4, 2012, pp. 317–327, DOI: 10.3233/AOP-120024.

[113] Felici, M., "Capturing Emerging Complex Interactions: Safety Analysis in Air Traffic Management," *Reliability Engineering & System Safety*, Vol. 91, No. 12, Dec. 2006, pp. 1482–1493, DOI: 10.1016/j.ress.2006.01.010.

[114] Fisch, D., Jänicke, M., Sick, B., and Müller-Schloer, C., "Quantitative Emergence: A Refined Approach Based on Divergence Measures," *Proceedings of the 2010 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010)*, 5630529, 2010, pp. 94–103, DOI: 10.1109/SASO.2010.31.

[115] Fisch, D., Jänicke, M., Müller-Schloer, C., and Sick, B., "Divergence Measures as a Generalised Approach to Quantitative Emergence," *Organic Computing: A Paradigm Shift for Complex Systems*, ed. by Müller-Schloer, C., Schmeck, H., and Ungerer, T., Vol. 1, Autonomic Systems, Springer Basel, 2011, pp. 53–66, DOI: 10.1007/978-3-0348-0130-0_3.

[116] Friedman, J. H., Bentley, J. L., and Finkel, R. A., "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, Vol. 3, No. 3, Sept. 1977, pp. 209–226, DOI: 10.1145/355744.355745.

[117] Fromm, J., "On Engineering and Emergence," *ArXiv:nlin/0601002*, 2006, pp. 1–11.

[118] Fromm, J., "Ten Questions about Emergence," *ArXiv:nlin/0509049*, 2005, pp. 1–13.

[119] Fuentes, M. A., "Complexity and the Emergence of Physical Properties," *Entropy*, Vol. 16, No. 8, 2014, pp. 4489–4496, DOI: 10.3390/e16084489.

[120] Gardner, M., "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game Life," *Scientific American*, Vol. 223, 1970, pp. 120–123.

[121] Gautreau, A., Barrat, A., and Barthélemy, M., "Microdynamics in Stationary Complex Networks," *Proceedings of the National Academy of Sciences*, Vol. 106, No. 22, 2009, pp. 8847–8852, DOI: 10.1073/pnas.0811113106.

[122] Gell-Mann, M. and Lloyd, S., *Effective Complexity*, tech. rep. SFI Working Paper: 2003-12-068, Santa Fe Institute, 2003.

[123] Gell-Mann, M. and Lloyd, S., "Information Measures, Effective Complexity, and Total Information," *Complexity*, Vol. 2, No. 1, Sept. 1996, pp. 44–52, DOI: 10.1002/(SICI)1099-0526(199609/10)2:1<44::AID-CPLX10>3.0.CO;2-X.

[124] Gershenson, C. and Fernández, N., "Complexity and Information: Measuring Emergence, Self-Organization, and Homeostasis at Multiple Scales," *Complexity*, Vol. 18, No. 2, Nov. 2012, pp. 29–44, DOI: 10.1002/cplx.21424.

[125] Gilbert, N., *Agent-Based Models*, ed. by Liao, T., Sage University Paper Series on Quantitative Applications in the Social Sciences 07-153, SAGE Publications, 2008, p. 98.

[126] Giunta, A. A., "Use of Data Sampling, Surrogate Models, and Numerical Optimization in Engineering Design," *40th AIAA Aerospace Sciences Meeting & Exhibit*, AIAA 2002-0538, 2002, pp. 1–12, DOI: 10.2514/6.2002-538.

[127] Gmeiner, P., "Some Properties of Persistent Mutual Information," *Proceedings of the European Conference on Complex Systems 2012*, ed. by Gilbert, T., Kirkilionis, M., and Nicolis, G., Springer Proceedings in Complexity, Springer International Publishing, 2013, pp. 867–876, DOI: 10.1007/978-3-319-00395-5_105.

[128] Goldstein, J., "Emergence as a Construct: History and Issues," *Emergence: Complexity and Organization*, Vol. 1, No. 1, Mar. 1999, pp. 49–72, DOI: 10.1207/s15327000em0101_4.

[129] Gore, R. and Reynolds Paul F., J., "Applying Causal Inference to Understand Emergent Behavior," *Proceedings of the 2008 Winter Simulation Conference*, 4736133, 2008, pp. 712–721, DOI: 10.1109/WSC.2008.4736133.

[130] Gore, R., Reynolds Paul F., J., Tang, L., and Brogan, D. C., "Explanation Exploration: Exploring Emergent Behavior," *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation, PADS*, 4262797, San Diego, CA, June 2007, pp. 113–120, DOI: 10.1109/PADS.2007.18.

[131] Grassberger, P., "Toward a Quantitative Theory of Self-Generated Complexity," *International Journal of Theoretical Physics*, Vol. 25, No. 9, 1986, pp. 907–938, DOI: 10.1007/BF00668821.

[132] Grossman, R. L., Sabala, M., Gu, Y., Anand, A., Handley, M., Sulo, R., and Wilkinson, L., "Discovering Emergent Behavior From Network Packet Data: Lessons from the Angle Project," 2009, pp. 1–18.

[133] Haglich, P., Rouff, C., and Pullum, L., “Detecting Emergent Behaviors with Semi-Boolean Algebra,” *AIAA Infotech at Aerospace 2010 Conference, Atlanta, GA*, AIAA 2010-3503, Apr. 2010, DOI: 10.2514/6.2010-3503.

[134] Hand, D., Mannila, H., and Smyth, P., *Principles of Data Mining*, The MIT Press, 2001.

[135] Haque, M. A., “Biologically Inspired Heterogeneous Multi-Agent Systems,” PhD thesis, Georgia Institute of Technology, Nov. 2010.

[136] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second, Springer, 2009.

[137] Hastings, D. and McManus, H., “A Framework for Understanding Uncertainty and Its Mitigation and Exploitation in Complex Systems,” *Engineering Systems Symposium, MIT: Cambridge, MA, March 31, 2004*, 2004.

[138] Hazen, R. M., Griffin, P. L., Carothers, J. M., and Szostak, J. W., “Functional Information and the Emergence of Biocomplexity,” *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 104, No. Suppl 1, May 2007, pp. 8574–8581, DOI: 10.1073/pnas.0701744104.

[139] Healey, R. A., “Holism and Nonseparability,” *The Journal of Philosophy*, Vol. 88, No. 8, 1991, pp. 393–421, DOI: 10.2307/2026702.

[140] Hempel, C. and Oppenheim, P., “On the Idea of Emergence,” *Aspects of Scientific Explanation and Other Essays in the Philosophy of Science*, New York: Free Press, 1965, pp. 258–264.

[141] Heylighen, F., “The Science of Self-Organization and Adaptivity,” *The Encyclopedia of Life Support Systems*, ed. by Kiel, L. D., Eolss Publishers, Oxford, 2001.

[142] Holland, J. H., “Complex Adaptive Systems,” *Daedalus*, Vol. 121, No. 1, 1992, pp. 17–30.

[143] Holland, J. H., *Emergence: From Chaos to Order*, First Printing, Helix Books: Addison-Wesley Publishing Company, Inc., Nov. 1998.

[144] Holland, J. H., *Hidden Order: How Adaptation Builds Complexity*, Addison Wesley, Reading, MA, 1995.

[145] Holland, O. T., “Taxonomy for the Modeling and Simulation of Emergent Behavior Systems,” *SpringSim ’07: Proceedings of the 2007 Spring Simulation Multiconference*, Society for Computer Simulation International, 2007, pp. 28–35.

[146] Hovda, P., “Quantifying Weak Emergence,” *Minds and Machines*, Vol. 18, No. 4, Dec. 2008, pp. 461–473, DOI: 10.1007/s11023-008-9123-5.

[147] Humphreys, P., “How Properties Emerge,” *Philosophy of Science*, Vol. 64, No. 1, Mar. 1997, pp. 1–17, DOI: 10.1086/392533.

[148] Huneman, P., “Emergence Made Ontological? Computational Versus Combinatorial Approaches,” *Philosophy of Science*, Vol. 75, No. 5, 2008, pp. 595–607, DOI: 10.1086/596777.

[149] Ihara, S., *Information Theory for Continuous Systems*, World Scientific Publishing Co., Singapore, 1993.

[150] Ilachinski, A., *Cellular Automata: A Discrete Universe*, World Scientific, 2001.

[151] Izenman, A. J., *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*, Springer New York, 2008, DOI: 10.1007/978-0-387-78189-1.

[152] Jaynes, E., “Information Theory and Statistical Mechanics I,” *Physical Review*, Vol. 106, No. 4, 1957, pp. 620–630.

[153] Jaynes, E., “Information Theory and Statistical Mechanics II,” *Physical Review*, Vol. 108, No. 2, 1957, pp. 171–190.

[154] Johnson John J., I., Tolk, A., and Sousa-Poza, A., “A Theory of Emergence and Entropy in Systems of Systems,” *Procedia Computer Science: Complex Adaptive Systems*, Vol. 20, 2013, pp. 283–289, DOI: 10.1016/j.procs.2013.09.274.

[155] Kernstine, K. H., “Inadequacies of Traditional Exploration Methods in Systems-Of-Systems Simulations,” *Systems Journal, IEEE*, Vol. 7, No. 4, 2013, pp. 528–536, DOI: 10.1109/JSYST.2013.2252864.

[156] Kernstine, K. H., “Design Space Exploration of Stochastic System-Of-Systems Simulations Using Adaptive Sequential Experiments,” PhD thesis, Georgia Institute of Technology, Aug. 2012.

[157] Kernstine, K. H., Coisnon, R., Lim, D., and Mavris, D., “Inadequacies of Stochastic Search Methods for System-Of-Systems Problems,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2012-5681, Sept. 2012, DOI: 10.2514/6.2012-5681.

[158] Kim, J., “Making Sense of Emergence,” *Philosophical Studies*, Vol. 95, No. 1-2, Aug. 1999, pp. 3–36, DOI: 10.1023/A:1004563122154.

[159] Kim, J., “The Nonreductivist’s Troubles with Mental Causation,” *Mental Causation*, ed. by Heil, J. and Mele, A., New York, Oxford University Press, 1993, pp. 189–210.

[160] Kitto, K., “Modelling And Generating Complex Emergent Behaviour,” PhD thesis, The Flinders University of South Australia, June 2006.

[161] Kleijnen, J. P. C., "An Overview of the Design and Analysis of Simulation Experiments for Sensitivity Analysis," *European Journal of Operational Research*, Vol. 164, No. 2, July 2005, pp. 287–300, DOI: 10.1016/j.ejor.2004.02.005.

[162] Kraskov, A., Stögbauer, H., and Grassberger, P., "Estimating Mutual Information," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, Vol. 69, No. 6, 6 June 2004, p. 066138, DOI: 10.1103/PhysRevE.69.066138.

[163] Krippendorff, K., *Information Theory: Structural Models for Qualitative Data*, Sage University Paper Series on Quantitative Applications in the Social Sciences 07-062, SAGE Publications, 1986, p. 96.

[164] Kube, C. R. and Zhang, H., "Collective Robotics: From Social Insects to Robots," *Adaptive Behavior*, Vol. 2, No. 2, 1993, pp. 189–218, DOI: 10.1177/105971239300200204.

[165] Kubik, A., "Toward a Formalization of Emergence," *Artificial Life*, Vol. 9, No. 1, 2003, pp. 41–65, DOI: 10.1162/106454603321489518.

[166] Langton, C. G., "Artificial Life," *Artificial Life, SFI Studies in the Sciences of Complexity*, ed. by Langton, C. G., Addison-Wesley, 1989, pp. 1–48.

[167] Langton, C. G., "Computation at the Edge of Chaos: Phase Transitions and Emergent Computation," *Physica D: Nonlinear Phenomena*, Vol. 42, No. 1-3, June 1990, pp. 12–37, DOI: 10.1016/0167-2789(90)90064-V.

[168] Laughlin, R. B. and Pines, D., "The Theory of Everything," *Proceedings of the National Academy of Sciences*, Vol. 97, No. 1, Jan. 2000, pp. 28–31, DOI: 10.1073/pnas.97.1.28.

[169] Lempert, R. J., "A New Decision Sciences for Complex Systems," *Proceedings of the National Academy of Sciences*, Vol. 99, No. Suppl 3, May 2002, pp. 7309–7313, DOI: 10.1073/pnas.082081699.

[170] Lerman, K., Martinoli, A., and Galstyan, A., "A Review of Probabilistic Macroscopic Models for Swarm Robotic Systems," *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers*, ed. by Şahin, E. and Spears, W. M., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 143–152, DOI: 10.1007/978-3-540-30552-1_12.

[171] Lewes, G. H., *Problems of Life and Mind: First Series, The Foundations of a Creed, Vol. II*, Boston, James R. Osgood and Company, 1875.

[172] Li, W., "On the Relationship Between Complexity and Entropy for Markov Chains and Regular Languages," *Complex Systems*, Vol. 5, 1991, pp. 381–399.

[173] Lloyd, S. and Pagels, H., "Complexity as Thermodynamic Depth," *Annals of Physics*, Vol. 188, No. 1, 1988, pp. 186–213, DOI: 10.1016/0003-4916(88)90094-2.

[174] Lucas, T., Sanchez, S. M., Brown, L., and Vinyard, W., "Better Designs for High-Dimensional Explorations of Distillations," *Maneuver Warfare Science 2002*, 2002, pp. 17–45.

[175] Lüdtke, N., Panzeri, S., Brown, M., Broomhead, D. S., Knowles, J., Montemurro, M. A., and Kell, D. B., "Information-Theoretic Sensitivity Analysis: a General Method for Credit Assignment in Complex Networks," *Journal of the Royal Society Interface*, Vol. 5, No. 19, 2008, pp. 223–235, DOI: 10.1098/rsif.2007.1079.

[176] Maes, P., "Modeling Adaptive Autonomous Agents," *Artificial Life*, Vol. 1, No. 1-2, 1993, pp. 135–162, DOI: 10.1162/artl.1993.1.1_2.135.

[177] Maier, M. W. and Rechtin, E., *The Art of Systems Architecting*, 2nd ed., CRC Press, 2000.

[178] Maldonado, C. E. and Gómez Cruz, N. A., "The Complexification of Engineering," *Complexity*, Vol. 17, No. 4, Mar. 2012, pp. 8–15, DOI: 10.1002/cplx.20395.

[179] Manley, E. and Cheng, T., "Understanding Road Congestion as an Emergent Property of Traffic Networks," *Proceedings of The International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2010)*, ed. by Baralt, J., Callaos, N., Chu, H. W., Savoie, M. J., and Zinn, C. D., 2010, pp. 109–114.

[180] Manson, S. M., "Simplifying Complexity: a Review of Complexity Theory," *Geoforum*, Vol. 32, No. 3, Aug. 2001, pp. 405–414, DOI: 10.1016/S0016-7185(00)00035-X.

[181] McBride, N., "Chaos Theory as a Model for Interpreting Information Systems in Organizations," *Information Systems Journal*, Vol. 15, No. 3, 2005, pp. 233–254, DOI: 10.1111/j.1365-2575.2005.00192.x.

[182] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–345.

[183] McLaughlin, B. P., "The Rise and Fall of British Emergentism," *Emergence or Reduction?: Essays on the Prospects of Nonreductive Physicalism*, ed. by Beckerman, Flohr, and Kim, Walter de Gruyter GmbH & Co., Berlin, 1992, pp. 49–93.

[184] McLaughlin, B. and Bennett, K., "Supervenience," *The Stanford Encyclopedia of Philosophy*, ed. by Zalta, E. N., Winter 2011, 2011.

[185] Menon, T. and Callender, C., "Turn and Face the Strange... Ch-Ch-Changes: Philosophical Questions Raised by Phase Transitions," *The Oxford Handbook of Philosophy of Physics*, ed. by Batterman, R. W., Oxford University Press, 2013, Ch. 5, pp. 189–223.

[186] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21, 1953, pp. 1087–1092.

[187] Mill, J. S., *A System of Logic*, 8th ed., New York: Harper & Brothers, 1882.

[188] Minai, A. A., Braha, D., and Bar-Yam, Y., "Complex Engineered Systems: A New Paradigm," *Complex Engineered Systems*, ed. by Braha, D., Minai, A. A., and Bar-Yam, Y., Vol. 14, Understanding Complex Systems, Springer Berlin / Heidelberg, 2006, pp. 1–21, DOI: 10.1007/3-540-32834-3_1.

[189] Mnif, M. and Mueller-Schloer, C., "Quantitative Emergence," *Proceedings of the 2006 IEEE Mountain Workshop on Adaptive and Learning Systems*, 2006, pp. 78–84.

[190] Moncion, T., Amar, P., and Hutzler, G., "Automatic Characterization of Emergent Phenomena in Complex Systems," *Journal of Biological Physics and Chemistry*, Vol. 10, 2010, pp. 16–23.

[191] Moore, C. and Mertens, S., *The Nature of Computation*, Oxford University Press, 2011.

[192] Mooz, H. and Forsberg, K., "Visualizing System Engineering and Project Management as an Integrated Process," *INCOSE International Symposium*, Vol. 7, No. 1, 1997, pp. 573–580, DOI: 10.1002/j.2334-5837.1997.tb02221.x.

[193] Morgan, C. L., *Emergent Evolution*, Second, Williams and Norgate, 1927.

[194] Morris, M. D. and Mitchell, T. J., "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, Vol. 43, No. 3, Feb. 1995, pp. 381–402, DOI: 10.1016/0378-3758(94)00035-T.

[195] Moshirpour, M., Mousavi, A., and Far, B., "Model Based Detection of Implied Scenarios in Multi Agent Systems," *Information Reuse and Integration (IRI), 2010 IEEE International Conference on*, Aug. 2010, pp. 63–68, DOI: 10.1109/IRI.2010.5558962.

[196] Moshirpour, M., Alhajj, R., Moussavi, M., and Far, B. H., "Detecting Emergent Behavior in Distributed Systems Using an Ontology Based Methodology," *2011 IEEE International Conference On Systems, Man, and Cybernetics (Smc)*, Oct. 2011, pp. 2407–2412, DOI: 10.1109/ICSMC.2011.6084038.

[197] Moshirpour, M., Mousavi, A., and Far, B. H., "A Technique and A Tool To Detect Emergent Behavior of Distributed Systems Using Scenario-Based Specifications," *22nd International Conference On Tools With Artificial Intelligence (Ictai 2010), Proceedings*, Vol. 1, Oct. 2010, pp. 153–159, DOI: 10.1109/ICTAI.2010.29.

[198] Müller-Schloer, C. and Sick, B., "Controlled Emergence and Self-Organization," *Organic Computing*, Vol. 21, Understanding Complex Systems, Springer Berlin / Heidelberg, 2008, pp. 81–103, DOI: 10.1007/978-3-540-77657-4_4.

[199] Mussington, B. D., *Concepts for Enhancing Critical Infrastructure Protection: Relating Y2K to CIP Research and Development*, tech. rep. MR-1259-OSTP, RAND Corporation, 2002.

[200] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, Second, John Wiley & Sons, Inc., 2002.

[201] Nagatani, T., "Nonlinear-Map Model for the Control of an Airplane Schedule," *Physica A: Statistical Mechanics and Its Applications*, Vol. 392, No. 24, 2013, pp. 6545–6553, DOI: 10.1016/j.physa.2013.08.076.

[202] Nagel, E., "Issues in the Logic of Reductive Explanations," *Teleology Revisited and Other Essays in the Philosophy and History of Science*, New York, Columbia University Press, 1979, Ch. 6, pp. 95–113.

[203] Nagpal, R., "A Catalog of Biologically-Inspired Primitives for Engineering Self-Organization," *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*, ed. by Marzo Serugendo, G., Karageorgos, A., Rana, O. F., and Zambonelli, F., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 53–62, DOI: 10.1007/978-3-540-24701-2_4.

[204] Nicolis, G. and Prigogine, I., *Exploring Complexity: An Introduction*, W. H. Freeman and Company, 1989.

[205] O'Connor, T., "Emergent Properties," *American Philosophical Quarterly*, Vol. 31, No. 2, Apr. 1994, pp. 91–104.

[206] O'Connor, T. and Wong, H. Y., "Emergent Properties," *The Stanford Encyclopedia of Philosophy*, ed. by Zalta, E. N., Spring 2012, 2012.

[207] O'Toole, E., Nallur, V., and Clarke, S., "Towards Decentralised Detection of Emergence in Complex Adaptive Systems," *Self-Adaptive and Self-Organizing Systems (SASO), 2014 IEEE Eighth International Conference on*, Sept. 2014, pp. 60–69, DOI: 10.1109/SASO.2014.18.

[208] Pfeifer, R. and Bongard, J., *How the Body Shapes the Way We Think: A New View of Intelligence*, MIT Press, 2007.

[209] Phelan, S. E., "A Note on the Correspondence Between Complexity and Systems Theory," *Systemic Practice and Action Research*, Vol. 12, No. 3, 1999, pp. 237–246, DOI: 10.1023/A:1022495500485.

[210] Polani, D., "Defining Emergent Descriptions by Information Preservation," *Unifying Themes in Complex Systems*, ed. by Minai, A. A., Braha, D., and Bar-Yam, Y., Springer Berlin Heidelberg, 2011, pp. 281–288, DOI: 10.1007/978-3-642-17635-7_34.

[211] Polanyi, M., *Knowledge and Being*, ed. by Grene, M., London: Routledge and Kegan Paul, 1969.

[212] Prokopenko, M., Boschetti, F., and Ryan, A. J., "An Information-Theoretic Primer on Complexity, Self-Organization, and Emergence," *Complexity*, Vol. 15, No. 1, Sept. 2009, pp. 11–28, DOI: 10.1002/cplx.20249.

[213] Ranque, P., Freeman, D., Kernstine, K., Lim, D., Garcia, E., and Mavris, D., "Stochastic Agent-Based Analysis of UAV Mission Effectiveness," *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2011-6956, Sept. 2011, DOI: 10.2514/6.2011-6956.

[214] Reynolds, C. W., "Flocks, Herds and Schools: A Distributed Behavioral Model," *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, New York, NY, USA: ACM, 1987, pp. 25–34, DOI: 10.1145/37401.37406.

[215] Ronald, E. M. A., Sipper, M., and Capcarrere, M. S., "Testing for Emergence in Artificial Life," *Advances in Artificial Life: 5th European Conference*, ed. by Floreano, D., Nicoud, J.-D., and Mondada, F., Vol. 1674, 1999, pp. 13–20.

[216] Ronald, E. M. A., Sipper, M., and Capcarrère, M. S., "Design, Observation, Surprise! A Test of Emergence," *Artificial Life*, Vol. 5, No. 3, 1999, pp. 225–239, DOI: 10.1162/106454699568755.

[217] Rouff, C., Vanderbilt, A., Hinchey, M., Truszkowski, W., and Rash, J., "Properties of a Formal Method for Prediction of Emergent Behaviors in Swarm-Based Systems," *SEFM '04: Proceedings of the Software Engineering and Formal Methods, Second International Conference*, IEEE Computer Society, 2004, pp. 24–33, DOI: 10.1109/SEFM.2004.1347500.

[218] Rouse, W. B., "Engineering Complex Systems: Implications for Research in Systems Engineering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 33, No. 2, May 2003, pp. 154–156, DOI: 10.1109/TSMCC.2003.813335.

[219] Ryan, A., "A Multidisciplinary Approach to Complex Systems Design," PhD thesis, The University of Adelaide, Mar. 2007.

[220] Ryan, A. J., "Emergence is Coupled to Scope, Not Level," *Complexity*, Vol. 13, No. 2, Nov. 2007, pp. 67–77, DOI: 10.1002/cplx.20203.

[221] Şahin, E., "Swarm Robotics: From Sources of Inspiration to Domains of Application," *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers*, ed. by Şahin, E. and Spears, W. M., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 10–20, DOI: 10.1007/978-3-540-30552-1_2.

[222] San Miguel, M., Johnson, J. H., Kertesz, J., Kaski, K., Díaz-Guilera, A., MacKay, R. S., Loreto, V., Érdi, P., and Helbing, D., "Challenges in Complex Systems Science," *The European Physical Journal Special Topics*, Vol. 214, No. 1, Nov. 2012, pp. 245–271, DOI: 10.1140/epjst/e2012-01694-y.

[223] Saunders, M. G. and Voth, G. A., "Coarse-Graining Methods for Computational Biology," *Annual Review of Biophysics*, Vol. 42, No. 1, 2013, pp. 73–93, DOI: 10.1146/annurev-biophys-083012-130348.

[224] Scheidler, A., Brutschy, A., Diwold, K., Merkle, D., and Middendorf, M., “Ant Inspired Methods for Organic Computing,” *Organic Computing: A Paradigm Shift for Complex Systems*, ed. by Müller-Schloer, C., Schmeck, H., and Ungerer, T., Vol. 1, Autonomic Systems, Springer Basel, 2011, pp. 95–109, DOI: 10.1007/978-3-0348-0130-0_6.

[225] Scott, D. W., *Multivariate Density Estimation*, John Wiley & Sons, Inc., 1992.

[226] Seth, A. K., “Measuring Emergence Via Nonlinear Granger Causality,” *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems, ALIFE 2008*, 2008, pp. 545–552.

[227] Seth, A. K., “Measuring Autonomy and Emergence Via Granger Causality,” *Artificial Life*, Vol. 16, No. 2, Mar. 2010, pp. 179–196, DOI: 10.1162/artl.2010.16.2.16204.

[228] Shah, A. P., Feigh, K. M., Pritchett, A. R., Kalaver, S. A., Jadhav, A., Holl, D. M., Bea, R. C., and Gilgur, A. Z., “Analyzing Air Traffic Management Systems Using Agent-Based Modeling and Simulation,” *6th USA/Europe Air Traffic Management Research and Development (ATM R&D) Seminar, 27-30 June 2005*, 2005.

[229] Shalizi, C. R., “Methods and Techniques of Complex Systems Science: An Overview,” *Complex Systems Science in Biomedicine*, ed. by Deisboeck, T. S. and Kresh, J. Y., Topics in Biomedical Engineering. International Book Series, 2006 2, New York: Springer, 2006, pp. 33–114, DOI: 10.1007/978-0-387-33532-2_2.

[230] Shalizi, C. R. and Shalizi, K. L., “Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences,” *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, ed. by Chickering, M. and Halpern, J., UAI ’04, Banff, Canada: AUAI Press, 2004, pp. 504–511.

[231] Sheard, S. A. and Mostashari, A., “Principles of Complex Systems for Systems Engineering,” *Systems Engineering*, Vol. 12, No. 4, Dec. 2009, pp. 295–311, DOI: 10.1002/sys.20124.

[232] Silverman, B. W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986.

[233] Silverman, B. W., “On the Estimation of a Probability Density Function by the Maximum Penalized Likelihood Method,” *The Annals of Statistics*, Vol. 10, No. 3, 1982, pp. 795–810.

[234] Simon, H., “Alternative Views of Complexity,” *The Sciences of the Artificial*, Third, Cambridge, MA: The MIT Press, 1996, Ch. 7.

[235] Sobol, I., “Global Sensitivity Indices for Nonlinear Mathematical Models and Their MonteCarlo Estimates,” *Mathematics and Computers in Simulation*, Vol. 55, 2001, pp. 271–280.

[236] Sperry, R. W., "A Modified Concept of Consciousness," *Psychological Review*, Vol. 76, No. 6, 1969, pp. 532–536, DOI: 10.1037/h0028156.

[237] Sudeikat, J. and Renz, W., "Toward Systemic MAS Development: Enforcing Decentralized Self-Organization by Composition and Refinement of Archetype Dynamics," *Engineering Environment-Mediated Multi-Agent Systems: International Workshop, EEMMAS 2007, Dresden, Germany, October 5, 2007. Selected Revised and Invited Papers*, ed. by Weyns, D., Brueckner, S. A., and Demazeau, Y., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 39–57, DOI: 10.1007/978-3-540-85029-8_4.

[238] Szabo, C. and Teo, Y. M., "Post-Mortem Analysis of Emergent Behavior in Complex Simulation Models," *Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM-PADS '13, Montreal, Canada: ACM, 2013, pp. 241–252, DOI: 10.1145/2486092.2486123.

[239] Teo, Y. M., Luong, B. L., and Szabo, C., "Formalization of Emergence in Multi-Agent Systems," *Proceedings of the 2013 ACM SIGSIM Principles of Advanced Discrete Simulation (SIGSIM-PADS '13)*, Montreal, Canada: ACM, 2013, pp. 231–240, DOI: 10.1145/2486092.2486122.

[240] Terano, T., "Exploring the Vast Parameter Space of Multi-Agent Based Simulation," *Multi-Agent-Based Simulation VII*, ed. by Antunes, L. and Takadama, K., Vol. 4442, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 1–14, DOI: 10.1007/978-3-540-76539-4_1.

[241] Tribus, M., *Thermostatics and Thermodynamics*, ed. by Hagerty, W. W., University Series in Basic Engineering, D. Van Nostrand Company, Inc., 1961, p. 649.

[242] Truszkowski, W., Hinchey, M., Rash, J., and Rouff, C., "Autonomous and Autonomic Systems: a Paradigm for Future Space Exploration Missions," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 36, 3, May 2006, pp. 279–291, DOI: 10.1109/TSMCC.2006.871600.

[243] Ulieru, M. and Doursat, R., "Emergent Engineering: a Radical Paradigm Shift," *International Journal of Autonomous and Adaptive Communications Systems*, Vol. 4, No. 1, Dec. 2011, pp. 39–60, DOI: 10.1504/IJAACS.2011.037748.

[244] Varela, F. J., Thompson, E., and Rosch, E., *The Embodied Mind: Cognitive Science and Human Experience*, MIT Press, 1993.

[245] Venables, W. N. and Ripley, B. D., *Modern Applied Statistics with S*, 4th, Statistics and Computing, Springer-Verlag New York, 2002, DOI: 10.1007/978-0-387-21706-2.

[246] Vicsek, T., "Complexity: The Bigger Picture," *Nature*, Vol. 418, July 2002, p. 131, DOI: 10.1038/418131a.

[247] Vrabic, R. and Butala, P., “Computational Mechanics Approach to Managing Complexity in Manufacturing Systems,” *CIRP Annals - Manufacturing Technology*, Vol. 60, No. 1, 2011, pp. 503–506, DOI: 10.1016/j.cirp.2011.03.050.

[248] Vrabic, R. and Butala, P., “Assessing Operational Complexity of Manufacturing Systems Based on Statistical Complexity,” *International Journal of Production Research*, Vol. 50, No. 14, 2012, pp. 3673–3685, DOI: 10.1080/00207543.2011.575098.

[249] Warwick, G., “ONR: Swarming UAVs Could Overwhelm Defenses Cost-Effectively,” *Aviation Week & Space Technology*, Apr. 2015.

[250] Weaver, W., “Science and Complexity,” *American Scientist*, Vol. 36, 1948, pp. 536–544.

[251] Weeks, A., “Neutral Emergence and Coarse Graining Cellular Automata,” PhD thesis, University of York, 2010.

[252] Welch, P. H., Wallnau, K., Sampson, A. T., and Klein, M., “To Boldly Go: an Occam-$\pi$ Mission to Engineer Emergence,” *Natural Computing*, Vol. 11, No. 3, Sept. 2012, pp. 449–474, DOI: 10.1007/s11047-012-9304-2.

[253] Wiener, N., *Cybernetics: Or Control and Communication in the Animal and the Machine*, Second, MIT Press, 1961.

[254] Wimsatt, W. C., “Aggregativity: Reductive Heuristics for Finding Emergence,” *Philosophy of Science*, Vol. 64, No. 4, Dec. 1997, pp. 372–384, DOI: 10.1086/392615.

[255] Winfield, A. F. T., Harper, C. J., and Nembrini, J., “Towards Dependable Swarms and a New Discipline of Swarm Engineering,” *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers*, ed. by Şahin, E. and Spears, W. M., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 126–142, DOI: 10.1007/978-3-540-30552-1_11.

[256] Winfield, A. F., Sa, J., Fernández-Gago, M.-C., Dixon, C., and Fisher, M., “On Formal Specification of Emergent Behaviours in Swarm Robotic Systems,” *International Journal of Advanced Robotic Systems*, Vol. 2, No. 4, 2005, DOI: 10.5772/5769.

[257] Wolfram, S., “Approaches to Complexity Engineering,” *Physica D: Nonlinear Phenomena*, Vol. 22, No. 1–3, 1986, pp. 385–399, DOI: 10.1016/0167-2789(86)90309-X.

[258] Wolfram, S., “Undecidability and Intractability in Theoretical Physics,” *Physical Review Letters*, Vol. 54, No. 8, Feb. 1985, pp. 735–738.

[259] Wolpert, D. H. and Macready, W. G., “Self-Dissimilarity: An Empirically Observable Complexity Measure,” *Proceedings of the International Conference on Complex Systems, Nashua, NH. Sep. 1997*, ed. by Bar-Yam, Y., 1997.

[260] Wright, W. A., Smith, R. E., Danek, M., and Greenway, P., "A Generalisable Measure of Self-Organisation and Emergence," *Artificial Neural Networks (ICANN 2001)*, ed. by Dorffner, G., Bischof, H., and Hornik, K., Vol. 2130, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 857–864, DOI: 10.1007/3-540-44668-0_119.

[261] Wulf, W. A., "Great Achievements and Grand Challenges," *The Bridge*, Vol. 30, No. 3/4, 2000, pp. 5–10.