Conceptually Robust Knowledge Generation in Early Stage Complex Design

by

Conner J. Goodrum

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in The University of Michigan
2020

Doctoral Committee:

Associate Professor David J. Singer, Chair
Associate Professor Matthew Collette
Professor Larry Seiford
Professor Nickolas Vlahopoulos

Conner J. Goodrum

cgoodrum@umich.edu

ORCID iD: 0000-0003-4679-0741

To my family.

ACKNOWLEDGEMENTS

My time at UofM has allowed me to learn and grow personally, professionally, and academically. So much of this is attributed to the people around me. Specifically, I would like to thank:

My advisor, David Singer, for his mentorship inside and outside of my research.

My committee members, for their time and comments in writing this dissertation.

Kelly Cooper at the Office of Naval Research for providing the funding that made this research possible.

My lab mates, for their help in editing this thesis, and for fostering a collaborative and enjoyable work environment throughout my time in the lab.

My family, who have always pushed me to pursue my interests and take on new challenges, and who have always been there to listen, offer advice, and support me in all my endeavors.

My fiancée, Sadie, for her love and understanding, her constant encouragement, and for continually inspiring me to be a better version of myself.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

ABSTRACT

Modern engineering design is a complex, path dependent process in which knowledge is generated for decision making through time. While this in and of itself poses an immense challenge, designers must also coordinate their efforts across a number of design disciplines to produce a converged design in the presence of exogenous factors such as shifting requirements and changing information landscapes. The presence of such factors all too often requires large portions of the design to be revised, leading to excessive rework, design churn, and integration failures in the design process. This has spurred an interest in the notion of conceptual robustness, however the approaches to date remain focused on the product being developed rather than on the knowledge generated to create the product. As such, little has been done to understand the conceptual robustness of a design process, and focusing on knowledge structures provides a novel method for designers to effectively manage these complex design tasks. This helps ensure design activities are robust against future changes in the design landscape.

The framework presented in this thesis, the Knowledge-Information (K-I) Framework, utilizes a multi-layer network approach to represent how information sources are translated into knowledge structures and is analyzed using a number of novel entropy metrics. The framework considers the information-knowledge interplay at two scales: (1) at a local, intra-discipline level and (2) at a global, design integration level. These multi-layer networks are analyzed to reveal conceptual robustness insights for individual disciplines, and throughout the process of integrating disparate sources of knowledge between disciplines. The framework provides a novel perspective of what

it means for a knowledge structure to be robust, and enables emergent design failures to be identified earlier on in the design process.

The utilization and analysis of the K-I Framework enable design knowledge to be explored in the context of conceptual robustness. First, novel entropy-based temporal metrics are developed which leverage concepts from both Network Theory and Information Theory to provide new perspectives to analyze knowledge and information structures over the course of a design activity. Second, the theoretical basis of the K-I Framework is outlined, along with the processes by which local and global structures are developed and the way in which they interact. Third, a case study is presented which highlights how different calculation strategies yield different local knowledge structures in relation to calculating the same desired knowledge entity. Finally, an additional case study is presented which focuses on capturing global knowledge integration dynamics in performing an Analysis of Alternatives (AoA) study of a naval distribution system. The results of the case studies are used to draw conclusions about the conceptual robustness of design knowledge generation.

CHAPTER I

Introduction

1.1 Motivation & Relevant Research

The advances of modern technologies and analysis methods have led to the creation of new engineering marvels. Computational and simulation advances have improved the accuracy and capabilities of engineering analyses, and have had a significant impact on what can be achieved. While these advances have addressed many technical challenges, they have also created new challenges in design development. The integration of large software suites and integrated tool-sets has reduced the transparency of engineering analyses and blurred the line between what is information and what is knowledge. Significant time, effort and investment has been put toward improving design analysis capabilities, but little has been done to understand how best to utilize these tools to generate useful information and novel design knowledge over the course of a design activity. Modern engineering design is a complex, path dependent process in which knowledge is generated for decision making through time, often leading to emergent design failures if not managed or understood properly.

These recent advances have provided novel approaches to product-centric problem solving, but unfortunately lack the ability to address the *wicked problem* that defines the landscape of modern engineering design. Originally coined by Webber and Rittel (1973), a wicked problem is a classification of problem that directly contradicts the ideas of traditional problem solving activities. Wicked problems were later generalized by Conklin (2006) as adhering to six defining characteristics. In the context of design, these characteristics are:

1. The product is not understood until after the formulation of a solution.
2. Design problems have no stopping rule.
3. Solutions to design problems are not right or wrong.
4. Every design problem is essentially novel and unique.
5. Every solution to a design problem is a "one shot operation."
6. Design problems have no given alternative solutions.

These six characteristics defining wicked problems are unfortunately inherent aspects of conducting novel engineering design activities, and represent significant difficulties to designers in determining appropriate requirements (Andrews 2012). Design as a wicked problem means that design is understood through *a-posteriori* (derived from experience), *tacit* (contained within the mind), and *procedural* (habitual processes) sources of knowledge.

A-posteriori knowledge is gained by first having an experience, then using logic and reflection to derive understanding from it. As traditional design knowledge is based on experiences, this source of knowledge is often subjective and open to interpretation. The subjectivity of reflecting on a design activity can make it difficult to find root causes of emergent behaviors and design failures. Additionally, traditional design knowledge is tacit, making the act of design comparable to playing an instrument or speaking a foreign language, in that designers "know more than they can tell" (Polanyi 2009). While designers can communicate certain aspects about a product or process, this is only a small subset of the knowledge, and it is impossible to fully communicate the notion of conducting design effectively to others without them doing it. This hinders the ability to communicate lessons learned to future design activities, which will necessarily be novel and unique. To better account for conducting future novel design activities, design is often framed using procedural knowledge, which details the steps or activities to perform a task or job. However, while the details of how to conduct the process is useful for formulating different design approaches, it is not the same as knowledge of how to actually do it. As procedural knowledge is unique to different individuals (or organizations), this can lead to ineffective design management, and a disconnect between engineering tasks performed and the engineering tasks needed for proper information generation. Operating within the confines of these sources of knowledge leads to the complexity of conducting a successful design activity, and can drastically impact the likelihood of a successful design outcome.

The issues associated with knowledge generation within the context of the wicked problem have been framed as necessary evils to be worked around. For this reason, engineering design has been focused on creating a successful product (product-centric design), and often neglects the knowledge used to create that product (knowledge-centric design). While a product-centric view of design is helpful in many scenarios, it is insufficient to understand emergent design failures, such as *design churn*, *rework*, and *failure to integrate* (Braha and Bar-Yam 2007). These concepts can be understood as:

> **Design Churn**: Yassine et al. (2003) states that design churn occurs when "the total number of problems being solved (or progress being made) does not reduce (increase) monotonically as the project evolves over time". Design churn arises when solving a problem leads to the creation of further problems, which result in the design's emergent path oscillating around the design path required to yield a successful outcome. This reduces the ability to measure and manage design progress and results in inefficiencies leading to increased development

times, cost overruns, and decreased problem solving abilities of design teams.

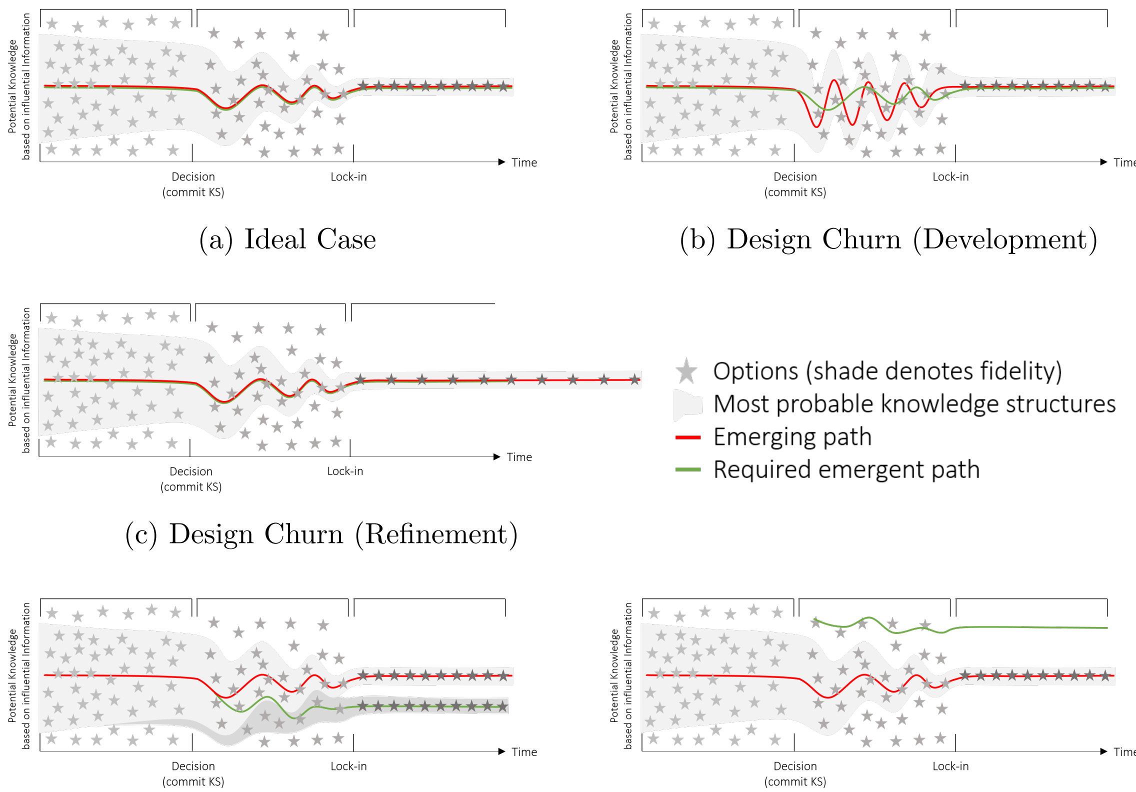
Excessive Rework: Rework is "the repetition of tasks due to the availability of new information generated by other tasks, such as changes in input, updates of shared assumptions, components, boundaries, or the discovery of errors ... As this missing or uncertain information becomes available, the tasks are repeated to come closer to the design specifications or goals" (Braha and Bar-Yam 2007). While this concept is a fundamental consequence of iterative design processes, it becomes problematic when rework becomes excessive and results in design churn.

Failure to Integrate: Conducting a successful complex engineering design activity means integrating a large number of components or systems into a final product. From a knowledge perspective, new requirements or new knowledge associated with these components or systems must integrate with the existing knowledge structure (Shields and Singer 2017). This can lead to integration issues in relation to both the product and knowledge required to define that product. In the best case, these integration issues result in design churn and or rework, but often lead to an inability to continue with current design activities or infeasibility of the final design.

Emergent design failures are fundamentally aspects associated with the act of designing the product, rather than in the product itself (Shields 2017). Over the course of a design activity, engineering analyses are conducted to learn about the design problem, and provide additional data and information for further design refinement. The resultant data and information is then used to inform future engineering analyses, which inherently creates path dependencies (Page 2005) throughout a design activity. This cyclical process is repeated until a final product is created. The connected and interdependent parts which comprise the product define the *product structure*.

More importantly, the relationships between the ideas, concept elements and evidence used to yield that product define the *knowledge structure*. This can be understood as a macro-level mental model associated with a product, created using knowledge generated within and integrated across design teams in the design activity. In order to fully understand and prevent emergent design failures, a fundamental shift is required toward a knowledge-centric perspective of design as opposed to the common product-centric perspective.

Figure 1.1 provides context for how each of the emergent design failures can affect decision-making paths in a design activity over time. The x-axis represents time, and the y-axis represents the potential knowledge which can be gained based on influential information. The stars in the image denote potential knowledge structures, and the shade of the stars represent the associated fidelity of the information associated with the knowledge structure (darker shade means more fidelity). Initially, a large number of potential knowledge structures exist, but are of relatively low fidelity. Later on in the design activity, there are a much more limited number of knowledge structures, and they are of much higher fidelity. A decision is a commitment to a knowledge structure at a point in time, which both restricts the possibilities of future knowledge structures, and limits the influence of information. The aforementioned path dependent process of conducting engineering analyses creates an emergent path (shown in red) which represents the current direction of the design activity. For wicked problems which do indeed have a solution, there must exist some 'required path' (shown in green) which represents an ideal sequence of engineering activities required to reach the design goal. It should be noted that for wicked problems the required path (green) is only known once the solution has been created, and the current emergent path (red) is constantly evolving. Thus, effectively managing the wicked problem should ensure the current path is best aligned with the required path over the course of a design activity, and can best realign the two paths when they



(a) Ideal Case

(b) Design Churn (Development)

(c) Design Churn (Refinement)

(d) Rework

(e) Failure to Integrate

Figure 1.1: The relation of path dependencies to Emergent Design Failures

divert as early as possible, with the minimal amount of effort.

In the ideal case (1.1a) the design decision results in an emergent path (red line) which is aligned with the required path to reach the goal (green line). In this case, the emergent path creates the "correct" knowledge structure by uncovering the appropriate relationships between engineering analyses and in the correct sequence, thus avoiding emergent design failures. If a knowledge structure is not robust, design churn can occur in the development of future knowledge structures (1.1b), in which inefficiencies in the process make it difficult to understand the required decision path. Additionally, design churn can occur in the refinement of a knowledge structure after lock-in has occurred (1.1c), which results in increased time in conducting design analyses. A decision which leads to rework (1.1d) can occur if the trajectory of the decision path diverges from the required path after the decision has been made. As the paths were congruent up to the decision point, rework can be understood as "retracing steps" to re-align the decision path with the required path. This should be realized as early as possible, so as to minimize the number of steps needing to be retraced to realign the paths. This is not possible in the case of failure to integrate (1.1e), in which there is no possibility to reach the required path from the current one due to external factors, or incompatible knowledge structures.

While wicked design problems are essentially "unsolvable", the probability of a successful design outcome can be maximized by mitigating emergent design failures through applying the idea of *conceptual robustness* to design knowledge, which is a relatively unexplored research area. Conceptual robustness, first coined by (Chang, Ward, et al. 1994), is a widely used term with no universally accepted definition. Robustness in the context of an engineering system is the ability to resist or cope with changes or perturbations without adapting its initial stable configuration. Robust design has been defined as: "An approach to designing a product or process that emphasizes reduction of performance variation through the use of design techniques that reduce sensitivity to sources of variation" (Dehnad 1989). Some of the first most notable contributions to the realm of product-centric views of robustness in design were from Genichi Taguchi, whose work includes three primary statistical contributions pertaining to robustness:

1. Designers should minimize the deviation of performance parameters from desired target values, to ensure quality and satisfaction of design requirements.

2. Products should be robust against manufacturing and environmental variations (physical noise), to ensure the product can meet design requirements over a wide range of manufacturing and operational environments.

3. Partial-factorial, orthogonal arrays should be used to create experiments rather than changing parameters individually in order to minimize the number of re-

quired experiments, and to capture the average effects of design parameters in the presence of noise.

This approach to incorporating robustness in designing products is based on statistical design of experiments (Cornell and Khuri 1987; Box, W. G. Hunter, and J. S. Hunter 1978), and it has gained notoriety in the realm of quality engineering and optimization schema (Chang and Ward 1995). These notions of robustness aim at managing either the variability of design parameters or reducing their sensitivity to the outcome of the product, and are limited to later design stages after conceptual design has been completed. These notions do not account for the underlying body of knowledge and information used to design the product, but focus on the product itself. In this respect, they account for robustness of the product, but do not address what it means for a design process to be conceptually robust.

Further work has been done to extend the Taguchi Parameter Design practices into concurrent engineering processes (Chang, Ward, et al. 1994; Chang and Ward 1995) and in doing so spurred the definition of "conceptually robust decisions", which are decisions that are robust against variations in the part of the design done by other team members, or "conceptual noise". While this extension is more comprehensive, it is still product-centric, and is limited to systems composed of well-defined components, of relatively low complexity. A later definition of conceptual robustness by Bernstein (1998): "refers to both the product's ability to function in a wide range of manufacturing and operating environments and to the design's ability to adapt to modifications which might be implemented later in its development." While this definition also remains product-centric, it extends previous definitions by taking a more holistic approach by incorporating the concept of future design developments into the definition. More recently, Singer, Doerry, and Buckley (2009) provided the definition: "Conceptual robustness is achieved when engineering decisions concerning one aspect of a design remain valid in the face of design decisions made in other aspects of the design." This definition is closest to being knowledge-centric, although it is not stated explicitly. What is required is a definition of conceptual robustness that is focused on the proper definition of design as a whole.

If design is defined as the act of generating knowledge for decision making through time, then a proper definition of conceptual robustness must include aspects of knowledge structure development and analysis. Additionally, the definition must relate the temporal knowledge structure evolution to the final creation of a successful product. The above definitions describe some of the key attributes of conceptual robustness in general terms, but must be applied in the context of knowledge structures if one hopes to capture how the development of knowledge provides a conceptually robust landscape. This spurs the following, proposed, knowledge-centric definition of conceptual robustness:

> "A conceptually robust knowledge structure is a knowledge structure that, through time, maximizes the likelihood that its current and future evolution is resilient to exogenous factors."

In other words, a conceptually robust knowledge structure enables the greatest potential to integrate future knowledge entities, and mitigates the impact of exogenous factors of the current structure, through time. Herein lies the difficulty, as the knowledge structure should support the integration of future knowledge while mitigating the impacts on previously integrated knowledge. A conceptually robust knowledge structure should provide a structure which maximizes the number of possibilities where new knowledge can be integrated into the existing structure, and yet should minimize the impact of future knowledge by limiting the amount of knowledge entities which need to be restructured in the presence of changes triggered by exogenous factors.

Complex design requires the development of knowledge at a number of levels. At a local level, design teams may create or utilize tools to develop individual knowledge structures specific to their design discipline. At a global level, the communication between teams integrates local sources of knowledge into a global structure required to yield a final product. The definition of conceptually robust knowledge structures holds for both the local and global levels of knowledge structure development. At the individual level (for example, within a design discipline), knowledge structures should be resilient to exogenous factors arising from changes made in other areas of the design. At the global level, the integrated knowledge structure should be resilient to exogenous factors such as changes to requirements. These two changes are tightly coupled - as exogenous factors applied to the global level will impact local structures' abilities to successfully accommodate the changes. Conversely, local changes (such as utilizing higher-fidelity tools) will impact the ability to create robust global structures. As the design progresses, both levels must be carefully monitored to minimize the likelihood and impact of emergent design failures.

A primary issue in how conceptual robustness is understood arises from a fundamental misunderstanding of the differences between data, information, and knowledge. For example, the US Navy conducts and manages design through product models (NAVSEA 2012), which are used to integrate definition tools with physics-based analysis tools to populate and explore a feasible design space. This design space is created using thousands of point based designs semi-automatically at relatively low fidelity, to characterize design tradeoffs and ensure that the 'correct' design is selected (Chalfant 2015; Kassel, Cooper, and Mackenna 2010). This approach implies that designers can gain new knowledge about design interdependencies by exploring tradeoffs and that including more designs provides more accuracy about this knowledge. In reality this approach cannot provide new knowledge; it extracts data and information embedded in the tool, which is a product of the knowledge structure used to create it. The interdependencies uncovered in analyzing the results of such an analysis are a product of the knowledge structure of the software tools used to create them, and thus exploration of these tools simply uncovers the relationships in the tools themselves (Sypniewski 2019). Each point design this tool generates is additional data, and exploration of how the data relates to each other yields information. Thus, conceptual robustness in this lens is related to selecting an appropriate design, rather than generating a conceptually robust design from the outset. The widespread acceptance that these tools yield novel knowledge is fundamentally flawed, and proves to be a large hurdle in exploring conceptual robustness from a knowledge perspective.

Significant work has been done to understand the role of knowledge generation in design activities using knowledge structures (Laxton 1969; Goldschmidt and Weil 1998; Cross 2001). Most notably, the framework developed by Shields (2017) formalized a network method for mapping the growth of design knowledge structures over time using actions and decisions. This work highlighted that emergent design failures can be mitigated either by increasing the predictability of design outcomes, or by limiting unexpected interdependencies by controlling the knowledge structure around design drivers. While this research has provided a promising step toward understanding the knowledge-centric measures of a design activity, it does not formally incorporate the process of utilizing data and information into knowledge structure generation. Additionally, while this approach provides a new perspective on how design outcomes are created from a knowledge structure, it does not present quantitative methods of relating how actions and decisions change the probability of certain design outcomes. This will be a crucial aspect in understanding and measuring the conceptual robustness of knowledge structures.

Addressing the issues associated with the wicked problem necessitates a new perspective of conceptual robustness which focuses on the process of generating knowledge over time to develop a product, rather than on the product itself. This requires the ability to track and understand the use of data and information and how they relate to the generation of design knowledge over time. Understanding the generation of knowledge over time re-frames the wicked problem into a more manageable problem. Ideally, the 'wickedness' of the wicked problem could be removed if designers could derive design solutions without needing to develop them (a-priori knowledge), if all aspects of design knowledge and a design activity could be totally communicated between designers (explicit knowledge), and if designers had full knowledge *of design* rather than just *how to design* (declarative knowledge). However, the wickedness of the problem means these knowledge sources are by definition unattainable; the best that can be done is to dynamically track the evolution of knowledge over time to prevent emergent design failures early on, and to understand the mechanisms under which they are likely to arise. This requires metrics to understand the dynamics of the evolution of knowledge from information and data sources to provide warning signs to designers and managers of potential emergent design failures. Current approaches to understanding conceptual robustness are inadequate in capturing the dynamics between how data and information relate to the generation of knowledge, and focus on the robustness of the product rather than the robustness of the knowledge used to create that product. The current knowledge-centric metrics are yet to be analyzed in the context of conceptual robustness. Currently there are no tools to qualitatively understand and guide the efforts to tackle the wicked problem, and fundamentally understand the root cause of emergent design failures. A knowledge-centric approach to conceptual robustness will lead to general strategies to approach wicked design problems, quantify the impact of the design process on the probability of yielding design outcomes, and minimize the probability of emergent design failures.

1.2 Research Scope

Three primary research questions are addressed in this work:

1. How can data, information, and knowledge be represented to model and analyze a design activity?

2. How can data, information, and knowledge be quantified in the context of a design activity?

3. How can conceptual robustness be understood from a knowledge-centric perspective?

The scope of the presented thesis is focused on answering these fundamental questions through the creation of a novel framework in which to analyze knowledge-centric conceptual robustness. The framework utilizes networks to represent the dynamics between data, information, and knowledge over the course of a design activity and leverages aspects of information theory to quantify their evolution. This is used to understand aspects of a knowledge structure which are robust against the presence of exogenous factors. The following section outlines how these issues are addressed in this thesis.

1.3 Organization of the Thesis

This remainder of this dissertation is divided into five chapters, and are organized as follows:

- Chapter II presents the required background information pertaining to Network Theory and Information theory which are utilized throughout the thesis. In addition to the existing metrics, this chapter also presents a number of hybrid metrics which leverage aspects from both disciplines used throughout the work to understand conceptual robustness.

- Chapter III introduces the formal multi-layer network framework (the K-I Framework) at the center of this work. This chapter outlines how data, information, and knowledge are represented in each layer of the framework. Additionally, the inter- and intra-layer dynamics are outlined to illustrate how information is used to generate knowledge through time at local and global levels of design.

- Chapter IV provides a case study focused on local knowledge structure generation. This case study utilizes the processes outlined in Chapter III to elucidate how different local knowledge structures can be developed based on different strategies of determining the same unknown bit of knowledge. The results of this case study are used to discuss the conceptual robustness of each approach, and draw conclusions about characteristics defining robust local knowledge structures.

- Chapter V expands the discussion of conceptual robustness to the global layers of the K-I framework by providing a case study conducting an Analysis of Alternatives (AoA) design activity. This case study considers how the interaction of local knowledge structures can be used to create global information and knowledge structures. Leveraging the metrics developed in this thesis, the results provide a means of understanding conceptual robustness at a more macro scale.

- Chapter VI details the contributions of this thesis and topics for future work.

CHAPTER II

Background

This research relies heavily on the areas of Network Theory and Information Theory. Network theory fundamentally studies the structures of systems of interest using relations between entities of the system. This provides significant insight into understanding how the system is structured, the dynamics of the system at different scales, and explaining system behavior. On the other hand, Information Theory studies aspects such as the quantification, storage, and communication of information between entities. While these disciplines have developed (and continue to develop) independently of one another, there are many areas of research which have integrated the two. This thesis leverages the advantages of both theories to provide new and novel insights to studying design. This section is included to familiarize the reader with the key concepts of each area used throughout this work, and which are used to create new metrics unique to this thesis (Section 2.3).

2.1 Network Theory

Much of this explanation comes from Goodrum, Shields, and Singer (2017) and Goodrum, Taylordean, and Singer (2018). Most simply, *networks* (or *graphs*) are abstract representations of systems using points (*nodes*) and lines (*edges*). In network terminology, nodes represent entities and edges represent relationships between them.

These edges can be directed or undirected, depending on the nature of the relationship. The versatility of these abstractions enable networks to be widely applied to many social and scientific disciplines. Their versatility and ability to represent a wide array of entities of varying fidelity make networks powerful tools to study design. Abstracting design to network space allows for new insights to be derived, otherwise limited by more traditional approaches. Most notably, several insights can be gained from the study of how nodes and edges relate to one another - the network's *structure* - which can be quantified in a number of ways. As such, this section briefly outlines key network terminology used throughout the remainder of this work to study network structure. For a comprehensive review of networks, see Newman (2010).

2.1.1 PageRank

PageRank is a centrality metric used to quantify the importance of a node in a network based on the network's structure. This metric, originally developed by Google, is a critical component to many search engines website ranking technology, and has since been applied to the study of a wide variety of network types. The PageRank algorithm assigns a numerical value in the range $[0, 1]$ to each node in the network, which represents the likelihood that a random walk will arrive at that particular node. Thus, the sum of all nodes' PageRank values equals 1. The algebraic definition of PageRank is given by Equation (2.1):

$$\mathbf{PR} = (\mathbf{I} - \alpha \mathbf{A}\mathbf{D}^{-1})^{-1}\mathbf{1} \qquad |\mathbf{PR}| = 1 \tag{2.1}$$

where $\mathbf{PR}$ is a column vector of node PageRank values, $\mathbf{I}$ is the identity matrix, $\mathbf{A}$ is the network's adjacency matrix, and $\mathbf{D}$ is a diagonal matrix with elements $D_{ii} = \max(k_i^{out}, 1)$ (for a more detailed derivation, see Newman (2010)). This metric is useful in quantifying node importance, and also proves useful in its use as an equivalence group in studying the Topological Entropy (TE) metric presented in Section 2.3.1.

2.2 Information Theory

Information theory studies the transmission, extraction, and utilization of information. But what *is* information? At its most basic level, information is the reduction of uncertainty about what an entity is. In information theory, information is understood as a sequence of symbols, yielding the quantitative definition:

> "[Information is] a mathematical quantity expressing the probability of occurrence of a particular sequence of symbols, impulses, etc. as contrasted with that of alternative sequences" (Oxford 2019).

Information theory began with the seminal paper by Shannon (1948), in which he illustrated that information could be quantified with absolute precision, and that essentially all communication modes could be encoded in 'bits' (Aftab et al. 2001). Information theory has since been studied extensively, and is widely used in the study of complex, adaptive, and artificially intelligent systems. Although much work has been done to further Shannon's original work, information theory continues to be an emerging field of study in finding novel applications and analysis methods.

A widespread metric used to study information is *Entropy*. Entropy is a measure of disorder, or randomness in a system. In information theory, entropy describes the rate of transfer of information in a particular message. High entropy corresponds to a large amount of disorder, while zero entropy corresponds to a completely certain outcome. Based on the above definition of information, entropy measures this amount of randomness using the probability of occurrence of the symbols. A number of different entropic measures have been developed to be suitable for various applications of quantifying information. This research utilizes the common information-theoretic concept of entropy to quantify the transmission of information over knowledge networks, as a means to understand the communication pathways, efficiency, and robustness of such networks. The applicable measures of entropy used in this work are included in the remainder of this section.

2.2.1 Shannon Entropy

The Entropy measure originally developed by Shannon (1948), now referred to as *Shannon Entropy (SE)*, quantifies the average information content in a message based on the probability of occurrence of each of the characters. Mathematically, Shannon Entropy is defined by Equation (2.2).

$$H(\mathbf{x}) = -\sum_{i=1}^{N} p_i log_2(p_i) \tag{2.2}$$

where $\mathbf{x}$ is message with N characters, and p_i is the probability of occurrence of character i. Note that for a message with a completely certain outcome (i.e. $p_i = 1$ for some value of i, and all other probabilities of 0), $H(\mathbf{x}) = 0$, and conversely, $H(\mathbf{x})$ is maximized when the probability distribution is uniform (i.e. when $p_i = 1/N \ \forall i$). For a derivation of this metric, see Shannon (1948).

While SE provides a mathematical formulation to quantify the average information content of a message, its interpretation relies heavily on how the probabilities are computed. In many cases, the probabilities associated with characters are based on the relative frequency of that character in the message, although this is not the only method of calculating probabilities. In fact the probabilities used in Equation (2.2) can be dependent on any *equivalence class*, in which elements belong to the same equivalence class if those elements share some pre-specified trait. The fraction of elements within this equivalence class can be used to determine probabilities. For example, consider the message shown in Equation (2.3). This message contains three unique values $(1, 2$ and $3)$, and each is associated with two colors: red (R) and blue (B). In this instance, values and colors are two equivalence classes based on the same message. If the probabilities were based on the value equivalency class, the equation would yield the result shown in Equation (2.4), and if based on the color equivalence class , the result would be that shown in Equation (2.5).

$$\mathbf{x} = \{(1, R), (2, R), (1, B), (3, B)\} \tag{2.3}$$

$$H_{\text{val}}(\mathbf{x}) = -0.5 \log_2(0.5) - 0.25 \log_2(0.25) - 0.25 \log_2(0.25) = 1.5 \text{ bits} \tag{2.4}$$

$$H_{\text{col}}(\mathbf{x}) = -0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1.0 \text{ bits} \tag{2.5}$$

Both representations of probabilities are correct mathematically; however, they reveal different insights. The result of Equation (2.4) says that on average, a person would need to ask 1.5 questions to correctly determine the value of any of the four entities in the message. Conversely, to determine the color of any of the entities, the person would only need to ask one question to be certain. Thus, equivalence classes play a critical role in defining the amount of information computed by the entropy metric, and must be selected carefully to be aligned with the question being asked.

Equivalence classes have been applied to networks based on a number of network-metric based equivalence classes. For example, the degree distribution has been used to quantify information content in a network, based on the frequency of node degree (Rashevsky 1955). There are additional metrics that have been applied, and the Topological Entropy metric developed in this work (Section 2.3.1) utilizes node PageRank (Section 2.1.1) as an equivalence class.

2.2.2 Cumulative Residual Entropy

An extension of Shannon Entropy (SE) was developed by Rao et al. (2004), called *Generalized Cumulative Residual Entropy (CRE)* which has a number of advantages in the application to this research, and is noted as having applications in the realm of Reliability Engineering. The main difference between CRE and SE is that CRE utilizes the cumulative distribution of a random variable, rather than the traditional probability distribution used in Equation (2.2). Additionally, CRE extends the properties of SE to consider random values with continuous distributions, and allows the values to be both positive and negative. This is advantageous when considering the Value Entropy metric developed in Section 2.3.5. For a random vector X in $\mathcal{R}^N$ equation for CRE is presented in Equation (2.6):

$$CRE(X) = -\int_{-\infty}^{\infty} P(|X| > \lambda) \log_2 P(|X| > \lambda) d\lambda \tag{2.6}$$

where $X = (X_1, X_2, ..., X_N)$, $\lambda = (\lambda_1, ...\lambda_N)$, $|X| > \lambda$ means $|X_i| > \lambda_i$.

To demonstrate the value of CRE, consider the following example adapted from Rao et al. (2004). Consider two fair six sided dice, A and B, which take possible values of $\{1, 2, 3, 4, 5, 6\}$ and $\{1, 2, 3, 4, 5, 100\}$, respectively. Each dice is fair, so the probability of rolling any of the possible values is 1/6 for both dice. Imagine that the dice values represent payoff schemes. Applying SE to dice A and B (using Equation (2.2) and value as the equivalence class) leads to the same value of 2.58 bits of information for both dice. In applying Equation (2.6) to dice A and B, the values are 2.07 and 22.67, respectively. Thus, CRE provides a metric to meaningfully quantify differences between numerically different random variables, suggesting the information content in the payoff schemes across the two dice are drastically different - something that SE fails to capture. The CRE metric highlights that the payoff associated with die B is more uncertain than that of die A.

2.3 Developed Entropy Metrics

The concepts introduced in Sections 2.1 and 2.2 are leveraged to create three entropy metrics used to provide insight into the robustness of knowledge structures in the K-I framework (introduced in Chapter III). Each metric was developed to measure different dynamics within the resultant networks. A *Topological Entropy* metric was developed to measure the information content of network *structure*, a *Target Value Entropy* metric was developed to measure the information content associated with the calculated *values* of a target node, and a *Data Status Entropy* was developed to consider the *growth of uncertainty* in a calculation approach over time. These three developed metrics are explained in more detail in this section.

2.3.1 Topological Entropy

A *Topological Entropy (TE)* metric has been developed to quantify the information contained by a network structure. While centrality metrics such as PageRank (Section 2.1.1) have been used to analyze the structure of networks, and entropic measures of information theory such as Shannon Entropy (Section 2.2.1) have been used to quantify information content, a novel metric is required to quantify the structural complexity and information of a network. This metric must thus integrate the centrality metrics of network theory with entropic metrics of information theory to be useful in this research.

As outlined in Section 2.1.1, the PageRank centrality metric ranks nodes in order of importance based on the structure of the network. Additionally, this metric has the convenient feature that they are bounded between 0 and 1, and the sum of all PageRank node values across the network sums to one Equation (2.1). Hence, the PageRank vector can be utilized in the same way as a probability distribution in Equation (2.2) to uniquely quantify the information content in a network. Combining these concepts yields the mathematical expression for Topological Entropy (TE):

$$\mathrm{TE}(\mathcal{G}) = -\sum_{n \in \mathcal{G}} (\mathrm{PR}_n) log_2(\mathrm{PR}_n) \tag{2.7}$$

where n is a node in network $\mathcal{G}$, and PR_n is the PageRank value of node n. As this metric is general, it can be applied to any network, and tracked over time.

2.3.2 Unknown Fraction

Although TE quantifies the information content contained in the structure of a network, it does not address the calculability of nodes in the network. To account for this, a binary *data status* indicator is assigned to each node in the network, which is one if and only if that node has data, and is zero otherwise. Tracking the portion of nodes with no data over time provides an intuitive method of understanding how the design is progressing in terms of unknown parameters becoming known, and extends the analysis beyond just the structural growth. To understand the unknown portion of the network, the *Unknown Fraction (p_0)* metric is proposed. This metric is defined as:

$$p_{0,t}(\mathcal{G}_t) = \frac{n_{0,t}}{N_t} \tag{2.8}$$

where $\mathcal{G}_t$ is network with N_t nodes, and $n_{0,t}$ are the number of nodes with data status 0, all at time t. UF takes a maximum value of 1 when all nodes in the network are unknown (all nodes have *data status* 0), and takes a value of 0 when all nodes contain data (*data status* of 1). Tracking this metric over time reveals if the number of unknowns in the network is monotonically decreasing, which can be used to identify design churn. The compliment of UF, referred to as the *Known Fraction (p_1)*, represents the proportion of the network which contains data, by considering the number of nodes with *data status* values equal to 1. Note that $n_{0,t} + n_{1,t} = N_t$, and $p_{0,t} + p_{1,t} = 1 \; \forall t$. The Known Fraction metric is defined as:

$$p_{1,t}(\mathcal{G}_t) = 1 - p_{0,t} = \frac{n_{1,t}}{N_t} \tag{2.9}$$

For simplicity, subsequent sections refer to $p_{0,t}(\mathcal{G}_t)$ and $p_{1,t}(\mathcal{G}_t)$ as $p_{0,t}$ and $p_{1,t}$, respectively.

2.3.3 Data Status Entropy

A mathematical solution to an equation can only be calculated if all of the dependent variables are defined by values. Understanding the process of gathering data from information sources to populate these dependent variables (used to calculate solutions) requires a novel metric. This metric, called the *Data Status Entropy (DSE)*, is used to quantify growth of a network's calculability over time when intermediate calculations are required. To account for this, the known and unknown fraction metrics from Section 2.3.2 are leveraged. One proposed entropy metric is presented in Equation (2.10):

$$\mathrm{H}_t(\mathcal{G}_t) = -(p_{0,t}) \log_2 (p_{0,t}) - (p_{1,t}) \log_2 (p_{1,t}) \tag{2.10}$$

where $\mathcal{G}_t$ is the network, and $p_{0,t}$ and $p_{1,t}$ are defined in Equations 2.8 and 2.9, respectively. Note that the equation is not defined for cases where $N = 0$, meaning at least one node must exist in the network in order for the metric to be applied.

This formulation considers both known and unknown fractions, and is bounded between 0 and 1. While this formulation has a number of useful qualities, it also presents a number of issues. First, note that the function is maximized when $(p_{0,t}, p_{1,t}) = (0.5, 0.5)$. This indicates entropy is maximized when the fraction of knowns is equal to the number of unknowns. Since there is equal likelihood of a randomly selected node being known or unknown in this case, it is intuitive that this should maximize the metric.

Also note that the function is symmetric about this maximum. The function takes a minimum value of 0 at $(p_{0,t}, p_{1,t}) = (0, 1)$ and $(1, 0)$, which occurs when either all nodes are calculable, or incalculable. Both cases represent complete certainty of the *data status* of a randomly selected node, yet have significantly different contexts in understanding the calculability of a network. What is required is a metric which differentiates the two scenarios, and enables further understanding of the network dynamics. In order to address these issues, a reformulation is proposed:

$$\text{DSE}_t(\mathcal{G}_t) = \begin{cases} 1 & \text{if} \quad p_{1,t} = 0 \\ -(p_{1,t})log_2(p_{1,t}) & \text{if} \quad p_{1,t} > 0 \end{cases} \tag{2.11}$$

where the parameters of the equation have been previously described.

The function in Equation (2.11) addresses the issues created by Equation (2.10) by focusing on the entropic contribution of only the calculable portion of the network. This avoids the issues presented with the symmetry of the previous metric. If the network is entirely incalculable (i.e. $(p_{0,t}, p_{1,t}) = (1, 0)$) then the entropy is assigned its maxmimum value. Conversely, if the network is entirely calculable ($(p_{0,t}, p_{1,t}) = (0, 1)$), the entropy is minimized. This is the metric utilized throughout the remainder of this thesis.

By viewing the DSE time series, the growth in uncertainty of a calculation approach can be determined by tracking how calculation steps progress over time, and how that leads to the growth of uncertainty in terms of a network's calculability. This can be used to ensure all calculation steps progress toward calculating the target node value appropriately. The metric also aids in identifying design churn and rework activities based on the data status of revisions made to knowledge and information structures.

2.3.4 Normalized Data Status Entropy

Additional insight can be gained from determining how the number of known or unknown variables change over the course of a design activity by comparing the change in DSE to that at the initial state. This metric, called *Normalized Data Status Entropy (NDSE)*, is given by Equation 2.12, and expresses the DSE at a given point in time in reference to the initial DSE of the network:

$$\mathrm{NDSE}_t = 1 - \frac{DSE_t}{DSE_i} \tag{2.12}$$

This metric takes a value of 0 at the initialization of the network, and approaches 1 as more parameters become known relative to the starting point. An increase in NDSE means more variables are becoming known, while a decrease means the ratio between known and unknown variables is approaching 50/50. Note that unlike the previous entropy metrics, NDSE can take negative values if initial timesteps lead to more unknown variables than that at the outset of the design activity. Similar to the insights gained by the metrics proposed in Equations (2.8) and (2.11), this metric helps identify design churn and rework activities, but provides a relative comparison to the starting condition of the design.

2.3.5 Target Value Entropy

The previous entropy metrics have focused on quantifying the information content associated with the structure and calculability of networks. However, they do not address the information content associated node values. Consider a static network structure which is fully calculable. The previous metrics could identify the uncertainties of these scenarios, but do not consider the way in which node values change as the result of calculations being performed on the structure. As was presented in Section 2.2.2, CRE is an appropriate metric to use to quantify information content within continuous distributions, and is sensitive to different sampled values. This makes it an appropriate entropic measure of calculated values, to quantify both information content and uncertainty. This spurred the development of a *Target Value Entropy (TVE)* metric. This metric can also be tracked over time (as more target values are calculated), and can be used to quantify and compare the uncertainty of calculated values of different analysis methods. The equation for the TVE is:

$$\mathrm{TVE}_t(V) = -\int_{-\infty}^{\infty} P(V > v) \log_2 P(V > v) dv \tag{2.13}$$

where $V = \{V_0, ..., V_t\}$ is the message of calculated target node values up to time t.

TVE considers the history of node values, and applies an entropy metric which accounts for changes in uncertainty over time. Static node values over time represent certain time histories, thus resulting in zero TVE. Similarly, node values which change will lead to a growth in TVE, as the distribution of observed values will flatten. There is no upper bound on TVE, and the magnitude of the growth is dependent on both the magnitude of the value and the sequence of previous values. Hence TVE is unit-dependent.

TVE provides a useful means to identify the sources of rework and design changes. Observing how node values change over time will highlight potentially problematic variables, and when considered in conjunction with the aforementioned metrics will enable designers to better understand potential sources of non-robust design activities.

2.3.6 Differential Target Value Entropy

A subtle reformulation of the TVE metric developed in Section 2.3.5 provides an additional perspective as to the associated uncertainty of a string of values within a node. This metric, called *Differential Target Value Entropy (DTVE)* focuses on the amount a value changes, rather than on the value itself. While TVE provide insight into the uncertainty in a string of values, DTVE provides a relative measure of entropy growth based on the differences of sequential values. DTVE is given by the following equation:

$$\mathrm{DTVE}_t(V) = -\int_{-\infty}^{\infty} P(\Delta V > v) \log_2 P(\Delta V > v) dv \tag{2.14}$$

where $V = \{V_0, ..., V_t\}$ is the message of calculated target node values up to time t and $\Delta V_T = \{(V_1 - V_0), ..., (V_t - V_{t-1})\}$ is the message of the sequential differences in calculated target node values up to time t.

DTVE obeys the same trends as those described for TVE while allowing designers to gain an alternative understanding of uncertainty. By focusing on the magnitude of value *changes* rather than the values themselves, the metric captures uncertainty growth relative to how much a variable has changed previously. This provides designers with an additional lens through which to view potentially problematic variables. Similar to TVE, values which do not change will have zero DTVE, and those which change will exhibit a growth. Additionally, the metric is more sensitive to immediate identification of variable value changes, which will become more clear in the case studies presented in this thesis.

CHAPTER III

The K-I Framework

This chapter introduces the formal multi-layer network framework (the K-I Framework) to capture the dynamics of how bodies of information lead to the generation of knowledge structures through time. A preliminary overview of the K-I framework is presented in Figure 3.1, and will be explained in more detail throughout this chapter. The framework considers the knowledge-information (K-I) interplay at two scales: at a local level (Section 3.1), representing design agents or disciplines (shown in the top two layers in Figure 3.1), and at a global level (Section 3.2) -

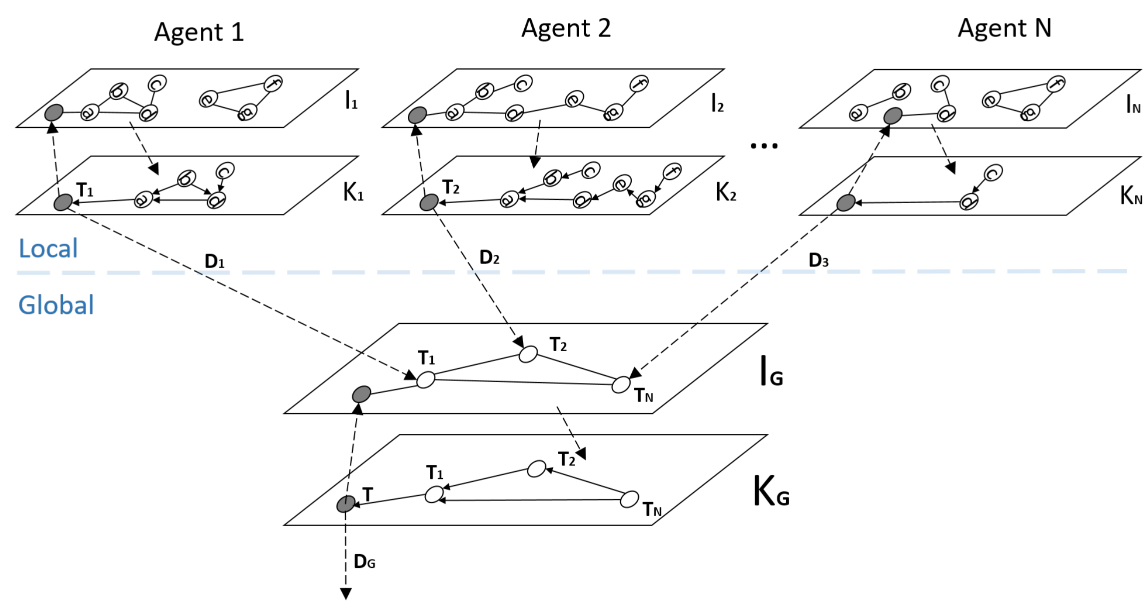


Figure 3.1: Overview of the multi-layer K-I Framework.

developing and integrating information and knowledge across agents (shown as the bottom two layers in Figure 3.1). This chapter also outlines the nomenclature and definition of framework-related terms used throughout the research. The networks describing knowledge structures form the basis for the information theory metrics to be applied, which provide insight to information flows and the robustness of various design approaches.

3.1 Local K-I Layers

This section outlines the discipline-level knowledge-information interplay to create local knowledge structures. Section 3.1.1 outlines how local information structures are represented in the local information layer, and Section 3.1.2 outlines the dynamics of how these local information structures are used to create local knowledge structures.

3.1.1 Local Information Layer

The local information network layer contains the information resources required to populate the local knowledge layer. The network represents a number of data elements, and their relation to one another. Given this definition, data entities are represented as nodes, and functional dependencies between data elements are represented by directed edges. An edge in the local information network is drawn from a variable X to a variable Y if and only if X is used to calculate Y. That is to say directed edges point to dependent variables from the variables used to calculate them. In this way, mathematical formulas can be represented as networks.

Consider as an example the equation $A = BCD$. This equation represents that there are functional dependencies between the variables A, B, C, and D. From a data perspective, this equation simply relates how the data entities are related, by describing the dependencies between variables. If solving for the variable A, directed edges would be drawn from variables B, C, and D, pointing to A. However, rearranging this equation to solve for any of the other variables would result in similar structures.

The variables A, B, C, and D are only symbolic representations of data entities - that is to say the equation holds true regardless of what the values of each variable are. Under the same reasoning, nodes are simply symbolic representations of data - a node can be populated in the information network as a data entity without containing any values. This separation between data structure and values is an important aspect, as they are related in how the data is utilized. The variable being solved for dictates the resulting structure, and conversely the variables that have values (contain data), dictate which variable can be calculated. To account for the first issue, the equation is rearranged to solve for each variable, and a directed network is created for each. The information network is created by merging each of these individual networks. By rearranging the equation to solve for each variable, the resulting information network captures all possible data dependencies, and thus defines how data within the network is organized (information). To solve the second issue, each node in the information network contains a *value* place-holder and a *data status* binary indicator, which is one if and only if the data entity contains one or more values, and is zero otherwise. This elucidates which variables contain data, and which don't, and also translate to what can be calculated very rapidly. Figure 3.2 shows the resulting network for this example case.

The representative network shown in Figure 3.2 describes the relations between data sources, and thus represents the information structure within the single equation. This network exists in a single local information layer. Each mathematical equation, or relation of data entities, represents a different information source. Even though equations may contain the same variables, they may be structured differently, and thus should be represented by different information networks. Each individual information network is located in the local information layer, and thus, the local in-

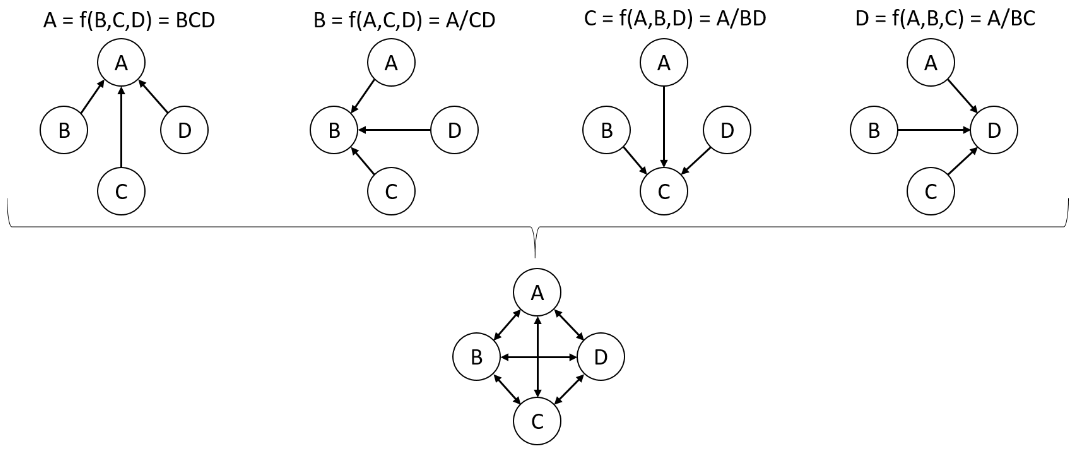


Figure 3.2: Resulting information network from data relations in $A = BCD$

formation layer is composed of a large number of small components, corresponding to different information structures. There are a very large, but finite, number of equations. While it is not feasible to create singular information networks for all existing equations, this may not be necessary. It is sufficient to populate a specific local information layer with all known, applicable equations to a given design discipline. The importance lies on the strategies of selecting these information structures which dictates how the resulting knowledge structures are developed.

3.1.2 Creating the Local Knowledge Layer

The local knowledge layer tracks the relation and evolution of knowledge elements over time in relation to a single design discipline, or design agent. While the local information layer represents the dependencies of data entities given a mathematical approach, the knowledge layer represents how that information is utilized. Once information is used to perform an analysis, it becomes a part of the knowledge structure. This is beneficial as it reduces the extremely large body of available information in the information layer to only that information which is used to build a knowledge structure of a given approach, effectively building a unique knowledge structure from the large body of available information.

The basic flow chart of knowledge structure evolution is presented in Figure 3.3. A knowledge structure is comprised of two fundamental types of nodes: *target nodes* and *supporting nodes*. Target nodes represent *what knowledge you need*, and supporting nodes represent *what knowledge you have*. From a network perspective, target nodes are those nodes which initially contain no values and need to be determined, and supporting nodes are nodes used to determine target nodes. Supporting nodes need not contain data initially (they can also be unknown), in which case they must be determined through intermediate calculations. As knowledge structures are developed in relation to seeking knowledge about some entity, a knowledge structure is initialized with a target node, with no supporting nodes (Figure 3.4a). Connecting supporting nodes to the target node is the basic dynamic of creating a knowledge structure, and is completed by seeking information from the information layer by way of a *hypothesis* and integrating that information structure into the knowledge structure by way of an *action*. At any point, the knowledge structure can be committed by a *decision* (Figure 3.4).

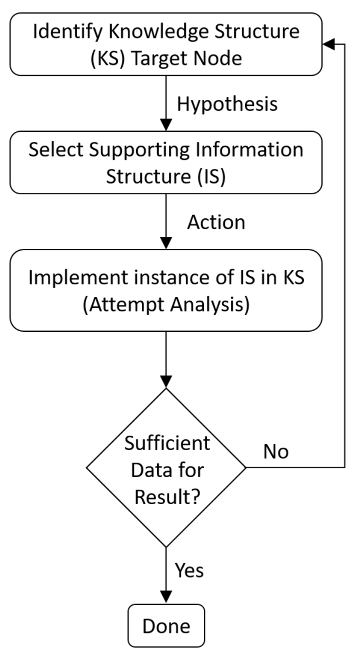
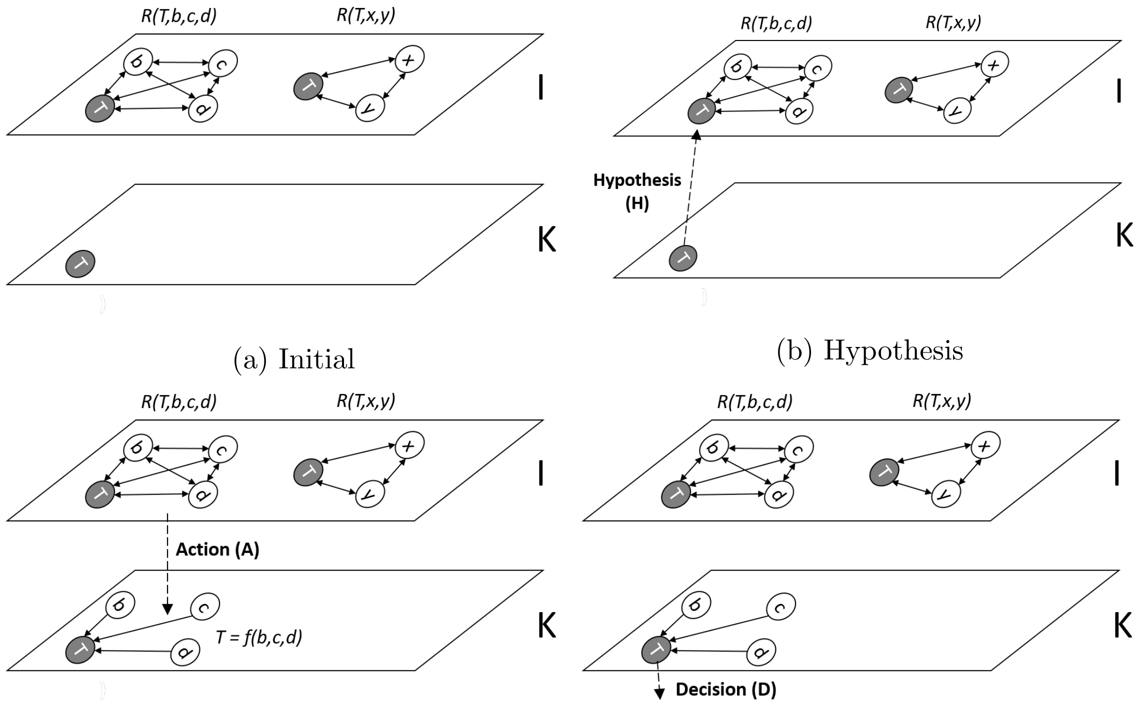


Figure 3.3: Basic algorithm describing knowledge structure growth.



(a) Initial

(b) Hypothesis

(c) Action

(d) Decision

Figure 3.4: Illustration of the four basic steps in generating a local knowledge structure

A hypothesis links what knowledge is required in the knowledge layer (the target node) to the information able to provide that knowledge in the information layer (Figure 3.4b). In this sense, it is the act of finding a suitable information network in the information layer which contains the same symbol as the target node - as this represents an information network which is structured to provide the required knowledge. In practice, this is represented as an inter-layer edge, from the target node in the knowledge layer to a similar node in an information network in the information layer. The end result of a hypothesis is a link between the target node in the knowledge layer to a network component in the information layer - effectively connecting the required knowledge to a body of information which is hypothesized to yield that knowledge. As there are likely a wide range of information structures which contain the ability to calculate the target node, posing a hypothesis is an abductive process which does not guarantee a successful outcome, and hence captures designer strategies of approaching the design problem.

An action is the act of utilizing the body of information to perform an analysis. Within the context of the K-I framework, it is the process which imports the directed relations of the supporting nodes into the knowledge layer, thus building the knowledge network (Figure 3.4c). For a given hypothesis (selecting a body of information), performing an action is as follows: for the target node in the information layer, follow all edges pointing to it, and find the nodes at the beginning of that edge. Then, import all the traversed directed edges and the nodes into the knowledge layer. This effectively represents a directional relationship for the information structure, representing an instantiation of the equation in which the target node is being solved for.

A decision commits the knowledge network structure at a point in time, by translating the data from the analysis in the knowledge structure (Figure 3.4d). A commitment to a result by proxy also commits the knowledge structure used to create that result. Multiple decisions can be made on a single knowledge structure.

3.2 Global K-I Layers

The framework presented in Section 3.1 describes how information structures are used to develop knowledge structures relating to some target entity. This target entity, the 'target node', represents *what we want to know* and provides the required context for finding supporting information sources and utilizing them as supporting knowledge. While this holds for a single design analysis - the reality is that complex engineering design activities involve many design agents (teams, individuals, departments, etc) conducting design analyses which need to be integrated into a larger context to yield a solution. Each discipline will have different required knowledge entities, and thus will undergo each of their own processes of building their own knowledge structures. While this process has been shown to have a large impact on the conceptual robustness of a calculation approach - the integration of the individual knowledge structures will be a critical activity in building a macro-knowledge structure relating to the final integrated design solution.

In addition to the local knowledge and information layers described in Section 3.1, there exist two additional global layers in the K-I framework: the *global information* and *global knowledge* layers. The purposes of these layers are to the capture the integration dynamics of design knowledge across design agents (disciplines), and how that relates to a macro-perspective of design knowledge required for a greater design goal. The aforementioned local knowledge and information layers now represent intra-discipline knowledge generation activities, while the global layers now capture the inter-discipline knowledge integration dynamics.

Sections 3.2.1 and 3.2.2 describe how the Global Information and Knowledge layers are represented in the K-I Framework, respectively.

3.2.1 The Global Information Layer

The global information layer encapsulates the information entities required to integrate the local knowledge structures toward the greater design goal. As was described in Section 3.1.2, a decision commits the current state of a local knowledge structure at a point in time, as well as the target node value. Local knowledge structures are not shared between teams - rather design agents communicate data and information between one another in the form of results from engineering activities. While teams may provide context for why certain data are needed, their local knowledge structure are *tacit*, meaning they are unable to be communicated fully. As such, the global information layer captures the interdependencies between each design team's local knowledge structures as a result of communicating required or calculated entities between decisions made by teams. As such, the global information layer represents functional dependencies between design groups, by capturing the dynamics of communicated or negotiated information entities. It is through the organic development of the global information layer that local knowledge structures are related to the global knowledge structure.

The global information layer represents significantly different types of information than that of the local information layers. As was discussed in Section 3.1.1, the local information layers represent all potential information resources required to build a local knowledge structure. These information sources represent undirected relations between data entities in the forms of equations, tools, or specific analysis types. These information structures are selected by the design agent through hypotheses and implemented through actions, in an attempt to relate a target node to supporting knowledge entities. However, in the case of a design integration activity there are no existing information structures which can be simply 'selected' to integrate information from design agents. Global design information is far more specific to the design activity being conducted, and information is generated as the design activity progresses. Design processes address sequences in which information should be generated, but do not necessarily predicate 'structures' of information. As such, the global information layer captures the generation and refinement of macro-design information as a result of the sequence in which design teams conduct their analyses and make design decisions, rather than from the selection of pre-existing information structures.

As design agents develop and utilize their own discipline-specific knowledge structures, they may require the results of other disciplines' knowledge structures to conduct their own analyses. These results are often communicated between teams using engineering deliverables such as reports or documents, or through the transfer of files within a design integration software. Herein lies a subtle yet important aspect of inter-agent communication - agents do not communicate their entire local knowledge structures, but rather communicate their data requirements or results, perhaps with context as to why they require that data. This does not communicate the underlying local knowledge structure, but is more focused on the inputs and outputs of their local knowledge structures. As the functional dependencies between data entities define information structures, the relation of inter-discipline communicated data entities defines the global information structure. The global information structure is then used to create a global knowledge structure. The ability to capture the interdependencies between data elements in the global information layer will thus play a critical role in the conceptual robustness of a design activity, and will be a product of not only what parameters are communicated between agents, but also the sequence in which they are communicated. It is the task of a design integration activity to develop (and understand the development of) design information through the analyses of design agents, which are used to develop a global knowledge structure.

The global information layer is defined by nodes which represent data entities, and directed edges which represent functional dependencies between data elements as a function of inter-discipline communication. Similar to the local information structures, nodes are placeholders for data entities, and contain both a *value* placeholder and a *data status* binary indicator. An important difference between the edges in the global and local information layers is that edges in the local layers are predefined undirected relations between data entities, whereas in the global layer they are projected directed edges based on dependency pathways using the inter-layer edges between local knowledge structures.

To understand the process of creating global information edges, consider Figure 3.5. Note that this figure has omitted the local information layers and the global knowledge layer, and focuses on the dynamics between local knowledge layers and the global information layer. This assumes the agents have already completed the process of building their local knowledge structures in relation to some target node (Section 3.1). The local knowledge structures may have some entities which are

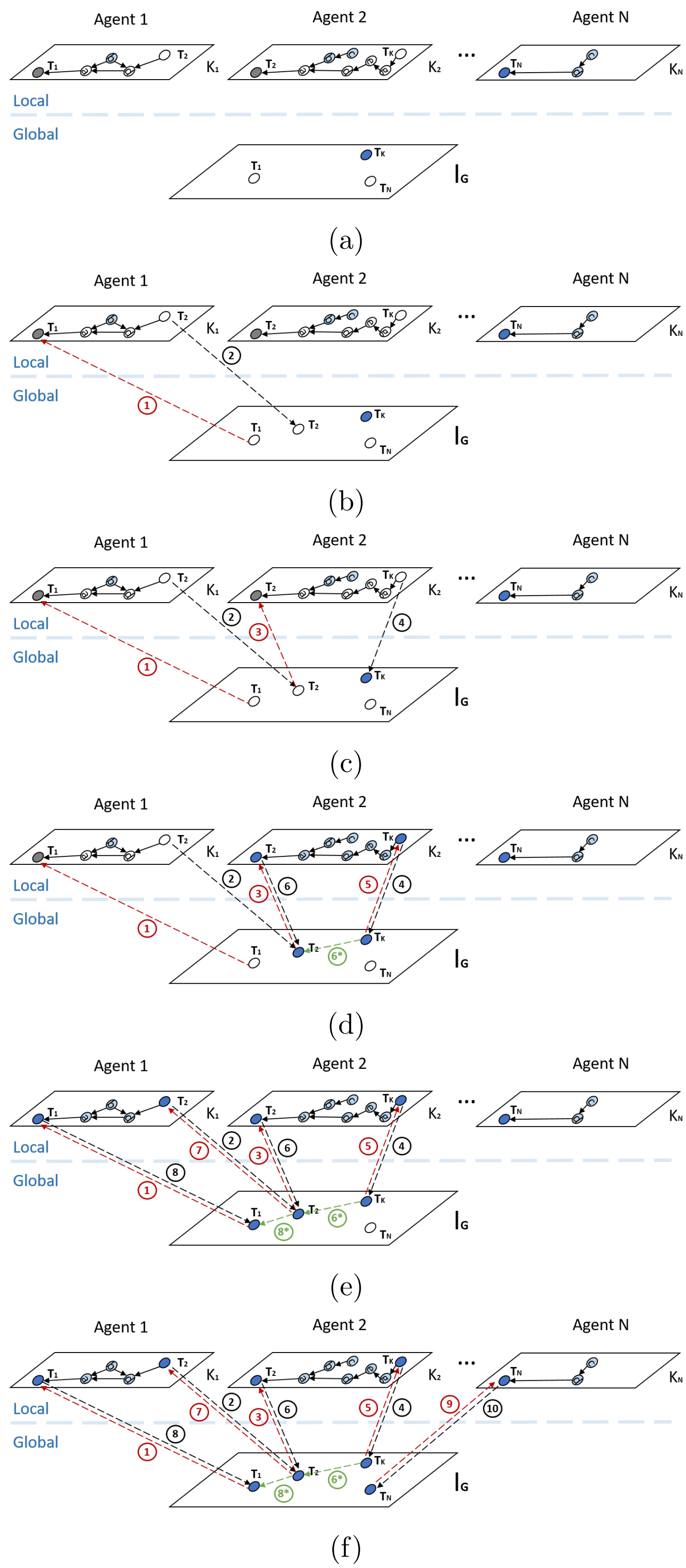


Figure 3.5: Global information structure development from local knowledge layers.

known (shown in blue) and some which are unknown (shown in white). The global information layer is initialized with the nodes in the global knowledge layer (Figure 3.5a), defining what entities are known and which are being sought out, shown in blue and white, respectively. The transmission of these nodes from global knowledge to global information provides context for what analyses the agents need to conduct to build the global information structure. Note that in this example the target nodes in the in local knowledge structures coincide with initialized nodes in the global information layer.

In this hypothetical example, the agents begin building the global information structure by first solving for $\mathbf{T_1}$ (Figure 3.5b). An edge (edge 1) is drawn from the selected node ($\mathbf{T_1}$) to a knowledge structure containing the same symbol, hypothesized to yield the desired result. In this case, the $\mathbf{T_1}$ symbol is contained by *Agent 1* - but note that a number of agents may contain the same symbol, and thus the selection of an agent may not be the only feasible decision path. *Agent 1* determines the required inputs to their knowledge structure, and determines that $\mathbf{T_2}$ is required to yield the appropriate result. Thus, the agent has two ways forward: The agent may begin the process of growing their local knowledge structure by (1) seeking an appropriate information structure (hypothesis) from their local information layer and implementing an action, or (2) communicating the required knowledge entity from their local knowledge layer to the global information layer to be determined by another agent. In this case, *Agent 1* performs option (2), and communicates the unknown $\mathbf{T_2}$ knowledge entity to the global information layer, by drawing edge 2.

The process continues by now seeking an agent to populate the newly created supporting global information node $\mathbf{T_2}$ with a value (Figure 3.5c). The same procedure is conducted as that of *Agent 1*, only now *Agent 2's* local knowledge structure is now selected. *Agent 2* requires $\mathbf{T_K}$ as an input to their knowledge structure, so rather than populate a new node in the global information layer for another agent to decide, edge 4 is drawn to the known node to be incorporated into their local knowledge structure. The known global information node $\mathbf{T_K}$ is integrated into the local knowledge structure by the procedure shown in Figure 3.5d. After edge 5 transmits the value from the global information layer to *Agent 2's* local knowledge structure, the $\mathbf{T_2}$ target node becomes calculable, and is communicated to the global information layer through edge 6. Upon updating the data status of $\mathbf{T_2}$ in the global information layer, all other global information nodes are queried to determine if a path exists through any local knowledge structures to global node $\mathbf{T_2}$. In this case, the only existing path is from $\mathbf{T_K}$ to $\mathbf{T_N}$ through *Agent 2's* local knowledge structure as a result of inter-layer edges 5 and 6. Hence, edge projection 6* is drawn in the global information layer - effectively illustrating the functional dependence of $\mathbf{T_2}$ on $\mathbf{T_K}$.

After $\mathbf{T_2}$ has been determined in the global information layer, the same process is conducted for *Agent 1* to conduct their analysis (Figure 3.5e). Similarly, the projected edge 8* is drawn as a result of the path from $\mathbf{T_2}$ to $\mathbf{T_1}$ through *Agent 1's* local knowledge structure, and inter-layer edges 7 and 8.

The final unknown node in the global information layer is $\mathbf{T_N}$, and is populated using the process shown in Figure 3.5f. *Agent N* is selected to populate this node (edge 9). *Agent N's* local knowledge structure is independent of inputs from any other agents, and as such the calculated knowledge entity is communicated to the global information layer in a single step by drawing edge 10. Note that drawing this edge does not result in any additional paths to other global information nodes, and as such, no additional projected nodes are drawn.

The resultant global information structure illustrates the functional dependencies between the initially included nodes that were previously unknown. The presence of the projected path $\{\mathbf{T_K}, \mathbf{T_2}, \mathbf{T_1}\}$ indicates that the unknown parameter $\mathbf{T_1}$ is dependent on $\mathbf{T_K}$ through the intermediate variable $\mathbf{T_2}$. The absence of a projected path to $\mathbf{T_N}$ highlights its independence from the other data. These dependencies have been uncovered through the inter-agent communication dynamics.

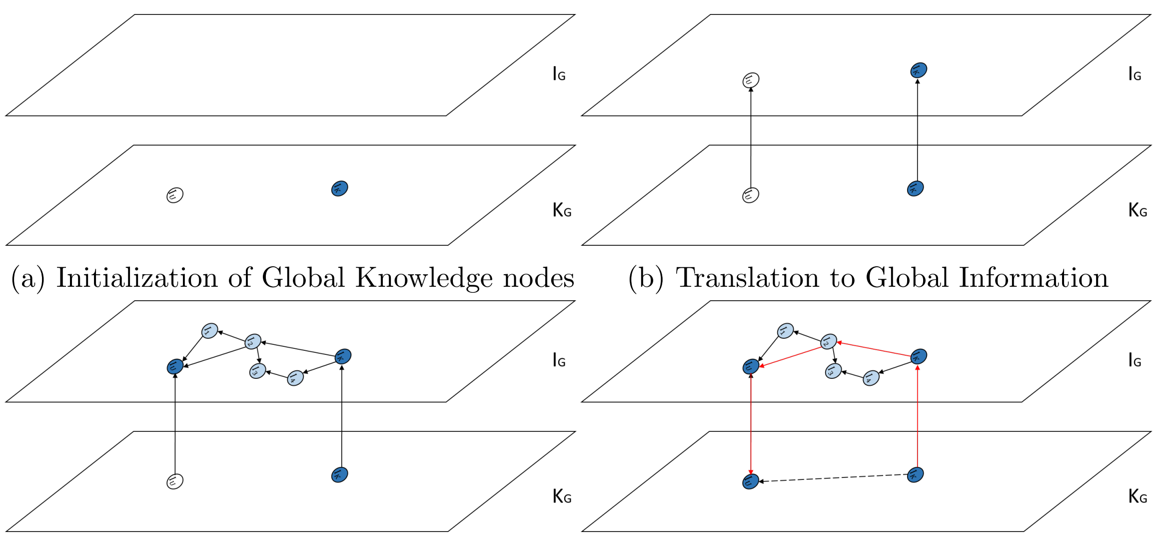
While the example depicted in Figure 3.5 illustrates the process of creating global information structures, it is important to note that the presented sequence of conducting the operations is not unique. For example, the procedure for calculating $\mathbf{T_N}$ could have been conducted first. Furthermore, instead of initially solving for the unknown data of $\mathbf{T_1}$ in the global information layer, the known entity $\mathbf{T_K}$ could have been initially utilized by *Agent 2* to conduct their analysis. While the same resultant information structure would have been calculated independent of sequence in this trivial example, for more complicated examples, the time-dependent dynamics of uncovering global information interdependencies will almost certainly be influenced by the sequence of operations. This presents a significant opportunity to uncover robust inter-agent communication strategies.

A similar process to that described in Section 3.1 will be utilized to create a global knowledge layer using the created global information structure. This global knowledge layer will provide a more high-level knowledge structure of the integrated design activity, and will provide insights toward the final design outcomes as a result of the interdisciplinary knowledge integration.

3.2.2 The Global Knowledge Layer

The global knowledge layer of the K-I framework encapsulates the integrated knowledge structure of a design activity, by relating a global target node (or set of global target nodes) to other supporting knowledge entities. The global knowledge layer obeys many of the same properties as the local knowledge layers (Section 3.1.2), but interacts with the global information layer rather than the local information layers. The global knowledge layer is comprised of a number of knowledge entities which are known (for example parameters which arise from requirements), and knowledge entities which are unknown (what parameters about the design need to be determined). Thus, it is the purpose of the design activity to not only make these unknown knowledge entities known, but also (and perhaps more importantly) to understand the interdependencies between global knowledge elements - relating how the known elements impact the unknown elements. Uncovering these relations not only elucidates how the known elements impact those that are unknown, but also how the unknown parameters are related to one another. Understanding both types of interdependencies will fully define the global knowledge structure - and will be a result of both local design agent knowledge generation and the inter-agent communication pathways. Thus, the global knowledge layer accounts for various levels of design activities, and will be impacted by not only the robustness of local knowledge structures, but also the development of the global information structure as a result of communication pathways between agents.

The process of creating the global knowledge layer is depicted in Figure 3.6. At the beginning of the design activity, the global knowledge layer is initialized with



(a) Initialization of Global Knowledge nodes (b) Translation to Global Information

(c) Development of Global Information (d) Creation of Global Knowledge Structure

Figure 3.6: Creation of Global Knowledge Structure from Global Information Structure. Known nodes (*data status* = 1) are shown in blue, and unknown nodes are shown in white. Dark blue indicates a target node, light blue indicates supporting information entities.

some number of known and unknown knowledge entities (Figure 3.6a). As is the case with the other layers in the framework, nodes contain *value* and *data status* placeholders, which differentiate the known and unknown nodes. The known knowledge entities describe design parameters which are known a-priori (usually as a result of requirements, or design targets). The unknown knowledge entities define aspects of the integrated design that are being investigated. These unknown entities are required as they define the knowledge-seeking activities of the design agents by prescribing the goals (defining the target nodes) of the analyses conducted at the local levels. Without the presence of the unknown nodes, there is no ability to allocate tasks to the design agents.

After initialization of the nodes in the global knowledge layer, the second step in the process is to initialize these knowledge nodes in the global information layer (Figure 3.6b). This procedure translates the known and unknown knowledge to nodes in the global information and provides context for the local agents to guide their engineering activities.

Over the course of the design activity, the global information layer grows based on the communication between design agents, as well as the projected dependence pathways between the target information nodes and generated supporting information (Figure 3.6c). Note that at this point, the previously unknown knowledge node becomes known in the global information layer through the analysis and communication between agents.

Upon communicating this target node in the global information layer back to the global knowledge layer, an edge is drawn between two nodes in the knowledge layer if there exists a path through the global information structure between the two knowledge nodes (Figure 3.6d). The directed edge points to dependent knowledge from the knowledge used to create it. This projected, directed edge thus captures the functional dependencies between global knowledge nodes, based on how the global information structure was developed by the inter-agent design effort. This process is repeated for each node in the knowledge layer by examining if a path exists between any two knowledge nodes. In this way, the edges in the global knowledge layer uncover the relations between the known and unknown knowledge entities at a macro-scale based on the framework's inter-agent dynamics captured in the global information layer.

CHAPTER IV

Local Knowledge Structure Case Study

A representative case study was developed to demonstrate the creation, dynamics, and analysis of the K-I framework. The case study is based upon four approaches to calculating an average volume from a database of previously built oceanographic vessels from Marine Structures Design Laboratory (2019). This data contains the principal dimensions of the constructed vessels: Length (L), Beam (B), Draft (T), and Block Coefficient (C_B), and is displayed in Table 4.1. Using the principal dimensions, the underwater volume can be calculated as the product of these principal dimensions.

The case study shows that the simple task of calculating an average value can have significant impacts on the way information is utilized, as well as the predictability, robustness, and structure of the resulting knowledge structures. Although a simple

Table 4.1: Oceanographic ship data used for case study

Oceanograpic Ship	L	B	T	C_B
AGOR 16	69.75	12.92	4.45	0.538
Atlantis II	59.52	13.53	4.92	0.537
Chas. Darwin	62.50	14.40	5.11	0.539
Endeavor	50.30	10.31	5.34	0.500
Littlehales(T-AGS51)	58.96	12.82	4.02	0.551
Maury (T-AGS39)	141.73	20.54	8.49	0.564
Melville (AGOR14)	69.19	14.42	4.65	0.518
Pathfinder (T-AGS60)	93.09	16.33	5.35	0.460
Protea	73.75	15.36	4.70	0.551
Researcher (OSS-03)	78.86	15.77	4.93	0.457
Robert Conrad (AGOR3)	58.96	11.12	4.28	0.427
Silas Bent (AGS26)	80.82	14.69	4.59	0.472
Stalwart (T-AGOS1)	63.24	12.16	4.21	0.560
Thomas Thompson	77.66	14.93	5.41	0.462

multiplication may seem like a trivial calculation, the results across the different methods yield vastly different knowledge structures, and highlight the different ways in which information is used to build these knowledge structures.

The four cases considered (representing different approaches to building knowledge structures) are summarized below. Sections 4.1.1 - 4.1.4 provide more detail of each case, and describe the process of generating each knowledge structure from the information sources. Section 4.1.5 presents a comparison of the resultant knowledge structures from each case. Section 4.2 presents the results associated with the growth dynamics of each network, and Section 4.3 provides the results as they pertain to conceptual robustness and preventing emergent design failures.

Case 1 The *aggregated average* case. An average value is determined for each variable (L, B, T, and C_B). The average variable values are multiplied together to determine an average volume. In this case, the average variable values are determined for *all* ships in a single calculation. This approach is shown in Table 4.2. To illustrate the importance of understanding the relations between data entities, Case 1 contains two methods of tracking data: *labeled* and *unlabeled*. The labeled case assumes the indexes of each ship are known and tracked in the knowledge structure (i.e. all variables corresponding to the same ship have the same indexes), and the unlabeled case assumes the relations between data and ships are unknown (the data entities and ship indexes are uncorrelated). This distinction does not affect the dynamics of the knowledge structure growth, but has significant impacts on the likelihood of emergent design failures. These effects are outlined in more detail in Section 4.3.

Case 2 The *baseline* case. A volume is calculated for each ship, and the resultant volumes are utilized to determine an average volume. This approach is presented in Table 4.3.

Case 3 The *limited information* case. This case uses the same approach as Case 2 (the baseline case) in that each ship has an associated volume value. However, this case omits the principal dimension values, and only includes the volume data. This represents a fundamentally different data source; it is included to show that although the volume values (and calculated average) are identical to those of Case 2, this data set contains less information, and leads to a different fundamentally different knowledge structure. This approach is presented in Table 4.4.

Case 4 The *moving average* case. This case uses the same approach as Case 1, with the only difference being that a moving average is calculated for each row in the data set (using all values above and including that row). In this case, the final result of the moving average is identical to the resulting average calculation in Case 1 and Case 3, although the dynamics of building the knowledge structure are significantly different. This approach is presented in Table 4.5.

This case study is an exercise to demonstrate that although the same set of data is utilized across a number of calculation approaches, the predictability and robustness of the developed knowledge structures differ significantly. These differences are a product of the ability for each approach to account for hidden relations in the data, through the use of proper sources of information and structuring of the knowledge entities. The values in Table 4.1 are just *data*, however these data correspond to products created by careful engineering analyses and the knowledge structures used to create them. The knowledge structures used to create these data are not apparent in analyzing the data alone, but nonetheless have created hidden interdependencies between variables for each ship. Statistically, if the data were independent, the covariance between them would be zero, and the final result of calculating an average volume would be identical if averages were aggregated across the rows or along the columns in the table. However, as the data correspond to physical vessels and their associated knowledge structures, they not independent, and the robustness of determining a final average value is heavily related to the ability to account for the hidden data interdependencies.

4.1 Local Information and Knowledge Structures

4.1.1 Case 1 - Aggregated Averages

The aggregated average case conducts operations as shown in Table 4.2, in which averages are determined for each variable (L, B, T, and C_B) and multiplied together to determine an average volume value. This section outlines the information structures utilized in this operation and the resultant knowledge structure for this approach.

The first step in representing this approach in the K-I Framework is representing the information structures in the information layer utilized in this approach. This is shown in Figure 4.1. In Case 1, five distinct information sources define the information sources used to build this approach: four define the relation between the individual variable data points relate to the average variable value (for L, B, T and C_B), and one defines how the average values are combined to yield a resultant average volume value. Note, however, that the information layer is also comprised of all possible information

Table 4.2: The calculation procedure for Case 1.

Oceanograpic Ship	L	B	T	C_B	**Volume**
AGOR 16	69.75	12.92	4.45	0.538	
Atlantis II	59.52	13.53	4.92	0.537	
Chas. Darwin	62.50	14.40	5.11	0.539	
Endeavor	50.30	10.31	5.34	0.500	
Littlehales(T-AGS51)	58.96	12.82	4.02	0.551	
Maury (T-AGS39)	141.73	20.54	8.49	0.564	
Melville (AGOR14)	69.19	14.42	4.65	0.518	
Pathfinder (T-AGS60)	93.09	16.33	5.35	0.460	
Protea	73.75	15.36	4.70	0.551	
Researcher (OSS-03)	78.86	15.77	4.93	0.457	
Robert Conrad (AGOR3)	58.96	11.12	4.28	0.427	
Silas Bent (AGS26)	80.82	14.69	4.59	0.472	
Stalwart (T-AGOS1)	63.24	12.16	4.21	0.560	
Thomas Thompson	77.66	14.93	5.41	0.462	
Average	74.17	14.24	5.03	0.510	**2708.1**

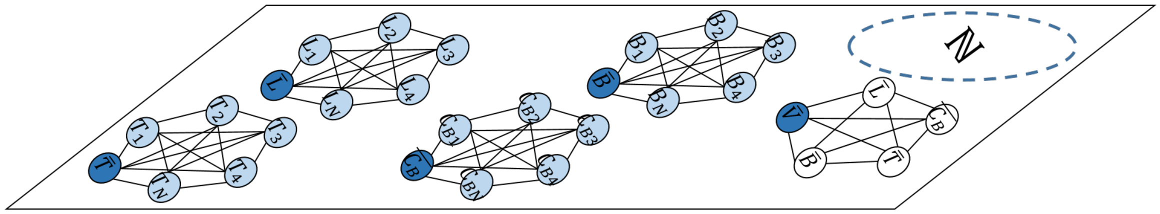


Figure 4.1: Information Layer associated with Case 1

structures (ℕ) - the structures displayed are only those which are utilized in this case, and thus are only a small subset of ℕ. Mathematically, these information relations used in Case 1 are described in equations (4.1) - (4.5):

$$\bar{C}_B = \frac{1}{14}\sum_{i=1}^{14} C_{Bi} \tag{4.1}$$

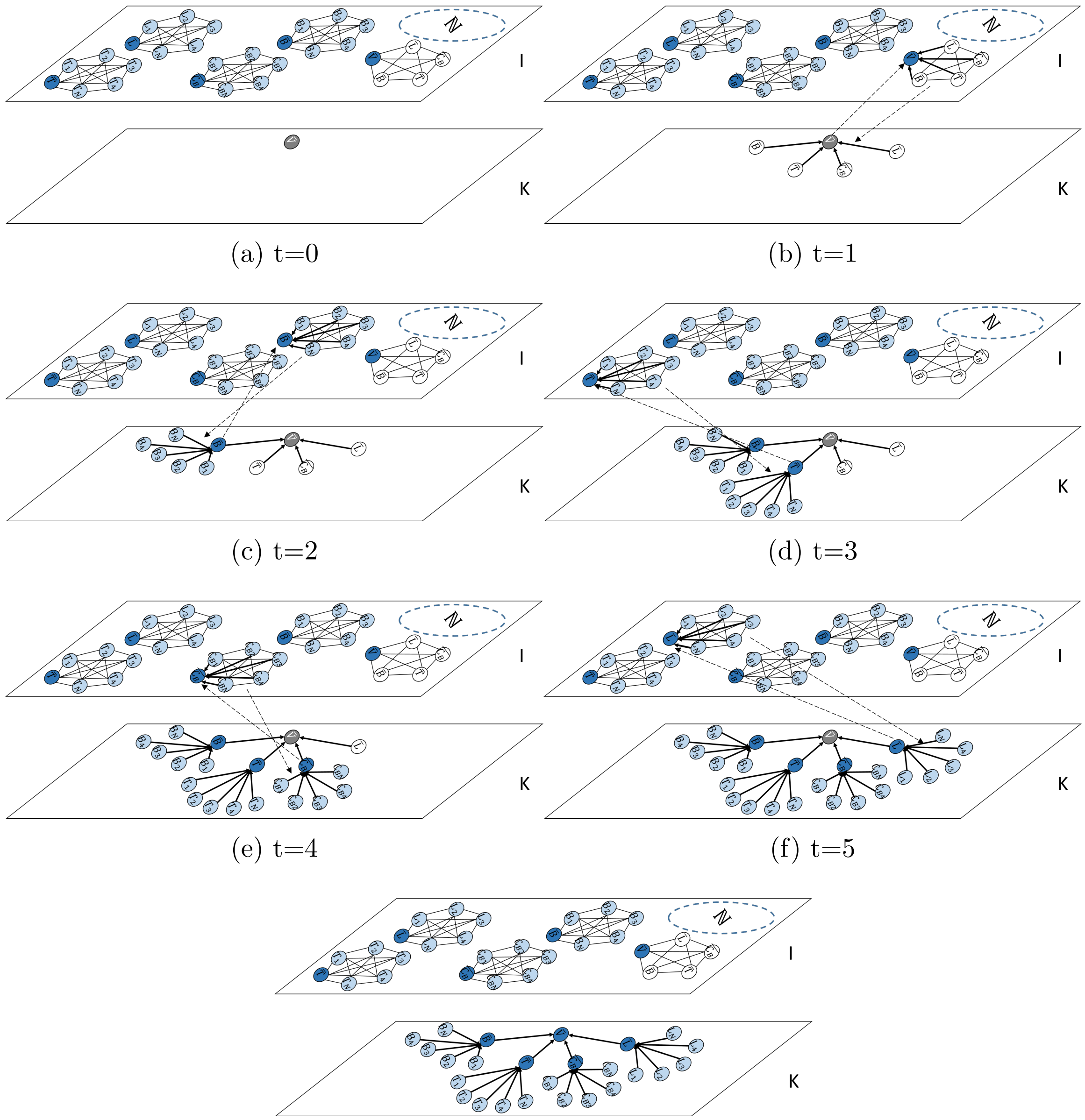
$$\bar{L} = \frac{1}{14}\sum_{i=1}^{14} L_i \tag{4.2}$$

$$\bar{B} = \frac{1}{14}\sum_{i=1}^{14} B_i \tag{4.3}$$

$$\bar{T} = \frac{1}{14}\sum_{i=1}^{14} T_i \tag{4.4}$$

$$\bar{V} = \bar{C}_B\bar{L}\bar{B}\bar{T} \tag{4.5}$$

In Figure 4.1, the colors of the nodes represent the data statuses of each node (data point) in the information structure. Uncolored nodes represent variables which do not have any data supporting them, the light shaded nodes represent variables which contain data from the database, and dark colored nodes are those variables for which there is sufficient data for their values to be calculated. These information structures are used to create a unique knowledge structure for this case. The process of creating this knowledge structure is shown in Figure 4.2.



(a) t=0

(b) t=1

(c) t=2

(d) t=3

(e) t=4

(f) t=5

(g) Final Structure

Figure 4.2: Generation of Knowledge layer from information structures in Information layer over time for Case 1 (aggregated averages) to calculate an average volume value. The resultant knowledge structure is developed through a sequence of hypotheses (directed edges from knowledge layer to information layer) and actions (directed edges from information layer to knowledge layer).

4.1.2 Case 2 - Baseline

The baseline case utilizes the reverse order of operations to Case 1 to determine the value of average volume. In this approach a volume is calculated for each oceanographic ship, and the resultant volumes are combined to determine an average volume. The calculation and resultant data are presented in Table 4.3.

Similar to Case 1, the first step in applying the K-I Framework to this approach is representing the relations between data entities in the information layer. The mathematical equations representing the information structures and associated information network layer are displayed in equations (4.6) - (4.7) and Figure 4.3, respectively.

$$V_i = C_{Bi} L_i B_i T_i \qquad i = 1, ..., 14 \tag{4.6}$$

$$\bar{V} = \frac{1}{14}\sum_{i=1}^{14} V_i \tag{4.7}$$

Table 4.3: The calculation procedure for Case 2.

Oceanograpic Ship	L	B	T	C_B	Volume
AGOR 16	69.75	12.92	4.45	0.538	2157.5
Atlantis II	59.52	13.53	4.92	0.537	2127.6
Chas. Darwin	62.50	14.40	5.11	0.539	2478.9
Endeavor	50.30	10.31	5.34	0.500	1384.6
Littlehales(T-AGS51)	58.96	12.82	4.02	0.551	1674.3
Maury (T-AGS39)	141.73	20.54	8.49	0.564	13939.6
Melville (AGOR14)	69.19	14.42	4.65	0.518	2403.2
Pathfinder (T-AGS60)	93.09	16.33	5.35	0.460	3741.1
Protea	73.75	15.36	4.70	0.551	2933.6
Researcher (OSS-03)	78.86	15.77	4.93	0.457	2801.9
Robert Conrad (AGOR3)	58.96	11.12	4.28	0.427	1198.2
Silas Bent (AGS26)	80.82	14.69	4.59	0.472	2572.1
Stalwart (T-AGOS1)	63.24	12.16	4.21	0.560	1813.0
Thomas Thompson	77.66	14.93	5.41	0.462	2898.0
Average					**3151.7**

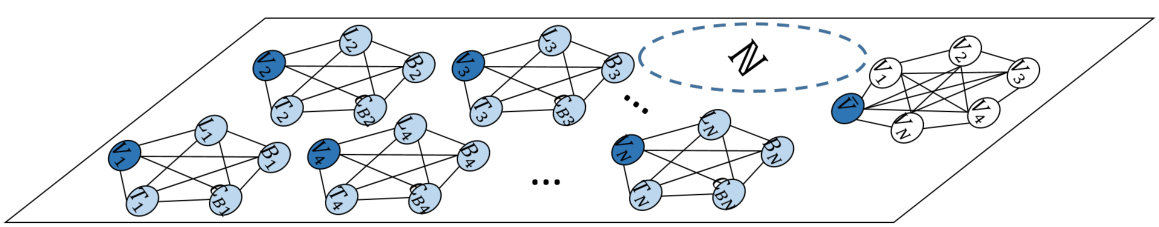
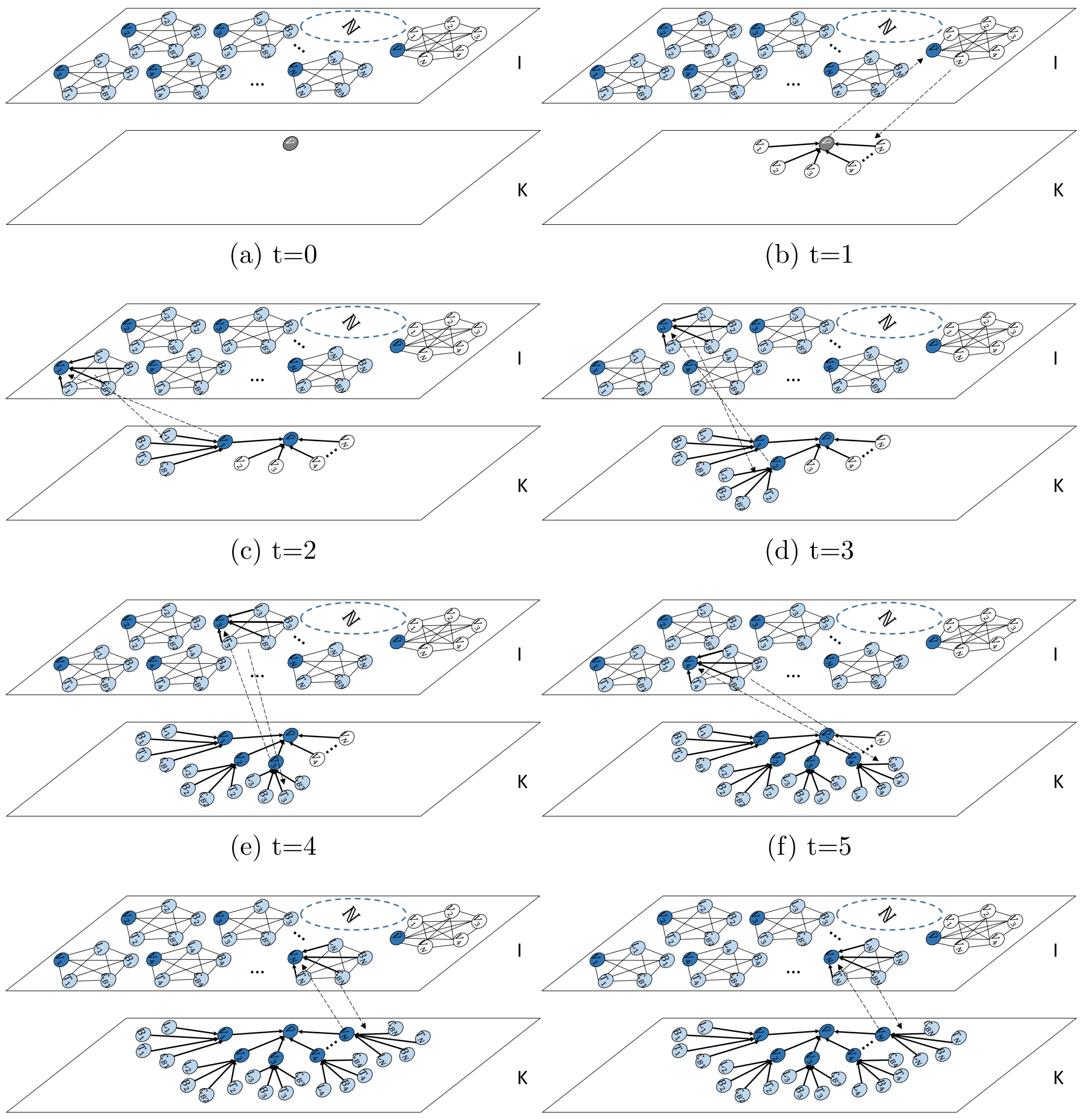


Figure 4.3: Information Layer associated with Case 2



(a) t=0 (b) t=1

(c) t=2 (d) t=3

(e) t=4 (f) t=5

(g) t=N (h) Final Structure

Figure 4.4: Generation of Knowledge layer from information structures in Information layer over time for Case 2 (baseline) to calculate an average volume value. The resultant knowledge structure is developed through a sequence of hypotheses (directed edges from knowledge layer to information layer) and actions (directed edges from information layer to knowledge layer).

4.1.3 Case 3 - Limited Information

The 'limited information' case is similar to Case 2 (the baseline case) in that each ship has an associated volume value; however, this case omits the supporting variable values. In this sense, the volume values are now data points rather than results of intermediate calculations. As the volume values are now data points, the ways in which they are calculated are unknown in this example. This is akin to communication between design teams: only the values are communicated, without the supporting information. In this case, the volume values associated with each vessel are identical to those of Case 2, yet lead to not only a more limited knowledge structure, but also more limited dynamics. This approach is presented in Table 4.4.

Table 4.4: The calculation procedure for Case 3.

Oceanograpic Ship	**Volume**
AGOR 16	2157.5
Atlantis II	2127.6
Chas. Darwin	2478.9
Endeavor	1384.6
Littlehales(T-AGS51)	1674.3
Maury (T-AGS39)	13939.6
Melville (AGOR14)	2403.2
Pathfinder (T-AGS60)	3741.1
Protea	2933.6
Researcher (OSS-03)	2801.9
Robert Conrad (AGOR3)	1198.2
Silas Bent (AGS26)	2572.1
Stalwart (T-AGOS1)	1813.0
Thomas Thompson	2898.0
Average	**3151.7**

The relations of data in the limited information case (the information structure) are only comprised of equation (4.7), and omits the intermediate calculation represented by (4.6). The resultant information layer is displayed in Figure 4.5.

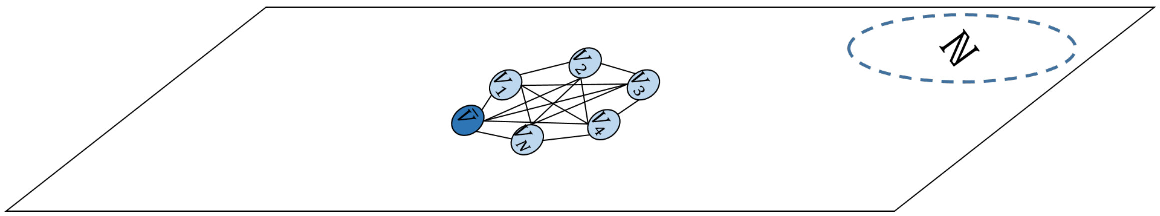
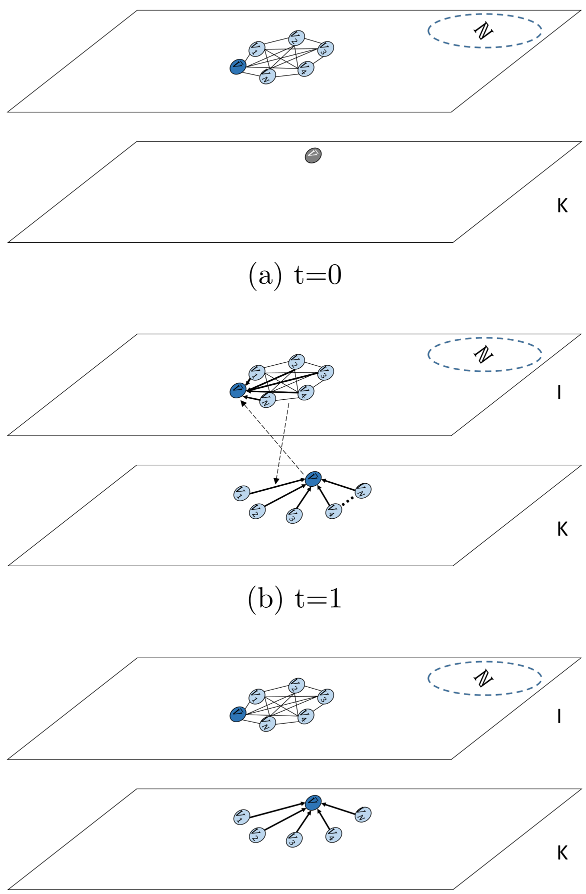


Figure 4.5: Information Layer associated with Case 3



(a) t=0

(b) t=1

(c) t=2

Figure 4.6: Generation of Knowledge layer from information structures in Information layer over time for Case 3 (limited information) to calculate an average volume value. The resultant knowledge structure is developed through a sequence of hypotheses (directed edges from knowledge layer to information layer) and actions (directed edges from information layer to knowledge layer).

4.1.4 Case 4 - Moving Average

The moving average case uses the same approach as Case 1, the 'aggregated average' case, with the only difference being the way the network grows. Case 1 sequentially calculates the intermediate variable averages considering all data points. This process requires five timesteps to fully develop the knowledge structure (step 1 importing equation (4.5), and steps 2 - 5 importing equations (4.1) - (4.4)). Case 4 considers more steps to build the knowledge structure by sequentially adding each row of ship data to the network, and calculating the intermediate averages at each timestep, for a total of 15 steps (first importing (4.5), then calculating 14 averages as each ship data is added). In this case, the final result of the moving average is identical to the result of Case 1, though the dynamics of this case's approach uncovers path dependencies in the calculation. The results are presented in Table 4.5.

The equations represented in the information layer for this approach are shown in equations (4.8) - (4.12). The resulting information layer from this approach is identical to that of Case 1, shown in Figure 4.1. The information layer is the same between Case 1 and Case 4 because the same fundamental data relations are preserved, although the number of data points considered changes. Both information layers contain 5 information structures used for the calculation (in addition to $\mathbb{N}$), and although the

Table 4.5: The calculation procedure for Case 4.

Oceanograpic Ship	L	B	T	C_B	**Volume**
AGOR 16	69.75	12.92	4.45	0.538	2157.5
Atlantis II	59.52	13.53	4.92	0.537	2152.5
Chas. Darwin	62.50	14.40	5.11	0.539	2260.3
Endeavor	50.30	10.31	5.34	0.500	2026.9
Littlehales(T-AGS51)	58.96	12.82	4.02	0.551	1957.8
Maury (T-AGS39)	141.73	20.54	8.49	0.564	3014.4
Melville (AGOR14)	69.19	14.42	4.65	0.518	2923.2
Pathfinder (T-AGS60)	93.09	16.33	5.35	0.460	3032.2
Protea	73.75	15.36	4.70	0.551	3024.2
Researcher (OSS-03)	78.86	15.77	4.93	0.457	3005.6
Robert Conrad (AGOR3)	58.96	11.12	4.28	0.427	2787.8
Silas Bent (AGS26)	80.82	14.69	4.59	0.472	2771.9
Stalwart (T-AGOS1)	63.24	12.16	4.21	0.560	2691.9
Thomas Thompson	77.66	14.93	5.41	0.462	2708.1
Final Average					**2708.1**

number of nodes in the information structures change, in both cases each information structure is a fully connected graph. Though the information structures are identical, the dynamics used to grow the knowledge network is different, and is shown in Figure 4.7.

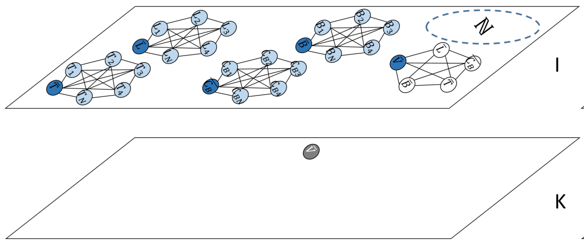
$$\bar{V}_j = \bar{C}_{Bj}\bar{L}_j\bar{B}_j\bar{T}_j \qquad j = 1, ..., 14 \tag{4.8}$$

$$\bar{C}_{Bj} = \frac{1}{j}\sum_{i=1}^{j} C_{Bi} \qquad j = 1, ..., 14 \tag{4.9}$$

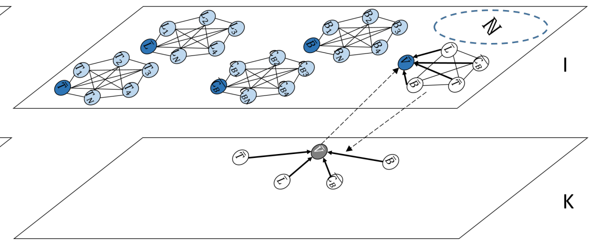
$$\bar{L}_j = \frac{1}{j}\sum_{i=1}^{j} L_i \qquad j = 1, ..., 14 \tag{4.10}$$

$$\bar{B}_j = \frac{1}{j}\sum_{i=1}^{j} B_i \qquad j = 1, ..., 14 \tag{4.11}$$

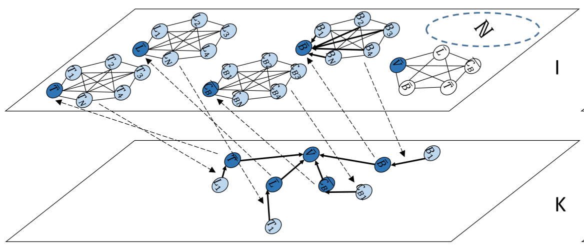
$$\bar{T}_j = \frac{1}{j}\sum_{i=1}^{j} T_i \qquad j = 1, ..., 14 \tag{4.12}$$



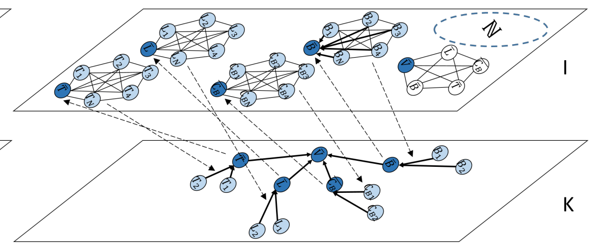
(a) t=0



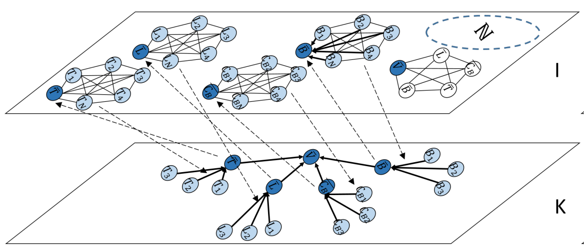
(b) t=1



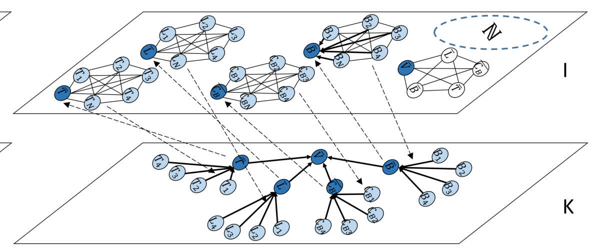
(c) t=2



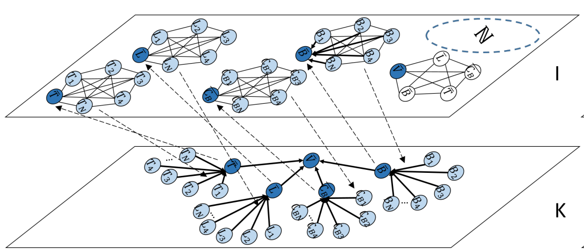
(d) t=3



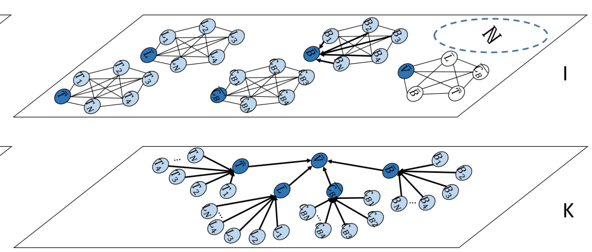
(e) t=4



(f) t=5



(g) t=N

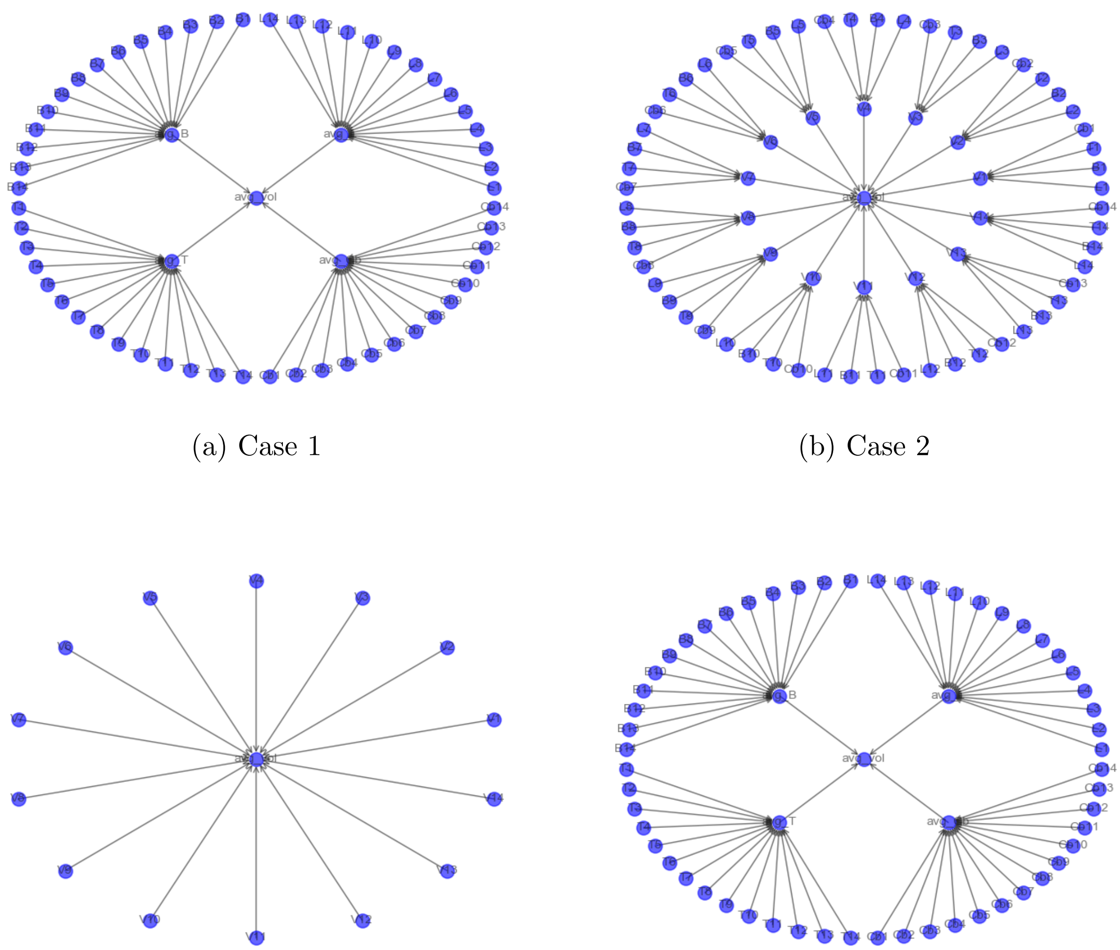


(h) Final Structure

Figure 4.7: Generation of Knowledge layer from information structures in Information layer over time for Case 4 to calculate an average volume value. The resultant knowledge structure is developed through a sequence of hypotheses (directed edges from knowledge layer to information layer) and actions (directed edges from information layer to knowledge layer).

4.1.5 Resultant Case Knowledge Structures

Figure 4.8 illustrates the resultant knowledge structures for Cases 1-4. In each network, the node at the center of the image is the target node (average volume), and the nodes in the outer-most ring represent the data entities in the associated data set. For Figures 4.8a, 4.8b, and 4.8d, the nodes between the target node and data entities represent the intermediate variables in each calculation approach. Note that the final knowledge structures of Case 1 and Case 4 are identical - although the dynamics used to create each are different.



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 4.8: Resultant knowledge structures for Cases 1-4.

4.2 Local Knowledge Structure Growth Results

4.2.1 Topological Entropy

The topological entropy time series for Cases 1-4 are presented in Figure 4.9. These curves represent the topological entropy of the knowledge networks' *structures* as they grow over time. In this figure, the x-axis represents the timestep, which corresponds to the growth step in the process defined in Figure 3.3 and the dynamics described in Section 4.1, and the y-axis is Topological Entropy (Section 2.3.1).

The curves shown in Figure 4.9 reveal a number of insights into the dynamics of each case. Each curve begins at the origin, and ends at the final topological entropy value of the resultant network. At time $t = 0$ the only node in the knowledge layer is the target node, and thus topological entropy is zero. The final value is dependent on the final structure of the knowledge network. Note that the different cases require a different number of timesteps to yield the final network structure with Case 3 requiring only one timestep, Case 1 requiring 5 timesteps, and Cases 2 and 4 each requiring 15 timesteps.

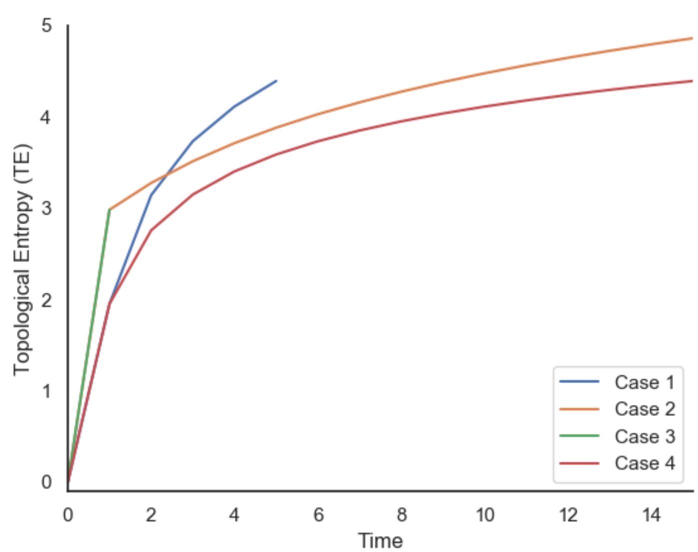


Figure 4.9: Topological Entropy Time Series Comparison, Cases 1-4

Additionally, each curve increases drastically in the first timestep (time $t = 0$), and is concave down in following steps. These two phases of knowledge structure growth define the *knowledge development* and *knowledge refinement*. The large initial entropy growth in the knowledge development phase arises as a result of the initial hypothesis and action utilized in selecting and integrating an information source into the knowledge structure (Section 3.1.2). The initial structuring of the knowledge layer is a critical factor in the growth of topological entropy, by defining not only the initial size of the network (structuring what subsequent knowledge is required), but also creating path dependencies in future knowledge-seeking activities. Note that the knowledge development phases are identical for Cases 1 and 4, and for Cases 2 and 3, as they both utilize the same initial information structure to implement their approach. The initial growth in entropy of Cases 2 and 3 is larger than that of Cases 1 and 4 due to the different sizes of the initial structures (the initial network of Cases 2 and 3 is comprised of 15 nodes, while that of Cases 1 and 4 is only comprised of 5 nodes). Given the different approaches, the knowledge refinement stages differ between cases, which are a result of the different network growth dynamics.

The knowledge refinement stage integrates additional knowledge into the initial knowledge structure (time $t > 1$). The concavity of the knowledge refinement stage suggests that early refinement of knowledge leads to higher entropy growth than at later stages, which illustrates that the marginal increase of knowledge contained does not increase linearly over time, and that the entropy of the knowledge structure is not directly correlated to the number of data points. Case 1 has the largest increase in topological entropy during knowledge refinement, with 14 additional nodes added to the network at each timestep. This means the network grows most rapidly during knowledge refinement, which differs from that of Case 4, in which each timestep integrates just 4 additional nodes into the network. While this takes a longer time to yield the final network (and the final topological entropy values are the same given that the resultant structures are the same), Case 4 better illustrates the decreasing marginal increase in knowledge over time. Case 2 has the most gradual increase in knowledge refinement over time, and yields the highest knowledge in the resulting structure. Note that Case 3 does not contain a knowledge refinement stage as the information source contains all data points when integrated into the knowledge network, meaning no subsequent knowledge needs to be determined to yield these values. While this may seem advantageous, note that the entropy of the resulting network is the smallest, meaning the structure of the network contains the least amount of encapsulated knowledge.

The topological entropy curves in Figure 4.9 provide a macro-perspective of topological entropy growth over time of the local knowledge layers. More detailed insights can be gained for each case by viewing the entropic contribution of each node (knowledge entity) in the network to the total topological entropy. These results are shown in Figures 4.10 - 4.13. By observing how each node contributes to the total topological entropy over time, more insight can be gained as to how the growth of individual knowledge entities lead to the growth of the entire network. The way in which the information relating to the *target node*, *intermediate variables*, and *data points* contribute to the total entropy over time is of particular interest.

For all cases, the growth in entropy of the target node is characterized by the initial structure of the network ($t = 1$), and remains relatively constant during the knowledge refinement stage. For Cases 1, 2, and 4 (Figures 4.10, 4.11, and 4.13, respectively) the final contribution of the intermediate variables to the total topological entropy is less than in the initial structure, while the contribution of the data points increases over time. This highlights a defining characteristic of the refinement stage, where the supporting knowledge from the data points is used to define the intermediate variables. This characterizes the increasing importance of the data points as more nodes are integrated into the structure.

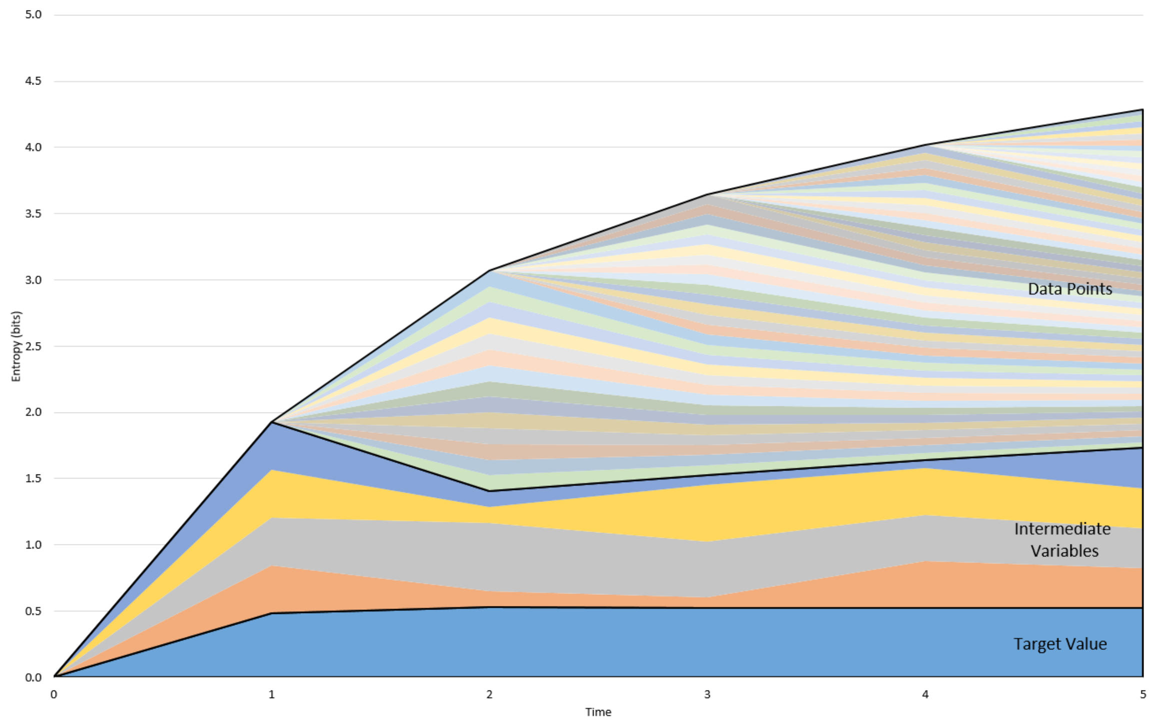


Figure 4.10: Case 1 Topological Entropy Time Series Composition

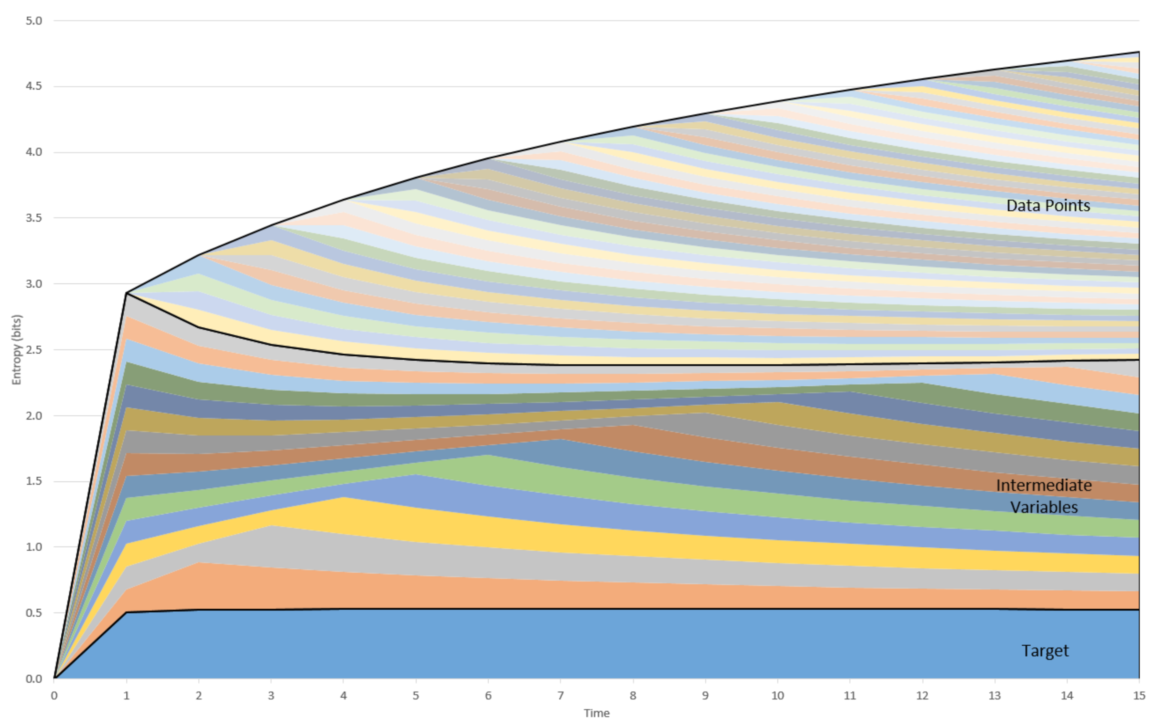


Figure 4.11: Case 2 Topological Entropy Time Series Composition

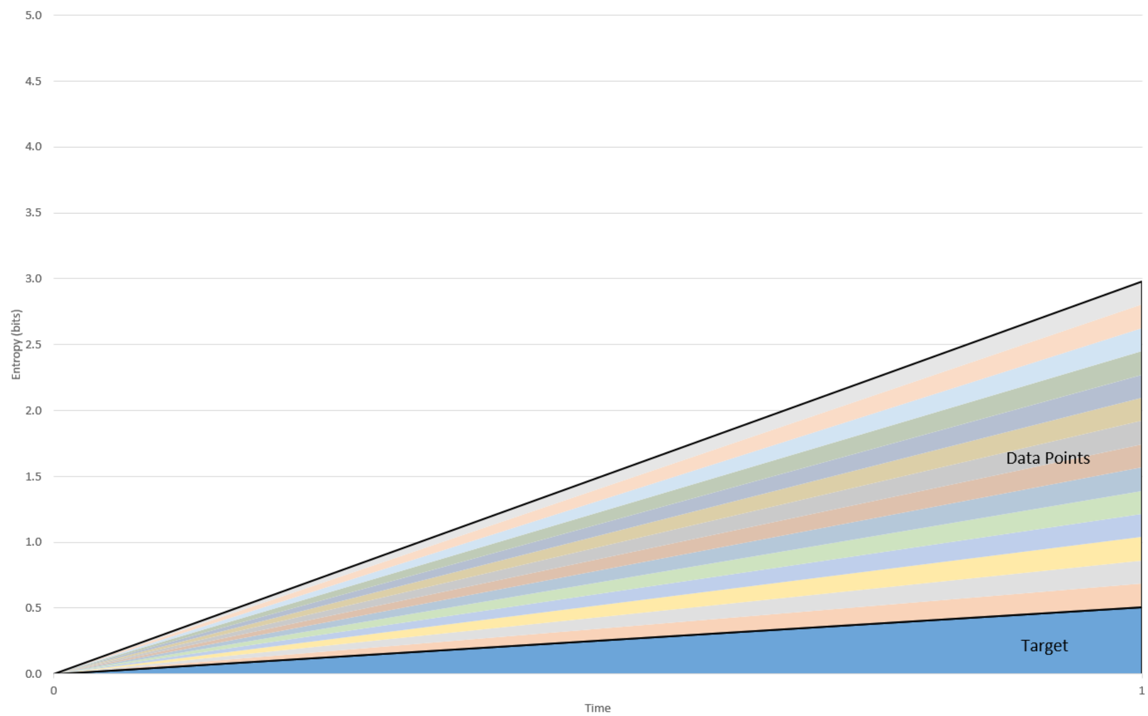


Figure 4.12: Case 3 Topological Entropy Time Series Composition

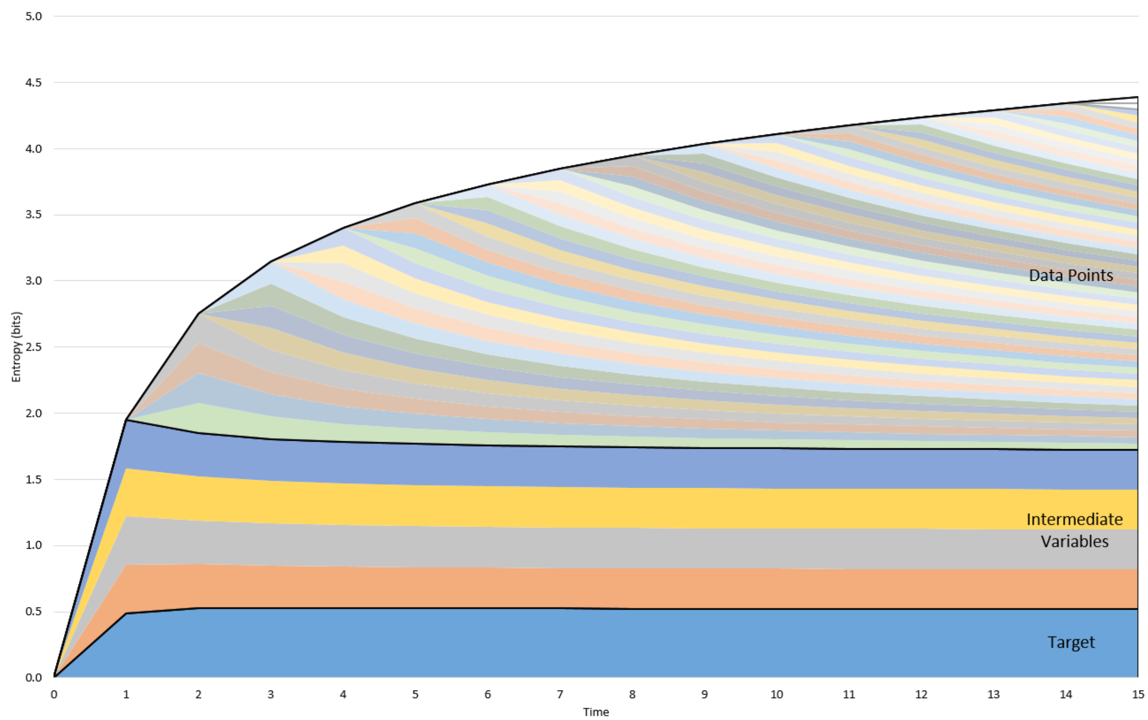


Figure 4.13: Case 4 Topological Entropy Time Series Composition

4.2.2 Data Status Entropy

The data status entropy results for Cases 1-4 are displayed in Figure 4.14. These curves represent the entropy of the known portion of the nodes in the network based on their respective data statuses. This provides insight to the growth of the 'calculability' of the network over time. The x-axis represents timesteps, while the y-axis represents the data status entropy (Section 5.4.2).

Each curve represents the transition from all data values being uncalculated to being fully defined. For all curves, the network begins with a maximal entropy value of 1, where all nodes are unknown, and ends with an entropy value of 0, when all nodes are known. This is due to the target node initially being undetermined. Note that the length of each curve corresponds to the number of timesteps to create the final network.

Cases 3 and 4 illustrate an immediate transition from maximum to minimum DSE, indicating that the network transitions from being completely incalculable to being completely calculable. In Case 3, this is because all nodes in the network have a data

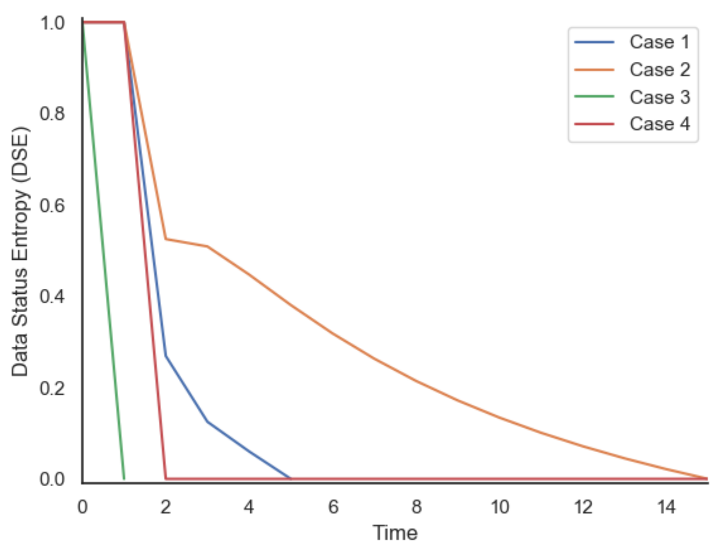


Figure 4.14: Data Status Entropy Time Series Comparison, Cases 1-4

status of 0 at time $t = 0$ (just the target node) and all nodes have a data status of 1 at time $t = 1$, when all data is imported. Similarly, all nodes have a data status of 0 at timesteps $t = 0$ and $t = 1$ in Case 4. At time $t > 1$ the addition of each data point node to each of the intermediate variables means that all subsequent data statuses are at 1 - meaning the network remains fully calculable throughout its growth.

Cases 1 and 2 exhibit more gradual reductions in DSE. For Case 1, at time $t \leq 1$, all of the nodes in the network have data statuses of 0. At time $t > 1$ the sequential introduction of supporting nodes to each of the intermediate variables means a portion of the approach is calculable, while a portion still contains data statuses equal to zero, up until all nodes have data statuses of 1 at time $t = 5$. Case 2 follows the same reasoning as Case 1 for $t \leq 1$ (all data statuses are at 0); however, the sequential addition of nodes supporting the intermediate volume variables results in a more gradual change in the network's calculability over time until all nodes have data statuses of 1. Note that the initial reduction in DSE is larger for both cases. This is due to the small initial size of the network - when a single node being calculable is more impactful than later in the process.

The shapes of the DSE curves reveal a number of insights into the dynamics of rework. In the presented cases, all DSE curves are monotonically decreasing which indicates that the structures are steadily becoming more calculable over time. Any observed increase in DSE would flag the beginning of design churn, as it would indicate the addition of incalculable nodes to the structure. This suggests that the process of calculating an answer has yielded additional unknowns. Design churn and other emergent design failures are further explored in Section 4.3.

4.3 Analyzing Local Knowledge Structure Robustness

While Section 4.2 outlined the insights gained by studying the dynamics *of* knowledge structure growth, this section presents insights gained from studying dynamics *on* the knowledge structures as they grow. The latter considers the robustness of knowledge structures when exposed to changes in the data landscape. Changes in the data landscape could be a consequence of future information, changes to requirements, or as a result of rework necessitated by creating a converged design. To frame this, consider the following scenario:

> There is a 'volume' design team, responsible for determining a hull volume estimate for a novel oceanographic vessel design. Initially, the team decided to use a database of previous vessels with similar characteristics to that of their design (Table 4.1) to determine a preliminary volume estimate. Given this goal, the team has conducted volume estimates using the approaches outlined in cases 1-4 (Section IV). Later in the design process, the results of another design team requires the volume design team to revise their estimate by removing the largest ship from their calculation to determine a new average volume.

The scenario presented above leads to a number of questions pertaining the to robustness of knowledge structures:

1. How easily can the outlier be identified while the knowledge structure is growing?

2. Once the outlier is identified, how easy is it to remove the outlier data from the knowledge structure?

3. If the estimates needed to be repeated to remove the second largest ship rather than just the largest ship, how easy would this be?

The questions presented in the above scenario are used to gain insights into the benefits and limitations of each of the cases. Specifically, Target Value Entropy (Section 2.3.5) is used to answer the first question by studying the path dependencies in the calculated target node values in Section 4.3.1. The time required to remove the outlier is used to address the second question and study excessive rework and design churn in Section 4.3.2. Potential integration failures are presented in Section 4.3.3, and are used to answer the third question. These insights are used to draw conclusions about which of the presented cases is most robust, and are summarized in Section 4.4.

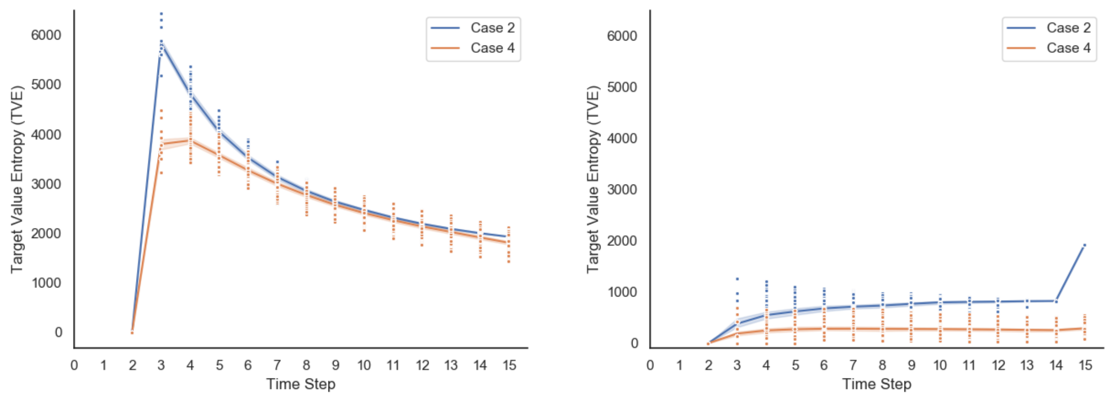
4.3.1 Predictability - Target Value Entropy

To answer the first question posed in Section 4.3, a simulation approach was utilized to uncover the likelihood of identifying the moment when the largest ship was introduced to the analysis. Ideally, a knowledge structure would be able to immediately identify the introduction of erroneous data into the calculation, and provide a flag to the design team independent of the time at which it was added. Path dependencies and the sequence of operations play a critical role in the ability to identify these outliers, thus the dynamics of both the knowledge network's structure and calculability are crucial in being able to identify potential data issues. Identifying these outliers can be determined by applying the Target Value Entropy metric (Section 2.3.5) to the evolution of the solution of the target node value over time.

Studying the evolution of the target node's value over time presents issues with the approaches of Cases 1 and 3. In Cases 1 and 3, the value in the target node can only be calculated at the final timestep ($t = 5$ and $t = 1$, respectively), meaning there is only ever one value associated with the target value. In Case 1, the final value is only calculable once all intermediate average values have been determined, leading to a singular final result. For Case 3, the volume data being imported all at once means there is no transparency as to which individual values lead to the calculated result. In both cases, there is no evolution of the target node value over time. It is only possible to identify the final result, without understanding the impact that each set of data played leading to the result. Herein lies the importance of the knowledge refinement phase of the network growth. The knowledge refinement phase should highlight the effects of adding additional input data to the values determined in the target node over time, to increase the transparency of the role of each data element to the calculated value. This is only possible using Cases 2 and 4, which are used to reveal the differences across the two approaches. It should be noted that the respective structures of Case 3 and Case 2 are similar, with the difference being that Case 2 contains the knowledge refinement phase. Also notable, Case 1 and Case 4 are similar in that their final structures are identical, though the growth dynamics of Case 4 enable the target value to be determined at each timestep in the knowledge refinement phase.

To examine the ability for Case 2 and 4's respective knowledge structures to identify the outlier data over time, 2200 trials were performed in which the order ship data added to each tested knowledge structure was randomized. These randomized trials were performed to account for different sequences of data being applied to the knowledge structure, leading to different path dependencies in the calculated values. The randomized trials were filtered to determine the instances when the outlier was added at each timestep from $t = 2 - 15$. Note that $t = 1$ marks the end of the knowledge development phase, and as such, the target node value is yet to be calculable (hence it is omitted). The results of adding the outlier at the first and last timesteps are displayed in Figure 4.15, and the intermediary timestep results are presented in 4.16. The points on the plot at each timestep represent the resultant TVE value for each of the filtered trials; the curves included display the average results across filtered trial, with shaded error bounds corresponding to a 95% confidence interval.

The results in Figure 4.15 illustrate a number of critical insights into the predictability of the results from Cases 2 and 4. In each of the plots, the spread of points



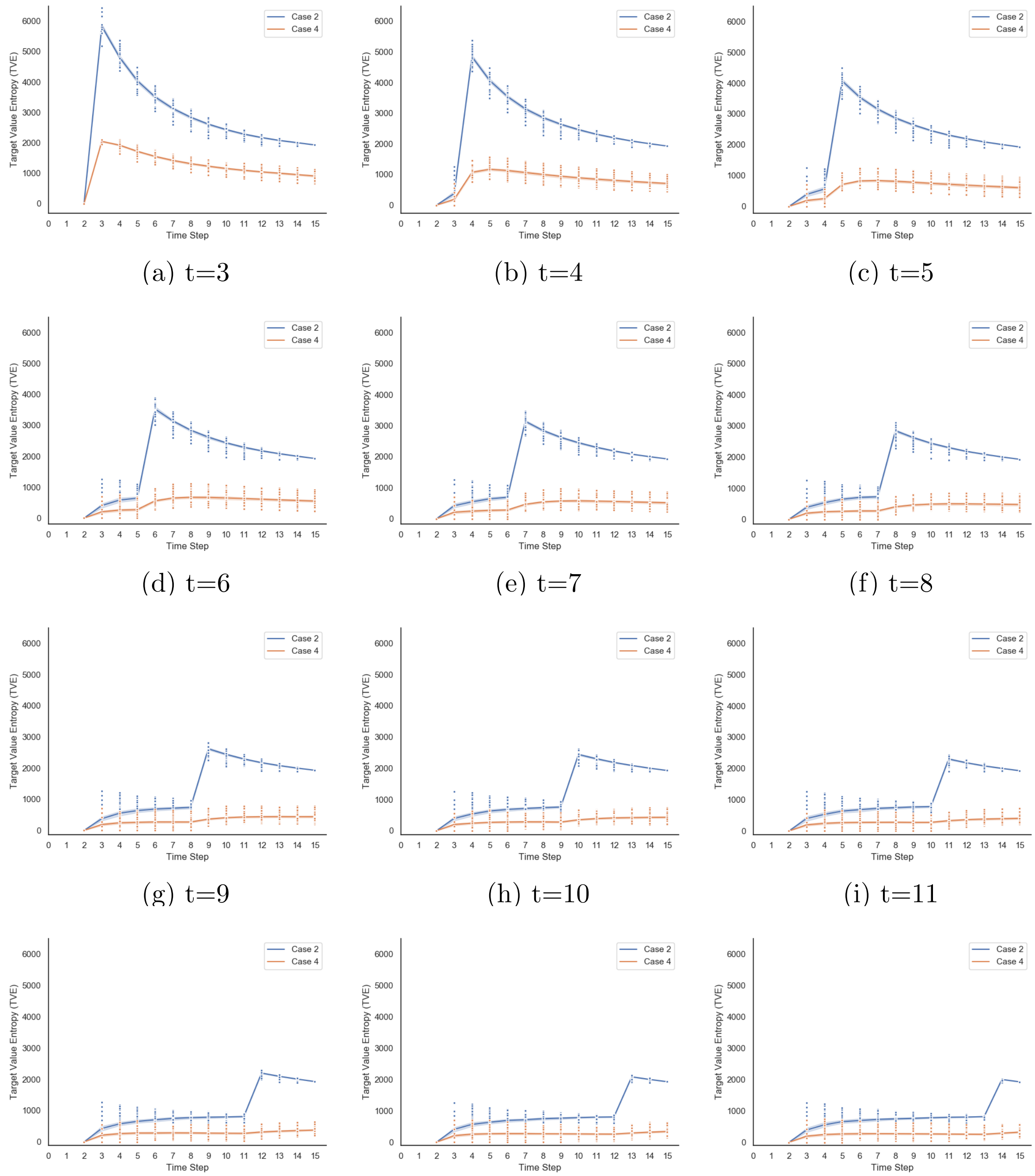
(a) t=2

(b) t=15

Figure 4.15: Target Value Entropy Results with outlier data at first and last timestep ($t = 2$ and $t = 15$).

at each timestep illustrates the range of TVE values observed using the samples of the filtered randomized trials. For all curves, a spike in TVE corresponds to the identification of the outlier's data being added to the knowledge structure, by quantifying the increased uncertainty of adding data which are significantly different from previously added data. As more of the data is added after the outlier the TVE decreases, which illustrates a decrease in uncertainty. Gradual increases in TVE before the outlier is added suggest that data added to the structure are similar (in relative terms) to data which have already been added. In all cases, the TVE values at time $t = 2$ are zero, at which point there is only a single calculated value, hence there is no uncertainty in the calculated value (the single outcome is certain).

The results in Figure 4.15 suggest there are significant difference in each case's ability to recognize the addition of the outlier over time. The plots in Figure 4.16 display the TVE results for the intermediary timesteps ($t = 3 - 14$), and confirm this hypothesis. When the outlier is added early (say, $t \leq 5$) the increase in uncertainty is apparent with a spike in TVE across both cases. In a special case, when the outlier is the first set of data added to the structure (Figure 4.15a) the spike is only apparent when the subsequent data is added. This is intuitive, as there is no way



(a) t=3 (b) t=4 (c) t=5

(d) t=6 (e) t=7 (f) t=8

(g) t=9 (h) t=10 (i) t=11

(j) t=12 (k) t=13 (l) t=14

Figure 4.16: Target Value Entropy Results with outlier data added at each timestep ($t = 2 - 15$).

of knowing whether the first data added is the outlier without subsequent data to compare it to. The additional data provided by the knowledge refinement process provide a relative measure of its value, and provide context as to when the outlier was added. The differences between Case 2 and 4's respective knowledge structures becomes apparent when the outlier is added in later timesteps (say, $t > 5$). For Case 2, the spike in TVE remains independent of when the outlier is added, and it becomes increasingly difficult for Case 4's knowledge structure to distinguish the erroneous point. This is a result of the erroneous data being 'washed out' by the intermediate average calculations in Case 4, which yields target node values which are more similar to previous calculations. This differs from Case 2, in which the sequence of values used in determining the target node value are the volume values themselves. This difference becomes more obvious in observing the resulting distributions of calculated target values between the two cases, which is shown in Figure 4.17.

The propensity for the intermediate average calculations to 'wash out' the values used to calculate the target node in Case 4 reveals an emergent path dependence based on the sequence in which the values are added to the knowledge structure. The same is not true for Case 2, which is only based on the values added, but not the order in which they were added. The path dependencies of Case 4 also mean the final TVE ($t = 15$) changes as a result of the sequence in which data is added. This is different from Case 2, which has the same final TVE value independent of when the

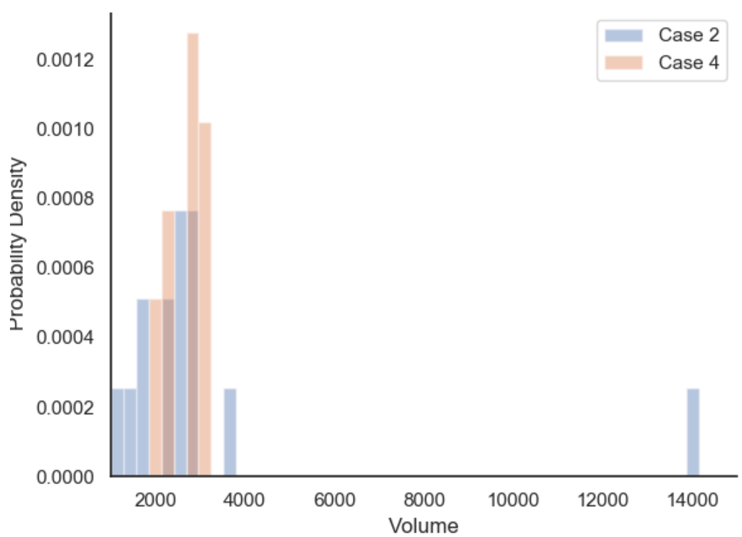


Figure 4.17: Distribution of volume values used in target value calculation, Cases 2 and 4

outlier is added. This suggests that Case 2 provides a knowledge structure which is more conducive to determining the instant the erroneous value is added.

4.3.2 Excessive Rework

To answer the second question posed in the hypothetical scenario outlined in Section 4.3, an analysis was conducted to determine the time required to remove the outlier from the respective knowledge structures. The time required to remove the outlier was determined by measuring the CPU time required for an algorithm to remove all associated outlier data for each respective case. Using CPU time to conduct rework provides a somewhat idealized view of the time required to conduct rework, but provides a good relative comparison between cases. As was outlined in the case descriptions in Section 4.3, two instances of Case 1 are presented to highlight the importance of data traceability in preventing excessive rework activities: *labeled* and *unlabeled*. The labeled case assumes that each principal dimension variable is indexed by the ship it belongs to (i.e. L_i, B_i, T_i, and C_{Bi}) all correspond to vessel i. The unlabeled case assumes that the indexes of each principal dimension variable do not necessarily correspond to the same ship (i.e. that the relations between the principal dimension values and associated ship are not known). This is representative of poor data traceability, which could be the result of incomplete communication of an analysis or poor data management. The introduction of the labeled and unlabeled cases both leave the network structure of Case 1 unchanged, and only affect the ability to conduct successful rework operations.

The different knowledge structures represented by Cases 1, 2 and 3 require different implementations in conducting rework. The algorithms associated with each rework activity are summarized in the flow charts presented in Figure 4.18.

The most simple process of conducting rework is on Case 3's knowledge structure. This requires the volume nodes in the network to be sorted, and the largest to be

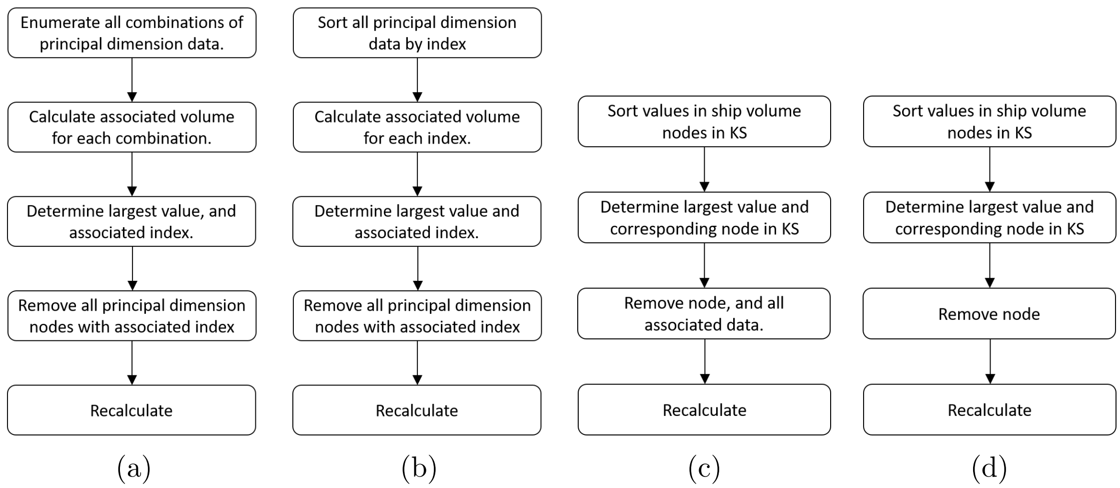
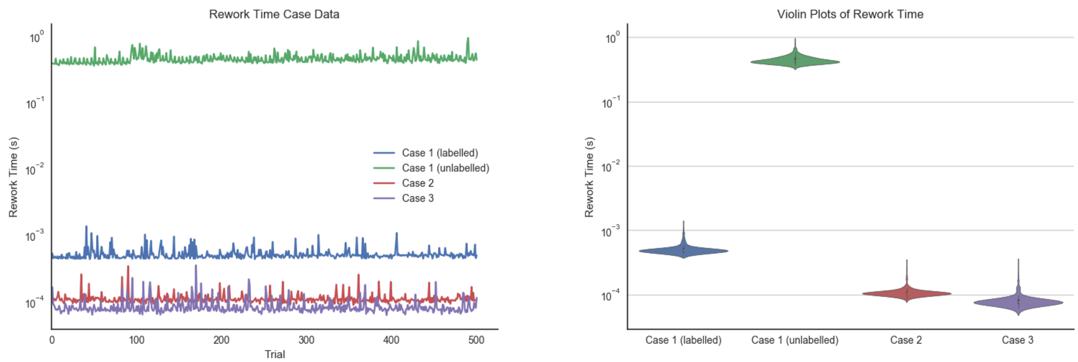


Figure 4.18: Flow charts for rework algorithms used for **(a)** Case 1 (unlabeled), **(b)** Case 1 (labeled), **(c)** Case 2, and **(d)** Case 3.

removed before the result can be re-calculated. Case 2's knowledge structure is a similar process, although it requires the supporting principal dimension data to be removed in addition to the removed volume node. Case 1 has the most difficult structure to conduct rework of the cases considered. The critical differences between the labeled and unlabeled versions of Case 1 are in the first step in the flow charts. For the labeled case, the associations between the principal dimension values are implied through their indexes, and only 14 calculations need to be conducted to determine the volumes for each ship. In the unlabeled case, the lack of indexes means that all possible combinations of principal dimension values must be enumerated to calculate an associated volume value for each. This requires $14^4 = 38,416$ volumes to be calculated. Thus, the presence of the indexes significantly simplifies the rework process.

The resulting time to conduct rework for Cases 1 (labeled and unlabeled), Case 2, and Case 3 are displayed in Figure 4.19. Figure 4.19a presents the raw data of the CPU times observed in conducting the rework activities, and 4.19b presents the associated violin plots for the raw data. The width of the violin plot corresponds to the probability of observing that value. Note that Case 4 has been omitted in the



(a) Rework Time Raw Data (b) Rework Time Violin Plots

Figure 4.19: Rework Time Results

figure, since the rework operations are conducted on the final network structure, and the final network structures of Case 1 and Case 4 are identical (hence, the rework times displayed for Case 1 will hold for Case 4 as well).

As can be seen in Figure 4.19, both knowledge structures and the traceability of data entities play a critical role in the ability to conduct successful rework activities. Both the labeled and unlabeled instances of Case 1 contain significantly higher rework times than for those of Case 2 and Case 3. In Cases 2 and 3, conducting rework is simply a task of sorting the volume nodes in the knowledge structure, finding the largest, and removing the associated data. Case 2 contains slightly higher rework times than that of Case 3, although they are not significantly different. This suggests that the increase in rework time associated with removing the principal dimension variables from the network in addition to the volume node itself is negligible. However, the knowledge structure of Case 1 has no explicit concept of individual ship volumes, and thus the steps required to determine the largest ship are more complicated. Conducting rework on Case 1 requires the principal dimension values to be combined to calculate a volume for each ship and sorted to determine the largest, then each associated data entity must be removed from the knowledge structure. This is an example of the critical role that knowledge structures play in supporting efficient rework activities.

The difference in observed rework times for the labeled and unlabeled versions of Case 1 illustrate the importance of data traceability in conducting rework. While the steps required to remove the outlier from the knowledge structure in the labeled case takes an order of magnitude more time than for Case 2 or 3, the unlabeled case increases the rework time to almost four orders of magnitude over Cases 2 and 3. The lack of data traceability means that in order to determine the largest ship, all combinations of principal dimension values must be calculated before the associated data can be removed. This represents significant extra effort, and is a prime example of excessive rework.

4.3.3 Failure to Integrate

In order to address the final question posed in the scenario outlined in Section 4.3, Cases 1 (labeled and unlabeled), 2 and 3 are considered in the context of conducting the rework process a second time. The rework algorithms and results from Section 4.3.2 were all able to identify the outlier as the largest ship across all presented methods. This section outlines the likelihood that each of the rework approaches would be able to determine and remove the second largest ship from the dataset. As the algorithms are unchanged from the case of determining the largest ship (only the selection operator of selecting the correct ship changes), the rework results from the previous section will remain the same for this discussion. However the probability of success of conducting rework for each changes drastically. This is used to demonstrate the increased potential of a failure to integrate information throughout a design activity.

A probabilistic approach was taken to demonstrate the potential failure to integrate for each case, which is summarized in Table 4.6. Consider the algorithmic flow charts presented in Figure 4.18. As previously stated, these algorithms are the necessary approaches required to revise each case's knowledge structure. These algorithms

Table 4.6: Failure to integrate in process of removing second largest ship.

	Case 1 (labeled)	**Case 1 (unlabeled)**	**Case 2**	**Case 3**
Required Calculations	14	38,416	0	0
Volumes to Sort	14	38,416	14	14
Index of Second Largest Volume	2	5810	2	2
P(Selecting Correct Ship)	1.0	0.0	1.0	1.0
P(Picking Ship Randomly)	$\frac{1}{14} \approx 0.07$	$\frac{1}{38416} \approx 3 \times 10^{-4}$	$\frac{1}{14} \approx 0.07$	$\frac{1}{14} \approx 0.07$
P(Recreating Full Dataset)	$\frac{1}{\binom{14}{14}} = 1.0$	$\frac{1}{\binom{38416}{14}} \approx 0.0$	$\frac{1}{\binom{14}{14}} = 1.0$	$\frac{1}{\binom{14}{14}} = 1.0$

remain unchanged from Section 4.3.2, except for the selection of the appropriate ship after the entries have been sorted. For Case 1 (labeled), Case 2 and Case 3, only 14 volumes need to be considered, corresponding to the 14 ships in the dataset. In Case 2 and 3, these volumes are explicitly represented in the knowledge structure, thus requiring no explicit calculations, while Case 1 (labeled) requires the 14 volumes to be calculated during the rework process. When there is no data traceability, such as in Case 1 (unlabeled), all combinations of variables must be enumerated, requiring $14^4 = 38,416$ volumes to be considered. None of these values are represented in the knowledge structure, thus each must be calculated explicitly. The volume values (either explicit or calculated) are then sorted in descending order to determine the second largest value of those considered. In Case 1 (labeled), Case 2, and Case 3, the second largest ship appears as the second index when sorted. This is intuitive, as only the 14 ships are considered during the sorting process. This means the sorting algorithms presented for these cases will be functional in selecting the correct ship and associated data. However, the same does not hold for the unlabeled case, in which the correct data corresponding to the second largest ship appears at index 5810, rather than index 2. This is because there are 5808 combinations of principal dimension values which yield higher values than those of the second largest ship - although these combinations do not represent any of the actual ships in the dataset.

This means that sorting the values and determining the second largest ship in the unlabeled case (at index 2) will remove the incorrect data entities from the knowledge structure and thus yield an incorrect result.

The expansion of both the number of volumes considered and number of calculations not only adds significant rework effort, but also decreases the probability of conducting a successful rework activity. This is evidenced by the last two rows in Table 4.6. With no data traceability, there is no ability to recover what specific combinations of variables were used to determine the yielded volumes, and so selection of the appropriate volume value is left to random chance. If a ship were to be selected randomly from each data set, the probability of that ship being the second largest decreases by 99.96% with the addition of the additional considered volumes. The probability of success diminishes even more when attempting to select all 14 correct ships from the datasets. For Case 2, Case 3, and Case 1 (labeled) the probability of recreating the entire correct dataset is 1 since the dataset contains only the 14 correct ships, whereas the additional erroneous volumes introduced in Case 1 (unlabeled) yields a probability of 5.7×10^{-54} of accurately recreating the correct dataset.

The inability to both conduct rework and to 'retrace steps' demonstrated by Case 1 (unlabeled) effectively illustrates an inability to progress with the design process by conducting rework. This roadblock represents an failure to integrate this design team's calculation approach (their knowledge structure) with the rest of the design effort. Solving this integration failure would require a full revision of the knowledge structure, but presents issues as decisions were likely already made throughout the design process based on the previous knowledge. Addressing this issue would represent a significant rework activity, and the reliance on the previous knowledge structure would likely lead to design churn later in the process.

4.4 Local Knowledge Structure Case Robustness

The analysis of local knowledge structure growth in Section 4.2, and suitability of the knowledge structure as a platform for conducting future work in Section 4.3, provide unique insights into the benefits and limitations of each approach. This section synthesizes the findings from previous sections to draw conclusions about which of the approaches contains the most conceptually robust knowledge structure.

The primary findings of the analysis conducted in this case study reveal a number of key factors that are critical to understanding the conceptual robustness of local knowledge structures:

1. Knowledge structure growth has a large impact on the ability to recognize and react to emergent design failures.

2. The initial selection of an information structure to begin the development of a knowledge structure (the initial hypothesis) in the knowledge development phase leads to the largest growth in topological entropy. This highlights the importance of selecting proper information sources in early design stages.

3. The knowledge refinement stage integrates additional information into the knowledge structure, by seeking supporting information sources for unknown knowledge entities. This only occurs when additional data are required to support the knowledge structure at a point in time. The presence of data alone does not require any knowledge refinement.

4. The knowledge refinement stage should be marked by relatively small increases in entropy. These gradual increases make it easier to identify 'flags' for potential emergent design failures. Initial knowledge refinement activities contribute more entropy than at later stages, illustrating decreased marginal gains as more data is added to the knowledge structure.

5. Impacts of adding data should be analyzed in the context of the effect on the knowledge structure target node to accurately determine applicability of additional engineering activities to the knowledge being sought.

6. Minimizing the time required to successfully conduct rework activities can be attained through a knowledge structure which best accounts for hidden data interdependencies.

7. Data and information traceability is critical in the ability to successfully conduct rework activities and prevent integration failures.

8. Approaches in which the minimum number of steps are required to yield a calculable answer should be taken. Once the answer is calculable, its evolution should be tracked over time to identify any increased potential for design churn and other emergent design failures.

9. Path dependencies should be minimized within calculations as a way of increasing predictability and monitoring the evolution of a design activity.

Considering the above conclusions in the context of the presented cases enables a new perspective of the conceptual robustness of each approach. Case 1 and Case 4 contained the same knowledge development phase, with markedly different knowledge refinement phases. Case 1 had a few knowledge refinement steps, with larger increases in topological entropy at each step, while Case 4 required more steps, with fewer marginal increases in topological entropy. This means more subtle changes to the development of the knowledge structure would be observable in Case 4 than would be in Case 1. Additionally, the knowledge refinement phase of Case 4 was marked by an immediate reduction in data status entropy, while Case 1's exhibited a more gradual decrease. This means in Case 1 the target value was only calculable at the final timestep in the knowledge refinement phase, as opposed to that of Case

4 which yielded target value results at each step throughout knowledge refinement. This means Case 4 yields more understanding of the evolution of the target value over time than Case 1, thus increasing the transparency of the impacts of adding data to the target value over time. This in turn leads to a better understanding of the impact of adding data to the knowledge structure over time. The resultant network structures of Case 1 and Case 4 yielded the same rework time results, and both highlighted the same propensity for integration issues when multiple rounds of rework were required. Thus, in comparing the two, the results suggest that Case 4 is more conceptually robust than Case 1, though Case 4's knowledge structure contains a larger potential for failure in future activities than that of Case 2 or 3.

Case 2 and Case 3 contained the same knowledge development phase, though Case 3 did not demonstrate a knowledge refinement phase. The presence of volume data with no context of the principal dimension data led to Case 3's knowledge structure being more limited in terms of both structure and growth dynamics than that of Case 2. The knowledge refinement demonstrated in Case 2 resulted in higher total topological entropy with a slowly decreasing data status entropy curve. Although this means portions of the network were incalculable, the monotonically decreasing curve indicates the team effectively worked toward increasing the certainty of known parameters. The target node remained calculable throughout the growth process which resulted in more calculated values, and more of an understanding of the evolution of the target node value over time as more data was added. Case 3 demonstrated an immediate decrease in data status entropy, and only one result was able to be calculated. In analyzing the resultant knowledge structures, both Case 2 and Case 3 required no additional calculations to be performed to conduct rework, due to the volume entities being explicitly represented in both knowledge structures. Both demonstrated low probabilities of rework issues, and the presence of the additional supporting data in Case 2 did not have a significant impact on rework time. This suggests Case 2 is a more conceptually robust approach than Case 3, as it integrates more information into the knowledge structure while highlighting more dependencies in the data.

The knowledge structures of Case 2 and Case 4 are significantly different, and lead to different growth and analysis dynamics. The knowledge development phase of Case 2 resulted in a higher topological entropy than that of Case 4 due to a larger network which encapsulates more data entities in the knowledge structure. The result of the larger initial network means a more gradual increase in topological entropy during the knowledge refinement phase in Case 2 than in Case 4, with a larger overall topological entropy at the final timestep. This means Case 2 incorporates more data into the knowledge structure by explicitly representing the intermediate volume variables than that of Case 4. Both Case 2 and Case 4 have gradual decreases in data status entropy over their growth, meaning they both provide insight into the evolution of the target node value over time. However, when looking at target value entropy, Case 2 was able to clearly determine flags for data inconsistencies over the course of the network growth, while the intermediate variable calculations of Case 4 caused these values to be 'washed out', leading to a reduced understanding of the causal relationships between adding data and resultant target values.

Another key differentiator between the conceptual robustness of Cases 2 and 4 is through the ability to conduct rework. The time required to conduct rework for Case 2 was significantly lower than for Case 4. This is due to Case 2 explicitly representing required data for rework in the knowledge structure, while Case 4 required these values to be calculated. It was shown that data traceability is a critical aspect for both rework times and probability of successful rework: Case 4 required indexes of data added to the structure to be carefully tracked, while Case 2 represents the data traceability explicitly through its structure (less dependency on tracking indexes). Additionally, the probability of an integration failure was shown to be far less for Case 2 than for Case 4.

The comparison of the cases considered in this section suggest that Case 2 is the most conceptually robust approach given its ability to accurately integrate the most data and information into its knowledge structure, and to capture hidden data dependencies. While this result will not come as a surprise to any Naval Architect - this case study has presented a novel quantitative method of considering conceptual robustness in the context of local knowledge structures.

CHAPTER V

Global Knowledge Integration Case Study

As outlined in Section 1.1, to fully understand conceptual robustness, knowledge-centric measures are required which extend beyond the traditional product-centric approaches implemented to date. A framework was presented in Chapter III which describes how knowledge structures evolve over time through the utilization of information sources. A case study was presented in Chapter IV which illustrates how the framework is utilized with respect to local knowledge structures, and the types of conceptual robustness insights it can yield. While these are critical first steps toward the development of a framework to study a local design activity, this chapter presents a case study on how the interaction of local agents leads to global information and knowledge structures, in the context of integrating knowledge towards the goals of a greater design activity. This case will demonstrate the successful utilization of the previously defined entropy metrics to analyze the global layers of the K-I Framework. Section 5.1 outlines the structure of this case study, and the remaining sections outline the resultant structures, dynamics, entropic results, and conceptual robustness insights gained through an examination of the case.

5.1 Case Study Overview

The case study presented in this chapter is a representative early stage design activity of creating an aircraft fuel distribution system to be implemented on a naval ship capable of launching and servicing a number of aircraft. This case study focuses on the integration of global design knowledge across disciplines over the course of a design activity, which is captured through the evolution of global information.

The aircraft fuel distribution system presented in this case study is a representative example of an early stage Analysis of Alternatives (AoA) design activity. This example has been selected to illustrate the growth of global information and knowledge in the information-sparse, multi-agent environment which plagues early design stages. As such, the presented distributed system design tools have intentionally been created to match the types of simplistic tools used to conduct an AoA. The case study not only enables direct comparisons across combinations of aircraft load-outs, but also highlights the differences of information flow within the different design activities as well as the robustness of a design activity in the presence of little prior information.

The AoA design task which this case study considers is outlined as follows. Principal dimensions have been determined for a naval vessel capable of supporting the operations of helicopters and aircraft capable of Vertical Takeoff and Landing (VTOL). These characteristics are presented in Table 5.1. The alternative in question has a flight-deck which spans its entire length, and as such, a wide range of potential aircraft could be feasibly deployed from the vessel. The vessel is required to support the launch and recovery of these aircraft, as well as refuel them between sorties. Four potential aircraft have been identified as feasible alternatives for the vessel to support: F-35B's, V-22 Osprey's, Sikorsky SH-60 Seahawks, and AV-8B Harriers. It is assumed that all considered aircraft utilize the same fuel (JP-5) for their operation. It is up to the designers to determine a feasible JP-5 distribution system to support a specific load-out of aircraft. The representative parameters of each vehicle type are

Table 5.1: Representative Case Study Hull Parameters

Hull Parameter	Value	Units
Length (LWL)	237.0	m
Beam (B)	32.0	m
Draft (T)	8.5	m
Depth (D)	20.0	m
Block Coefficient (Cb)	0.64	-
BMT	6.0	m
BML	180.0	m
KB	4.3	m
LCB	118.5	m from AP

listed in Table 5.2.

A team of designers with different expertise have been tasked with the conducting the design analysis for the presented vessel. The design team is comprised of three groups: a *Naval Architecture* group (NAVARCH), a *Flight Operations* group (OPS), and a *Distribution* group (DIST). Each of these groups is responsible for conducting a subset of design analyses to produce the integrated system. In order to produce a converged design, each of these groups will be required to conduct individual analyses using their own design tools, and may also be required to communicate data to other groups.

This case study represents an AoA design process with diverse, non-co-located design teams, and presents an example of how these design teams would approach such a distributed system design problem. Additionally, the presented case enables

Table 5.2: Representative Case Study Aircraft Parameters

Aircraft	Unit Weight ($tonnes$)	Combat Radius (nm)	Cruise Speed (kts)	Fuel Capacity (m^3)
F35-C	13.3	600.0	567.0	17.0
V22 Osprey	15.0	428.0	270.0	9.2
AV-8B Harrier	6.7	300.0	550.0	7.4
SH-60 Seahawk	8.1	200.0	120.0	2.0

an examination of global information and knowledge structures based on how teams interact with one another. It is important to note that this case study does not consider the creation of each discipline's design tool as a part of the presented design process, and it is assumed that these tools have been created a-priori. The result of this assumption is that local knowledge structures are fixed through time, and represent the embedded knowledge contained within the tool. A more detailed explanation of the local tools used by each design group, and the resultant local knowledge structures are presented in Sections 5.2.1 - 5.2.3.

To study the evolution of global information and knowledge structures, two scenarios are considered which highlight different inter-discipline communication dynamics. These two scenarios, a *Simple Case* and a *Hard Case*, represent different combinations of aircraft as requirements for the designers to meet. In the *Simple Case*, the required load-out of aircraft is comprised of a single type. The *Hard Case* considers a suite of four different aircraft types. In both cases, designers must design a JP-5 distribution system to accommodate the full load-out of aircraft, while ensuring it remains feasible within the vessel constraints. It will be shown that the inter-agent communication dynamics differ significantly between the two cases, and reveal a number of conceptual robustness insights as it pertains to the integration of design information and knowledge at a global level.

5.1.1 Requirements

As described in Chapter III, a design activity begins with a number of known and unknown global knowledge entities, pertaining to the global design. It is the task of the designers to uncover the relations and dependencies of the known entities on the unknown entities, and also between the unknown entities at a global level. In this case, the known knowledge entities are defined by the requirements, and the unknown parameters are the global design parameters selected to be determined. These entities exist in the global knowledge layer of the K-I Framework.

The design requirements for the *Simple* and *Hard* cases set global design goals for the designers to meet. These requirements define the feasibility constraints of the final design, and as such, play a critical role in the ability for designers to produce a converged product. The design requirements are used to differentiate the two cases by varying the design space constraints and number of design variables required to be considered. The *Simple* case reduces both the number of design constraints and number of variables considered in the analysis, while the hard case increases both. In both cases the requirements pertain to the load-out of aircraft considered for the ship design to support. Specifically, the requirements define the number of each type of aircraft to be accommodated by the design. The required values differ between the two considered cases, while the desired knowledge entities are held constant for comparison purposes. The desired global knowledge entities common to both cases are presented in Table 5.3. The order in which the unknown entities are listed in the table correspond to the sequence in which they are solved for in the design activity. The requirements for aircraft load-outs are presented in Tables 5.4 and 5.5 for the *Simple* and *Hard* cases, respectively.

Table 5.3: Desired knowledge entities common to both cases.

Knowledge Entity	**Status**	**Description**	**Units**
No. F-35B	Known	Required number of F-35B Aircraft	#
No. V-22 Osprey	Known	Required number of V-22 Aircraft	#
No. SH-60 Seahawk	Known	Required number of SH-60 Aircraft	#
No. AV-8B Harrier	Known	Required number of AV-8B Aircraft	#
GM_t	Unknown	Transverse Metacentric Height	m
%	Unknown	Difference between Total Weight and Displacement	%
Trim	Unknown	Difference between forward and aft draft	m
Required Power	Unknown	Power required by pumping system	kW
Pipe Diameter	Unknown	Diameter of pipe used for JP-5 distribution	m

Table 5.4: Aircraft load-out requirements (Simple Case)

Aircraft	Number
F-35B	0
V-22 Osprey	0
SH-60 Seahawk	36
AV-8B Harrier	0

Table 5.5: Aircraft load-out requirements (Hard Case)

Aircraft	Number
F-35B	1
V-22 Osprey	2
SH-60 Seahawk	3
AV-8B Harrier	8

5.2 Local Knowledge Structures

5.2.1 Operations (OPS) Group

The Operations (OPS) group is responsible for determining the parameters of the distribution system pertaining to the selected aircraft. At a high level, they are responsible for determining the aircraft locations and the required capabilities of the distribution system for a given aircraft load-out. More specifically, the OPS team determines estimates for the total weight of aircraft and the locations of the aircraft on the flight deck. The locations of aircraft on the flight deck are determined assuming aircraft of the same type are grouped together and must be sufficiently spaced between other groups. The net center of gravity of each aircraft type is decided by the OPS team. Additionally, the OPS team determines the refueling location, the required volume of JP-5 required onboard, and the volume flow rate to support all aircraft operations. As such, the OPS team essentially translates the required aircraft array to distributed system parameters.

As was previously mentioned, the process of creating the design tool is assumed to have occurred prior to this design activity, and as such, the local knowledge structure growth has already occurred. Hence, the OPS team's local knowledge structure has already been fully defined, and is a representation of the embedded knowledge within their design tool. The OPS team's local knowledge structure is presented in Figure 5.1.

The OPS local knowledge structure illustrated in Figure 5.1 illustrates the relations of local knowledge entities embedded within their design tool. Nodes are sized according to their PageRank value. Input variables are depicted as nodes with zero in-degree (shown in green), outputs of the tool are represented by nodes with zero out-degree (shown in blue), and all other nodes represent intermediate variables as a result of intermediate calculations (shown in grey). Table 5.6 presents each variable ordered by PageRank value, with the associated in-degree and out degree. Note that

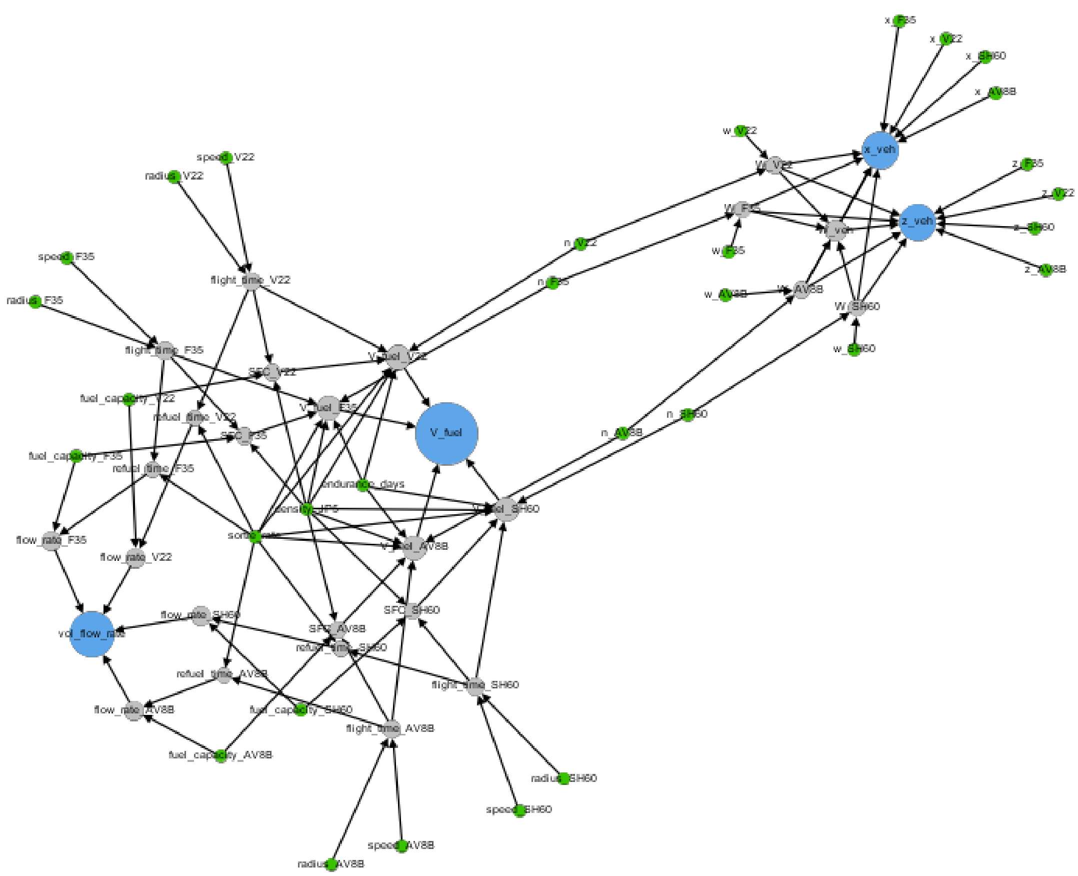


Figure 5.1: OPS Local Knowledge Structure

the output variables contain the largest PageRank values, while the inputs have the smallest. These values are predicated by the structure of the local network. Note that while the PageRank values are predicated by the network structure, the data values of the inputs to the tool is independent of structures and could be decided by the team itself or the results of another team. The process of populating the inputs to the tool will be a product of how the design approach is conducted. The equations used by the OPS team are presented in Appendix A.3.

The visualization of the OPS team's local knowledge structure reveals two communities of variables. The smaller community in the top right portion of the figure contains the parameters used to determine the weights and centers of the aircraft. The larger community on the bottom left of the figure contains the parameters used to calculate the volume of JP-5 required onboard, and the required distribution system flow rates. Comparatively, the larger community is comprised of many more intermediate variables and contains a more complex structure than that of the smaller community. This indicates that there are more parameters contained in conducting the calculation, and that the dependencies between variables are more complex for determining fuel estimates than for calculating weights and centers. Note that the two communities are connected through four input nodes which correspond to the number of each vehicle type. As these four nodes are inputs (and hence have no in-degree) there are no directed paths through the knowledge structure between the communities. This indicates that the fuel-related parameters are not functionally dependent on the weights and centers of the aircraft (and vice-versa) - both analyses are predicated by the number of each aircraft type.

Table 5.6: Variables contained in Operations Tool

Variable Name	In-Degree	Out-Degree	PageRank
V_fuel	4	0	0.1130
vol_flow_rate	4	0	0.0769
x_veh	9	0	0.0581
z_veh	9	0	0.0581
V_fuel_F35	6	1	0.0312
V_fuel_V22	6	1	0.0312
V_fuel_AV8B	6	1	0.0312
V_fuel_SH60	6	1	0.0312
W_veh	4	2	0.0245
flow_rate_F35	2	1	0.0206
flow_rate_V22	2	1	0.0206
flow_rate_AV8B	2	1	0.0206
flow_rate_SH60	2	1	0.0206
flight_time_F35	2	3	0.0185
flight_time_V22	2	3	0.0185
flight_time_AV8B	2	3	0.0185
flight_time_SH60	2	3	0.0185
SFC_F35	3	1	0.0157
SFC_V22	3	1	0.0157
SFC_AV8B	3	1	0.0157
SFC_SH60	3	1	0.0157
W_V22	2	3	0.0156
W_AV8B	2	3	0.0156
W_SH60	2	3	0.0156
W_F35	2	3	0.0156
refuel_time_F35	2	1	0.0128
refuel_time_V22	2	1	0.0128
refuel_time_AV8B	2	1	0.0128
refuel_time_SH60	2	1	0.0128
n_F35	0	2	0.0068
n_V22	0	2	0.0068
n_AV8B	0	2	0.0068
n_SH60	0	2	0.0068
w_F35	0	1	0.0068
w_V22	0	1	0.0068
w_AV8B	0	1	0.0068
w_SH60	0	1	0.0068
sortie_rate	0	8	0.0068
fuel_capacity_SH60	0	2	0.0068
fuel_capacity_F35	0	2	0.0068
endurance_days	0	4	0.0068
density_JP5	0	8	0.0068

Table 5.6 (continued)

Variable Name	In-Degree	Out-Degree	PageRank
fuel_capacity_AV8B	0	2	0.0068
fuel_capacity_V22	0	2	0.0068
radius_F35	0	1	0.0068
radius_V22	0	1	0.0068
radius_AV8B	0	1	0.0068
radius_SH60	0	1	0.0068
speed_F35	0	1	0.0068
speed_V22	0	1	0.0068
speed_AV8B	0	1	0.0068
speed_SH60	0	1	0.0068
x_F35	0	1	0.0068
z_F35	0	1	0.0068
x_V22	0	1	0.0068
z_V22	0	1	0.0068
x_AV8B	0	1	0.0068
z_AV8B	0	1	0.0068
x_SH60	0	1	0.0068
z_SH60	0	1	0.0068

5.2.2 Naval Architecture (NAVARCH) Group

The NAVARCH group is responsible for ensuring the stability, trim, and weights and centers of the vessel remain feasible for a given arrangement of aircraft. At a high level, the NAVARCH group is responsible for ensuring that the physical parameters of the distribution system are integrated appropriately with the vessel's hull parameters. More specifically, the NAVARCH group ensures the weights and locations of the aircraft and fuel tanks are located in such a way that the vessel's GM_T, Trim, and difference between total weight and displacement remain feasible. In this respect, the NAVARCH group act in many ways as the system integrators by ensuring the calculations performed by other design teams translate to feasible ship parameters.

Similar to the OPS team, the NAVARCH group utilizes a previously developed tool to conduct their design analysis. The resulting local knowledge structure of the knowledge embedded in the design tool is depicted in Figure 5.2, with inputs shown in green, outputs shown in blue, and intermediate variables shown in grey. Nodes are sized by their PageRank value, and are summarized in Table 5.7. More details about the equations used in the design tool is presented in Appendix A.2.

The NAVARCH team's knowledge structure reveals a larger ratio of inputs to outputs compared to the OPS team. This is intuitive, due to the large number of NAVARCH input variables related to the hullform parameters of the ship in question. As the NAVARCH team acts in more of an integration role, many inputs relate to outputs from other team's tools (for example, the weights and centers of the aircraft from the OPS team). Interestingly, unlike the OPS team's knowledge structure, the NAVARCH team's knowledge structure does not reveal clearly separable communities

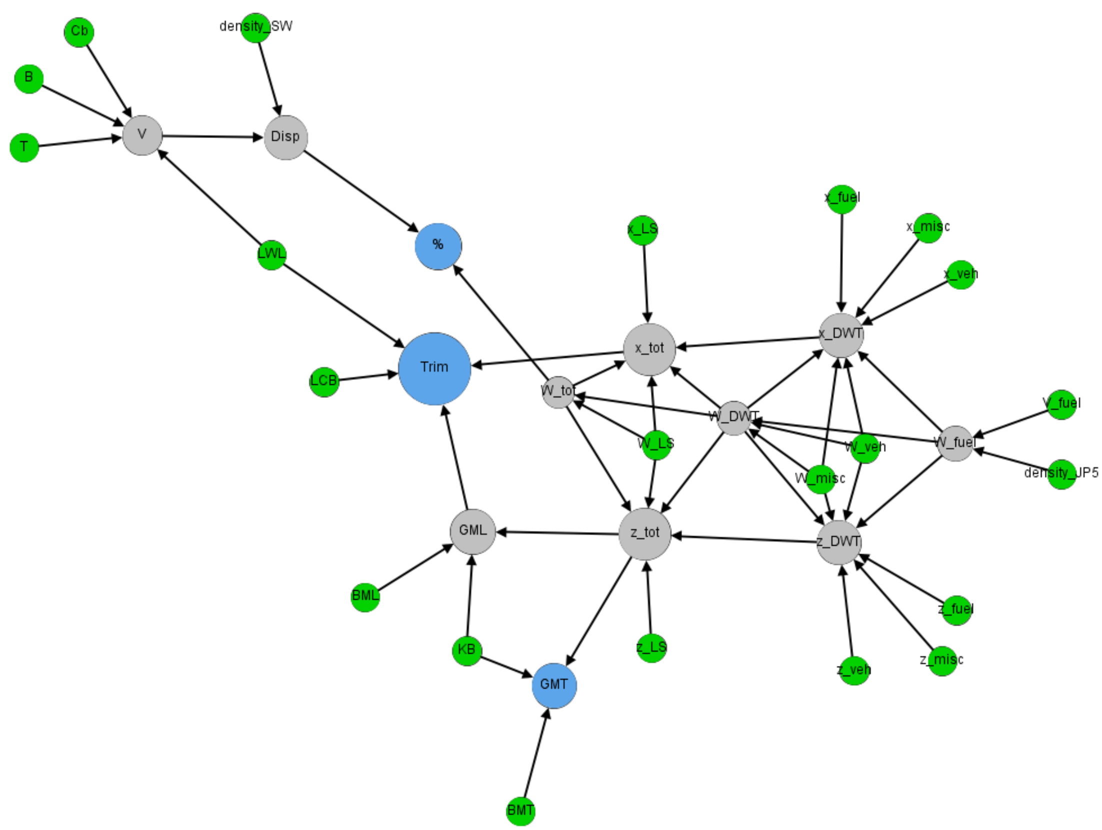


Figure 5.2: NAVARCH Local Knowledge Structure

Table 5.7: Variables contained in Naval Architecture Tool

Variable Name	In Degree	Out Degree	PageRank
Trim	4	0	0.1420
x_tot	5	1	0.0801
z_tot	5	2	0.0801
%	2	0	0.0633
GMT	3	0	0.0583
GML	3	1	0.0583
x_DWT	7	1	0.0564
z_DWT	7	1	0.0564
Disp	2	1	0.0559
V	4	1	0.0425
W_fuel	2	3	0.0289
W_DWT	3	5	0.0249
W_tot	2	3	0.0179
LWL	0	2	0.0107
B	0	1	0.0107
T	0	1	0.0107
Cb	0	1	0.0107
BMT	0	1	0.0107
BML	0	1	0.0107
KB	0	2	0.0107
LCB	0	1	0.0107
density_SW	0	1	0.0107
W_LS	0	3	0.0107
W_misc	0	3	0.0107
W_veh	0	3	0.0107
density_JP5	0	1	0.0107
x_misc	0	1	0.0107
z_misc	0	1	0.0107
x_veh	0	1	0.0107
z_veh	0	1	0.0107
x_fuel	0	1	0.0107
z_fuel	0	1	0.0107
x_LS	0	1	0.0107
z_LS	0	1	0.0107
V_fuel	0	1	0.0107

by input variable. This suggests the variables are far more inter-related than that of the OPS team, which implies that changes to a variable will have a larger number of impacts on other nodes in the network.

5.2.3 Distribution (DIST) Group

The DIST group is responsible for determining the preliminary estimates for the parameters pertaining to the piping system used to move the fuel from the ship's JP-5 tanks to the location of the aircraft. Primarily, the DIST group is responsible for determining the required pipe diameter for the distribution system and the required power of a pump to move the fuel from the tanks to the flight deck. As such, the DIST team is responsible for determining the feasibility of moving the fuel around the vessel. The embedded knowledge in the tool (and hence their local knowledge structure) is shown in Figure 5.3, with inputs shown in green, outputs shown in blue, and intermediate variables shown in grey. Nodes are sized by their PageRank value. Similar to the other groups, the nodes in the network are summarized in Table 5.8. The equations used by the DIST team are presented in Appendix A.1.

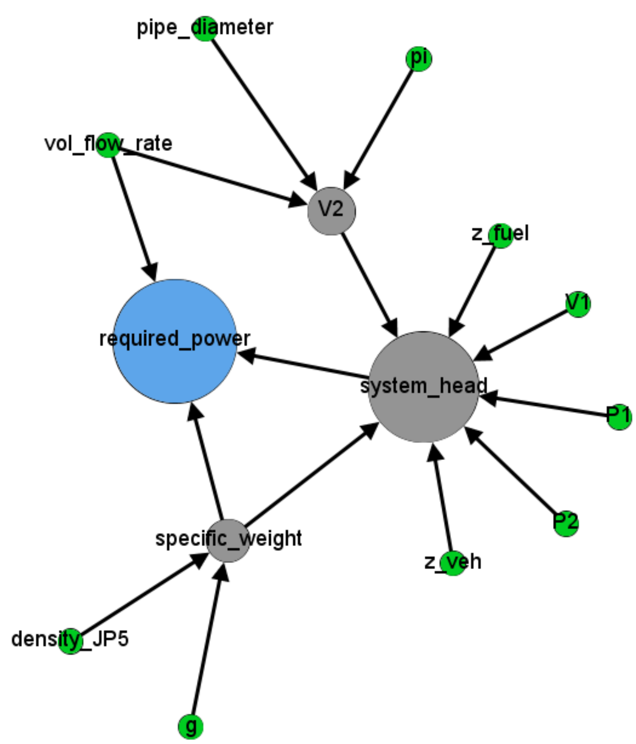


Figure 5.3: DIST Local Knowledge Structure

Table 5.8: Variables contained in Distribution Tool

Variable Name	In Degree	Out Degree	PageRank
required_power	3	0	0.2921
system_head	7	1	0.2576
V2	3	1	0.0889
specific_weight	2	2	0.0768
density_JP5	0	1	0.0285
g	0	1	0.0285
pipe_diameter	0	1	0.0285
vol_flow_rate	0	2	0.0285
pi	0	1	0.0285
P1	0	1	0.0285
P2	0	1	0.0285
z_fuel	0	1	0.0285
z_veh	0	1	0.0285
V1	0	1	0.0285

Compared to the other groups' structures, the DIST team has the simplest network. No clear communities exist, which suggests higher levels of knowledge dependencies in the tool; however, the minimal size of the network limits the structure's complexity. Additionally, the few intermediate variables in the network means inputs are more directly related to calculate outputs which contributes to the tool's simplistic structure.

5.2.4 Summary of Group Roles

Sections 5.1.1 - 5.2.3 have presented a case study which involves the coordination of multiple teams and their associated tools to meet a set of requirements pertaining to a feasible final design. The responsibilities of each team have been outlined, in addition to their local knowledge structures, which are results of their previously developed design tools. The summary of each design group's goals is presented in Table 5.9.

An examination of the teams' local knowledge structures reveals a number of insights about the embedded knowledge within each of their design tools. The number

Table 5.9: Summary of Case Study Design Group Roles

Design Group	Responsibilites	Parameters
Operations (OPS)	Determine estimates of distribution system requirements based on the proposed combination of aircraft.	Aircraft Weights Aircraft Locations Refueling Location Required Volume of Fuel Required Flow Rate
Naval Architecture (NAVARCH)	Determine the weights and centers of distribution system, and ensure broad feasibility of vessel related to weight vs. displacement, stability, and trim.	Weight of Fuel Location of Fuel Tank Transverse Metacentric Height (GM_t) Trim Difference between Total Weight and Displacement
Distribution (DIST)	Sizing of piping system and pump sizing required to transfer fuel from tanks to aircraft refueling location.	Pipe Diameter Required Pump Power

of nodes (knowledge entities) in each local knowledge network corresponds to the number of variables contained in the tool. The number of edges represents the number of functional dependencies between knowledge entities. The combination of these two aspects defines each network, and provides insight into the complexity and scope of each team's knowledge structure. A summary of each local knowledge structure is presented in Table 5.10.

The OPS network has the largest number of nodes and edges, which illustrates that the tool utilized considers the most variables and most interdependencies. While it is to be expected that a larger number of nodes leads to a larger number of edges,

Table 5.10: Summary of Local Knowledge Structure Parameters

Local KS	Nodes	Edges	Average Degree	Average Path Length
Operations (OPS)	60	98	1.633	1.734
Naval Architecture (NAVARCH)	35	49	1.400	2.043
Distribution (DIST)	14	15	1.071	1.567

the OPS network also has the highest average degree, which indicates that on average, each node in the network is connected to more nodes than the other groups. The clear separation of two communities in the OPS knowledge structure illustrates distinct groups of variables used to calculate different outputs. While this limits the functional dependencies of the variables in the entire structure, the large number of nodes results in a large number of dependencies being recognized within communities. This is reflected in the average path length, which falls in between the two other groups. In this case, average path length is indicative of the average depth of dependencies in the network.

The NAVARCH group has the second most nodes (variables) and edges (interdependencies), and contains a less clear community structure. The less defined separability of the network is illustrated by the highest average path length, and indicates that on average a node in the network has a longer depth of impact than the other teams. The average degree remains high for the NAVARCH team, and is in fact larger than the OPS team relative to the number of nodes in the network. This indicates that changes to a single input variable are more likely to impact a number of output variables than in the OPS case. Additionally, it suggests output variables are more closely related to one another through the interdependencies in the variables.

The DIST team has the least number of nodes and edges, which suggests their tool is the most rudimentary of the three groups. Low average degree and path length metrics highlight the simplistic network structure and demonstrates the least amount of complexity.

This section has provided a qualitative discussion of each local knowledge structure. A more quantitative discussion of each network structure is presented in discussion of Topological Entropy (TE) presented in Section 5.4.1.

5.3 Global Information and Knowledge Structures

5.3.1 Integration Sequence

In order to compare the results of conducting the design activity between the simple and hard cases, the sequence in which target nodes are solved is held constant (as stated in the discussion of Table 5.3). As such, prescribing the sequence a-priori enables the same design process to be analyzed in the context of accommodating the two sets of requirements in the AoA. The sequence of target nodes are listed in Table 5.11 along with the corresponding teams responsible for that node.

Each target node listed in Table 5.11 is selected by the team responsible for that node. That team attempts to calculate a value for that node. If a team is unable to calculate a value for that target node due to a lack of data in their local knowledge structure, they communicate their unknowns to the global information layer. These unknowns are listed on the right side of the table with the teams responsible for those nodes. Note that the transition of some nodes to global information creates a number of additional unknowns, while others do not. This occurs when the data required for a local node already exists within the local knowledge structure, or as data in a global information node. In the latter case, rather than creating a new unknown, the team simply utilizes the existing value in global information. Note that the creation

Table 5.11: Prescribed Sequence of Global Information Integration

Order	**Unknown Target Node**	**Team Responsible**	**Unknown Node Created**	**Team Responsible**
1	GM_t	*NAVARCH*	W_veh z_veh V_fuel z_fuel	OPS OPS OPS NAVARCH
2	*%*	*NAVARCH*	-	-
3	*Trim*	*NAVARCH*	x_veh	OPS
4	*required_power*	*DIST*	vol_flow_rate	OPS
5	*pipe_diameter*	*DIST*	-	-

of nodes does not necessitate the initialization of values - nodes can remain unknown until populated by a local layer at a later time.

As was explained in Section 3.2.1, the interaction of local layers with the global information layer are represented through inter-layer edges. The paths created between local knowledge structures through their interactions with global information entities creates projected values in the global information layer. Hence, the provided sequence prescribes both the order in which nodes are created, and in which interdependencies are realized, in global information. As the same global entities are provided for both cases and solved for in the same order, the resulting global structures will be the same.

While the target node sequence is fixed between simple and hard cases - the solution process between teams differs. As the simple and hard cases contain different values associated with many local and global nodes, the number of negotiations required between teams differ in order to yield suitable values for the respective teams. Additionally, changes made during negotiations may necessitate revisions to other previously negotiated nodes, and hence require different durations to conduct various analyses. Hence, the global dynamics will differ between the simple and hard cases in the form of updates to global information values, which provides the needed data to test the methods created.

The resultant K-I Framework for the prescribed knowledge integration procedure is displayed in Figure 5.4. In this image, the OPS, NAVARCH, and DIST teams' knowledge structures are displayed as the top-most networks, and are shown in magenta, green, and orange, respectively. The global information and knowledge layers are shown in blue and dark green, respectively. Edge color indicates the layer of the node being pointed from, and blended colors highlight the presence of bi-directional edges.

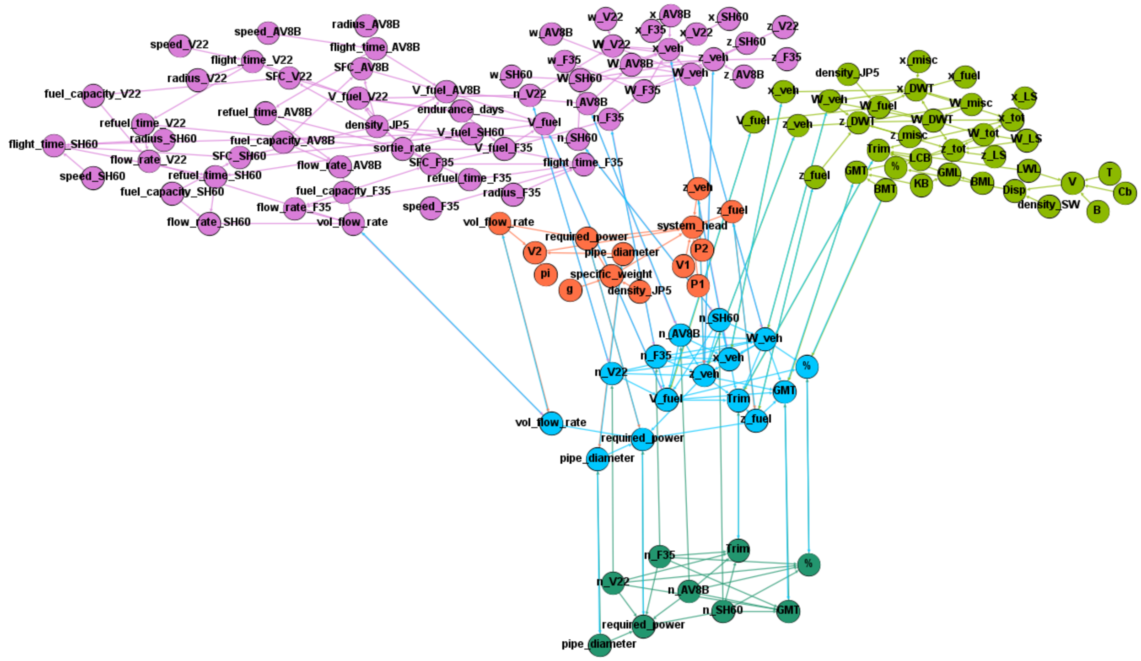


Figure 5.4: Resultant K-I Framework

5.3.2 Global Information Structure

The resultant global information structure for both the simple and hard cases is depicted in Figure 5.5. The network contains the nodes initially translated from the global knowledge layer (see Table 5.3) and the additional nodes transmitted as a result of inter-team negotiations (see Table 5.11). The edges in the network are projections of dependencies between entities based on the interactions of local knowledge structures. Nodes are sized according to their PageRank values, and a summary of node characteristics is provided in Table 5.12. In the table, nodes initialized from the global knowledge layer are highlighted in grey.

The developed global information structure reveals a number of aspects worth noting. First, note that nodes pertaining to the number of aircraft which originated as known global knowledge entities, all have zero in degree but non-zero out degree. This indicates that these parameters are only used to determine other values, but are themselves not dependent on other entities. As such, they act as inputs to the global

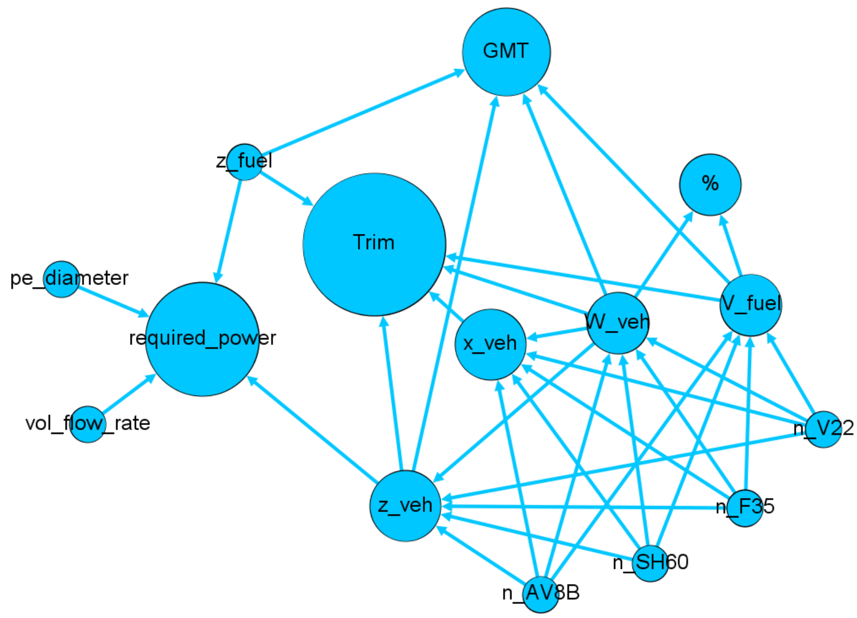


Figure 5.5: Resultant Global Information Structure

information structure. This is intuitive, as these parameters dictate the integration activity. Second, note that *Trim*, GM_t, *required_power* and % all have zero out degree, and non-zero in degree. This illustrates that these parameters are solely

Table 5.12: Variables contained in Global Information

Variable Name	In Degree	Out Degree	PageRank
Trim	5	0	0.1647
required_power	4	0	0.1295
GM_t	4	0	0.0985
z_veh	5	3	0.0779
x_veh	5	1	0.0779
W_veh	4	5	0.0666
V_fuel	4	3	0.0666
%	2	0	0.0662
n_F35	0	4	0.0360
n_V22	0	4	0.0360
n_AV8B	0	4	0.0360
n_SH60	0	4	0.0360
pipe_diameter	0	1	0.0360
z_fuel	0	3	0.0360

dependent on other parameters, and are not used to determine any others. Hence, these act as outputs of the information structure. Note however that *pipe_diameter*, a node translated from the global knowledge, has been determined as an input rather than an output. This illustrates that the recognition of interdependencies in global information is independent of where the node is initialized from. All other nodes in the structure are intermediate nodes, and create dependency pathways between inputs and outputs. The PageRank of intermediate nodes are indicative of how important they are within the structure.

5.3.3 Global Knowledge Structure

The global knowledge structure created by the simple and hard cases is depicted in Figure 5.6. This network is comprised of all nodes initialized into the framework (Table 5.3). Upon initialization these nodes were unconnected; however, the resultant structure includes projected edges uncovered through the development of the global

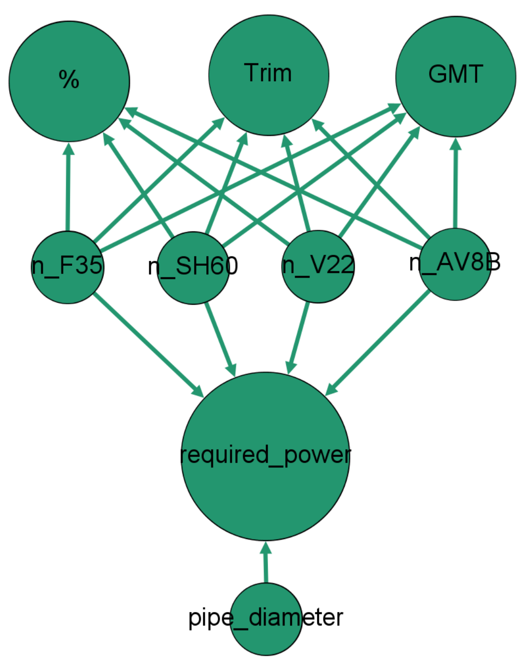


Figure 5.6: Resultant Global Knowledge Structure

Table 5.13: Variables contained in Global Knowledge

Variable Name	In Degree	Out Degree	PageRank
required_power	5	0	0.2037
GM_t	4	0	0.1396
%	4	0	0.1396
Trim	4	0	0.1396
n_F35	0	4	0.0755
n_V22	0	4	0.0755
n_AV8B	0	4	0.0755
n_SH60	0	4	0.0755
pipe_diameter	0	1	0.0755

information structure (through the process outlined in Section 3.2.2). Nodes in the figure have been sized according to their PageRank values, and a summary of node parameters is presented in Table 5.13. In the table, previously known parameters are highlighted in grey, while unknown parameters are unshaded.

The resulting global knowledge structure effectively relates the previously known knowledge entities to those that are unknown. Note that the network is fully connected, which illustrates that the teams were able to determine relations between all entities. All previously known entities have zero in degree and non-zero out degree, which indicate they are only used to determine other parameters, becoming the inputs to the global knowledge structure. The previously unknown entities have been uncovered as both inputs and outputs in the structure, through the pathways uncovered in global information. This successfully demonstrates the ability for the framework to recognize proper interdependencies between knowledge entities independent through appropriate information building activities. An important aspect of the global knowledge structure is that it does not contain intermediate nodes. The intermediate nodes in global information are used to recognize dependency pathways and project these dependencies into global knowledge. No new nodes are added to the structure throughout the presented case; the process simply uncovers relations between all initial nodes.

While both the simple and hard cases yield identical global knowledge structures, the dynamics and robustness of the approaches differ. This will be explored further in the next section through the application of the developed entropy metrics.

5.4 Knowledge Integration Results

This section presents the temporal results of the simulated design process created in this case study. The developed entropy metrics are applied to the K-I Framework over the network developments, and are used to understand the evolution of both local and global knowledge structures. First, the growth of each layer is analyzed in Section 5.4.1 using the Topological Entropy (TE) metric. Second, the temporal calculability results of local and global layers are presented in Section 5.4.2. Finally, the evolution of node values across the layers are analyzed in Section 5.4.3 which considers both input and output values, as well as intermediate values used over the course of their calculation. Each section presents the conceptual robustness insights that can be gained from utilizing each metric at the local and global levels.

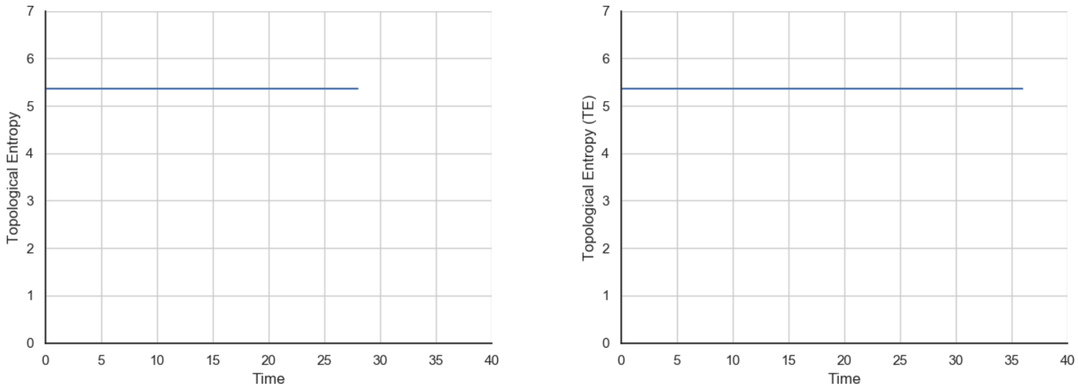
5.4.1 Topological Entropy

An examination of the Topological Entropy (TE) provides insight into how the amount of information contained within the structure of the network changes over time. As such, TE time series are displayed for each local team and the global layers in Figures 5.7 - 5.11 for both the simple and hard cases. These figures display the TE as a function of the number of framework timesteps. These framework timesteps do not correlate to physical time, but rather correspond to the steps of transmitting information either within or across layers, providing a more intuitive means for discussing the results. Of note is that the total number of framework timesteps required to complete the design activity differs between the two cases, with values of 28 and 36 steps, respectively. The additional steps in the hard case are a result of additional required communication between groups which are not present in the simple case. The TE dynamics differ across the two cases, which will be discussed in more detail in the remainder of this section.

There is no growth of the local knowledge structures (Figures 5.1 - 5.3) throughout this representative design activity in the simple or hard cases. This is the result of the same tools being used by each team in both cases, neither of which required to integrate additional knowledge elements. As the tools used by each team are the same, so too is the knowledge embedded in these tools. The TE plots of the local knowledge structures are presented in Figures 5.7 - 5.9. The unchanging local layer network structure leads to constant TE values, though the hard case TE spans a longer duration. This illustrates that there is no change to the structure of the knowledge embedded in their tools over time, although the data utilized across the tools may indeed change (this will be explained further in Section 5.4.2).

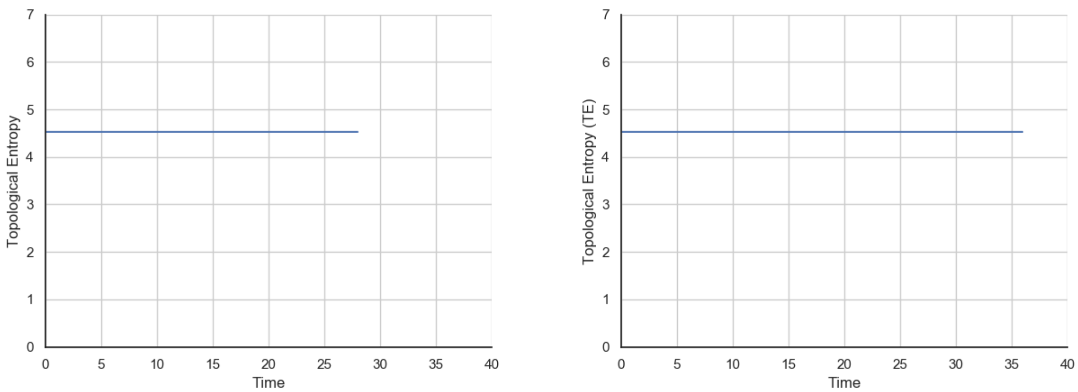
The networks presented in Figures 5.1 - 5.3 demonstrate the constant structure over the course of the design activity. While this provides a broad perspective of the network's evolution over time, the structure can be analyzed in more detail by understanding the contribution of input, intermediate, and output variables from a TE perspective. To better understand the composition of these structures by variable type, consider Figure 5.10.

Figure 5.10 illustrates the contribution of inputs, intermediate, and output variables to the total observed TE in the time series for each of the local knowledge structures (Figures 5.1 - 5.3). Table 5.14 summarizes these results, and includes the percentage contribution to the total TE for each team. While the number of inputs varies greatly between each local structure, the TE of the inputs remains relatively constant for each. This indicates that from a structural standpoint, the TE contribution of each individual input is diminished as the number of inputs increases. This is due to the absence of incident edges to input nodes, meaning they do not aid in



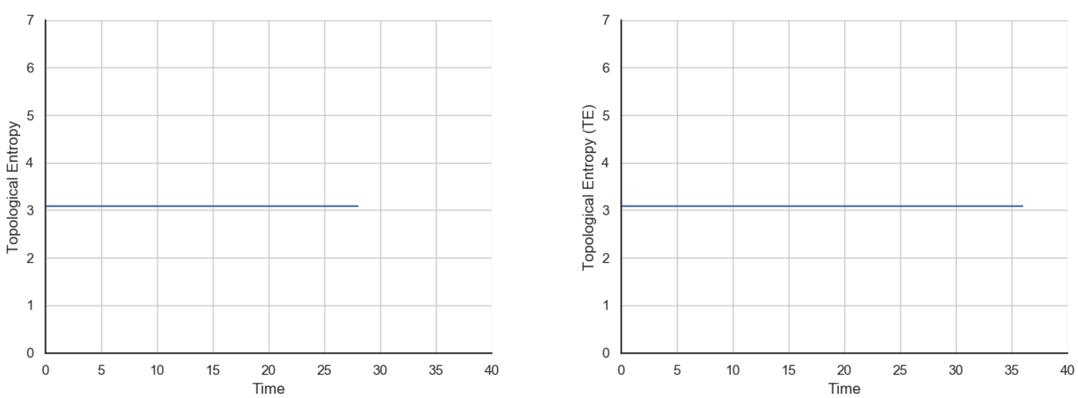
(a) Simple Case (b) Hard Case

Figure 5.7: Topological Entropy - OPS Team



(a) Simple Case (b) Hard Case

Figure 5.8: Topological Entropy - NAVARCH Team



(a) Simple Case (b) Hard Case

Figure 5.9: Topological Entropy - DIST Team

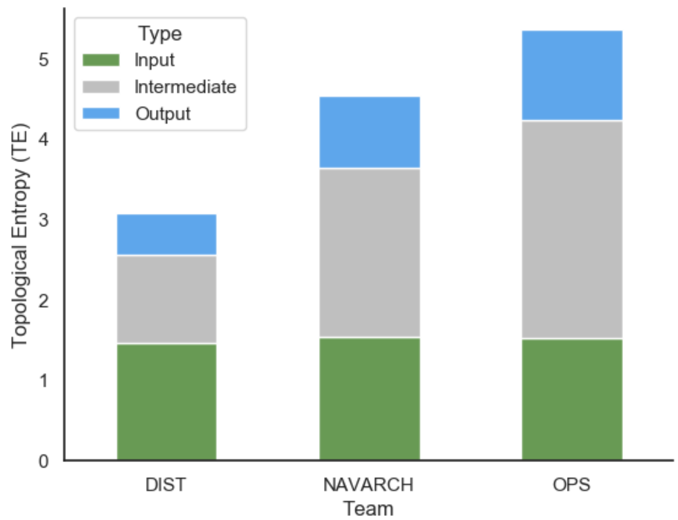


Figure 5.10: Team Topological Entropy (TE) Comparison

the transmission of data through the network - they are only the starting point of transmissions. This is confirmed by the diminishing percentage contributions for the inputs toward total TE.

The observed differences in total TE are mainly a result of increases in intermediate and output variables. The intermediate variables act as transitional nodes between inputs and outputs. A larger number of intermediate nodes coupled with more interdependencies between them lead to more structural complexity, and hence more TE. Not only does the TE of the intermediate values increase, but so too does the percentage contribution. This explains the larger observed contributions to the OPS TE than for the DIST group, as the OPS tool contains the largest number of intermediate variables. The TE of the outputs follows the same trend, which is due not only to the number of outputs contained within each tool, but also due to the

Table 5.14: Decomposition of TE by variable type

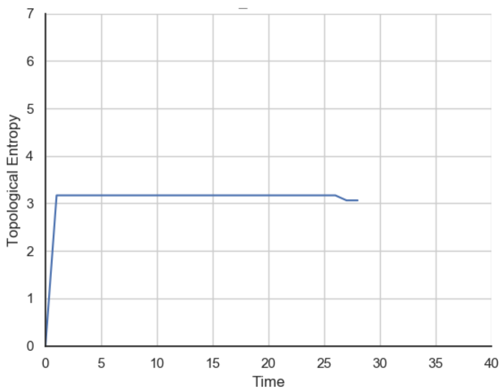
Team	Input	Intermediate	Output	Total
DIST	1.46 (47.5%)	1.10 (35.7%)	0.52 (16.8%)	3.08 (100.0%)
NAVARCH	1.54 (34.0%)	2.10 (46.4%)	0.89 (19.7%)	4.53 (100.0%)
OPS	1.52 (28.5%)	2.71 (50.7%)	1.12 (20.9%)	5.36 (100.0%)

number of connected intermediate variables. The growth in TE of the intermediate variables therefore leads to a growth in TE of the outputs.

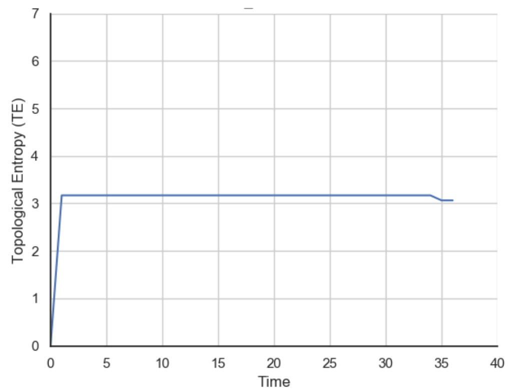
While the TE of local knowledge structures remain unchanged as the design activity progresses, the global layers exhibit changes in TE over time. The dynamics of global layer TE growth mimic those of the local knowledge structures in Section 4.2 - marked by a rapid initial growth phase which levels off over time. These two phases associated with TE growth were deemed *development* and *refinement*, respectively. The development phase is responsible for the initial structuring of the network, while the refinement phase integrates additional nodes and edges into the network. As was previously argued, these two regions play critical roles in the development and dynamics of local knowledge structures, and remains true for the development of the global structures over the course of an integration activity.

The integration process of the framework begins with the initialization of the global knowledge layer, which prescribes which knowledge entities are known and which are desired. As was previously discussed, the initial body of known and unknown knowledge is a product of the question being asked in the design activity, and provides the basis on which the design teams conduct their analyses. As such, the discussion of results in this section will first focus on the global knowledge layer. The aim of the design process simulation is to determine the relationships between the known and unknown entities - yielding a global knowledge structure. The final global knowledge structure for the simple and hard cases is presented in Figure 5.6. The TE results for the global knowledge layer are presented in Figure 5.11.

Initially, the known and unknown knowledge entities are populated as a number of unconnected nodes in the global knowledge layer. This defines the *development* phase of global knowledge growth, observed at time $0 \leq t \leq 1$. This illustrates that initial TE growth in global knowledge is a function of the original number of knowledge entities added to the layer. As these nodes are unconnected, the PageRank value of



(a) Simple Case



(b) Hard Case

Figure 5.11: Topological Entropy - Global Knowledge

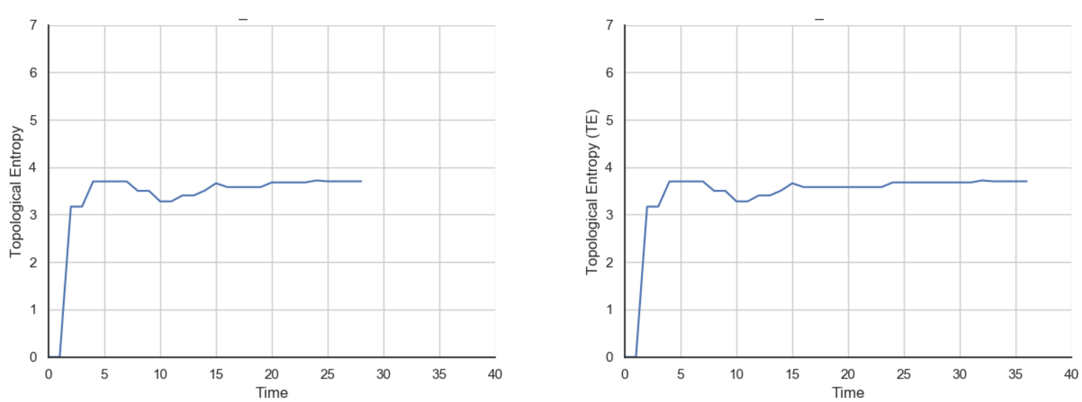
each node will be equivalent, and will be equal to $1/n$ where n is the initial number of nodes in the global knowledge layer. Applying this fact to Equation (2.7), it can be shown that the initial increase in TE of the development phase is given by $log_2(n)$. This illustrates that the more nodes are initialized in global knowledge, the more the initial increase in TE (in fact, the initial TE increases by one each time the number of initial nodes doubles). The further addition of unconnected nodes to the layer in later timesteps will result in an increase in TE, the magnitude of which will be dependent on both the number of nodes added, and the number of nodes already existing.

The *refinement* phase of the global knowledge layer ($t > 1$) characterizes the addition of nodes (knowledge entities) and/or edges (relations between knowledge entities) to the network. In general, the addition of nodes to the network will result in an increase in TE, while the addition of edges will decrease TE. The magnitude of the increase or decrease in TE will depend on the size and structure of the network when the new nodes or edges are added. The behavior of the TE metric is intuitive in relation to the dynamics of knowledge entities. The more knowledge entities are added, the larger the metric. As relations are drawn between existing knowledge entities through the addition of edges throughout the design activity, the overall TE decreases (for a fixed number of nodes the addition of edges reduces the uncertainty about the relation of nodes in the network). While the overall TE will decrease, the individual TE contribution of certain nodes will increase (dependent nodes) and others will decrease (independent nodes), due to their changes in PageRank value. These trends reveal some interesting insights into the global knowledge TE dynamics.

For both the simple and hard cases, the global knowledge layer consists of 9 nodes. Initially ($t = 0$) the global knowledge layer is empty, and at time ($t = 1$) the global knowledge layer is initialized with 9 unconnected nodes - 4 known knowledge entities (the number of each vehicle) and 5 unknown nodes (the global knowledge entities to be determined) (Table 5.3). The initial structure of the global knowledge layer is identical across the hard and simple cases, although the values contained by the known nodes differ between the two. As was previously outlined, the addition of 9 unconnected nodes corresponds to the global knowledge development phase, and leads to a large initial increase in TE - an increase of $log_2(9) \approx 3.17$ bits. This value remains constant throughout the majority of the knowledge refinement phase ($t > 1$) for both the easy and hard cases. This constant value is a result of an unchanged global knowledge structure after the initial knowledge development phase (all nodes remain unconnected), the duration of which is a result of the time required to build the global information layer from the individual disciplines. This explains the longer knowledge refinement phase of the hard case compared to the simple case - the hard case required more framework timesteps to develop the global information layer than the simple case. The final timestep ($t = 27$ and $t = 35$ for the simple and hard cases, respectively) is marked by a decrease in TE. This is a result of incorporating the projected edges into the global knowledge structure after the development of the global information structure. The final knowledge structures for the simple and hard cases are identical (Figure 5.6), which explains why the final TE value are equivalent.

Although the initial increase and final value of TE are identical across the two cases, the time required to uncover interdependencies between knowledge entities varies significantly. This is a product of additional timesteps required by the groups to fully develop the global information layer in the hard case. Only after the global information layer has been fully developed is it transmitted to global knowledge, and the more difficult problem landscape leads to additional iteration steps being required. Additional insights about the iterations can be gained through an analysis of the global information layer TE, the results of which are presented in Figure 5.12.

Having considered the topologies of the local and global knowledge layers, it is interesting to find that the fixed local knowledge structures can lead to a growing global knowledge structure. This occurs as a result of the global information layer (Figure 3.2.1) which is developed through inter-agent communication. The development region of the global information structure comes from the initial translation of the nodes in the global knowledge layer to nodes in the global information layer, marking the translation from knowledge entities to information entities for the disciplines to populate (shown in Figure 3.6). The translation of the 9 unconnected nodes in the global knowledge layer leads to an identical initial increase in TE for the global information layer as was initially observed in the global knowledge layer (3.17 bits). This increase happens one timestep after the global knowledge layer development phase ($t = 2$), and is consistent across the simple and hard cases. At timestep $t = 3$, the



(a) Simple Case (b) Hard Case

Figure 5.12: Topological Entropy - Global Information

GM_t node is selected as the first target node to be determined, and is selected by the NAVARCH team. In order for the NAVARCH team to determine a value for GM_t, their local knowledge structure requires values for the vehicle weights (W_{veh}), the height of the vehicles (z_{veh}), the volume of fuel (V_{fuel}) and height of the fuel (z_{fuel}). These variables are communicated to the global information layer to be determined by other teams at time $t = 4$ (as unconnected nodes in the information structure). The addition of these nodes corresponds to the further increase in TE observed in the time series plots.

The global information refinement phase illustrates a number of increases and decreases in TE as the teams' interactions refine the global information structure. Note that the refinement process for the simple and hard cases are identical for $t < 16$, the dynamics of which differ after this point by the number of iterations required to yield a result. The identical TE values illustrate that the initial process of global information development and refinement are identical: both cases conduct the same steps of developing the global information structure. Across both cases, early refinement steps lead to larger changes in TE than those of later steps due to changes being more influential to smaller structures than those of larger structures.

The reduction of TE observed from time $t = 7$ to $t = 10$ corresponds with the initial process of the teams determining GM_t. The maximum observed TE ($t = 4$ to $t = 8$) corresponds to the time in which all 13 information entities are unknown and are unconnected to each other (the TE value at this point is $log_2(13) \approx 3.70$). At time $t = 8$, projected edges are added between the global information entities used by the OPS team to determine values for the aforementioned nodes added to the structure by the NAVARCH team. These newly determined values are utilized by the NAVARCH team at time $t = 9$ to yield a result for GM_t, which is communicated to global information at time $t = 10$ along with the associated projected edges, further decreasing TE. The associated decrease in TE is the result of the initial projection of edges in relation to GM_t from the previously completely independent set of information entities. The relation of these previously independent information entities to a *single* target node simplifies the structure significantly, which is captured by the decrease in TE.

Starting at time $t = 12$ the TE of the global information structure is observed to increase for both cases, based on the disciplines' continued integration efforts. At time $t = 12$ and $t = 14$ projected edges are added toward the next target nodes of % and *Trim*, respectively. The additional integration efforts related to these two target nodes leads to more captured interdependencies between nodes, leading to a more complicated information structure relative to the prior timestep's. The further increase in TE observed at timesteps $t = 15$ and $t = 16$ correspond to the addition of the x_{veh} node with projected edges to the existing structure, and the connection of x_{veh} to *Trim*, respectively. This adds further complexity to the structure, and increases the observed TE.

After time $t = 16$ the integration activities conducted by both teams remain the same. However, the timesteps required to conduct these activities is longer for the 4 vehicle case than for that of the 1 vehicle case. The next target node being solved for in the global information layer is the required pump power (*required_power*), which is determined by the DIST team. In order to conduct their analysis, the DIST team requests values for z_{fuel}, z_{veh} and the required volumetric flow rate (*vol_flow_rate*), and in so doing, must transmit the latter as a new node in global information. This occurs at time $t = 20$ for the simple case, and time $t = 24$ for the hard case. After negotiating these values between the OPS and DIST teams, edges are drawn to the *required_power* node from other global information nodes at time $t = 24$ and $t = 32$ for the simple and hard cases, respectively. Finally, at time $t = 25$ (simple) and $t = 33$ (hard), the resulting global information structure is developed by drawing the final edge from *pipe_diameter* to *required_power*.

Analyzing the TE results of the global layers reveal a number of insights about the global information integration effort:

1. Changes in TE are most drastic early in the process, and exhibit the law of diminishing returns as more information entities are added to the structure. This reduction of changes in TE are indicative measures of lock-in; the sequence of operations performed on the information structure limits the ability and influence of future entities to be integrated.

2. TE decreases during the first phase of developing the knowledge structure by simplifying the previously unconnected nodes into a relatively simple structure as related to a single target node. As more interdependencies are uncovered and additional information entities are added through further interactions between disciplines, the structure becomes more complicated, and TE increases. The initial decrease in TE is based on the order of selected target nodes, and limits the future growth of the network topology. This suggests that conceptual robustness is tightly coupled to the order in which information is sought and integrated over the course of a design activity.

3. The periods of unchanging TE illustrate timesteps in which the global information structure does not change. Periods of unchanging TE could be the result of values in the existing structure becoming populated with data, or data being negotiated between disciplines. These extended periods of unchanging TE are indicative measures of refinement design churn (see Figure 1.1c), whereby additional time or calculation steps are required to conduct design analyses. Long static periods of TE are indicative of potentially inefficient design analysis types, which could be improved to improve conceptual robustness. The effort required to conduct these analyses could easily be confused by designers as a growth in information or knowledge, when in actuality this increased effort

is just processing data within an unchanging structure.

The TE metric results of this section have revealed a number of interesting insights about the development of global information and knowledge structures in a design integration activity. This metric focuses on the growth of global structures - yet this metric alone is insufficient to fully differentiate the conceptual robustness of the two cases considered. The TE metric only focuses on structural changes of information or knowledge; it does not account for what entities are becoming known, or how the values of parameters are changing over time. Considering this metric in conjunction with Data Status Entropy (DSE) and Target Value Entropy (TVE) can yield a deeper understanding of conceptual robustness.

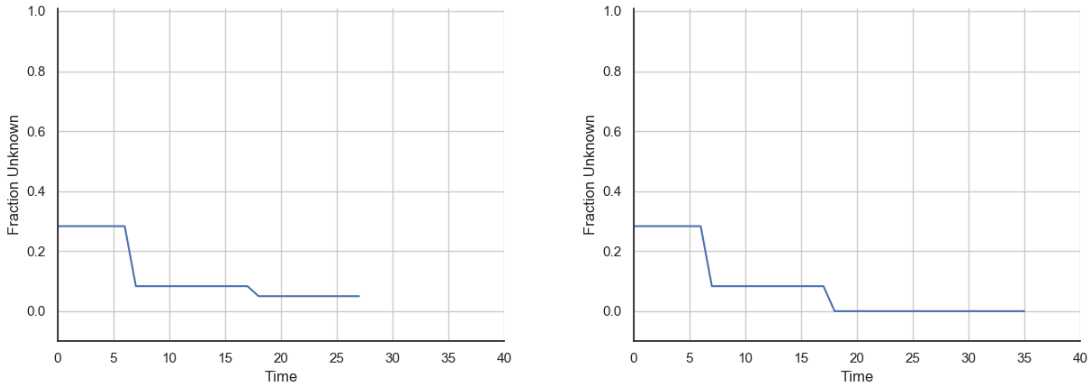
5.4.2 Data Status Entropy

An examination of the Data Status Entropy (DSE) for this case study reveals the efficacy of a design activity in relation to the proportion of the information or knowledge entities which are known or unknown. In essence, it provides insight into the growth of uncertainty of a design activity in relation to the data statuses of the nodes in the information and knowledge structures. The Normalized Data Status Entropy (NDSE) provides a similar information, but normalizes the metric in relation to the initial DSE (the DSE at time $t = 0$). Both the DSE and NDSE are tracked over time to reveal the efficacy of calculations of the hard and simple case studies, in relation to the local and global layers. The plots of DSE and NDSE are displayed in Figures 5.18 - 5.22, and Figures 5.23 - 5.27, respectively. Similar to the plots shown for TE, the figures utilize framework timesteps as a representative "time" to ease the discussion of the results. Note that the total number of framework timesteps differ between the cases, and are identical to those of the TE metrics (the simple and hard cases requiring 28 and 36 steps, respectively). The DSE and NDSE dynamics differ across the two cases, which will be discussed in more detail in the remainder of this section.

Prior to discussing the DSE and NDSE, it is useful to provide additional context about the local and global layers related to the computability of entities in the local and global structures. Insight regarding the calculability of the approach can be gained by tracking the fraction of nodes in each layer which are unknown over time. Ideally, the number of unknown entities in an information or knowledge structure should be monotonically decreasing over time, as this would suggest that calculations are being performed to determine other nodes in the structure without the addition of other unknown variables. A non-monotonically decreasing trend suggests that additional unknown entities are being added to the structure, which requires further calculations. Either an oscillating or consistently increasing trend may be indicative of design churn, in which performing a calculation leads to the creation of additional unknowns. The time series of the fraction of unknown nodes are shown for the simple and hard cases in Figures 5.13 - 5.17.

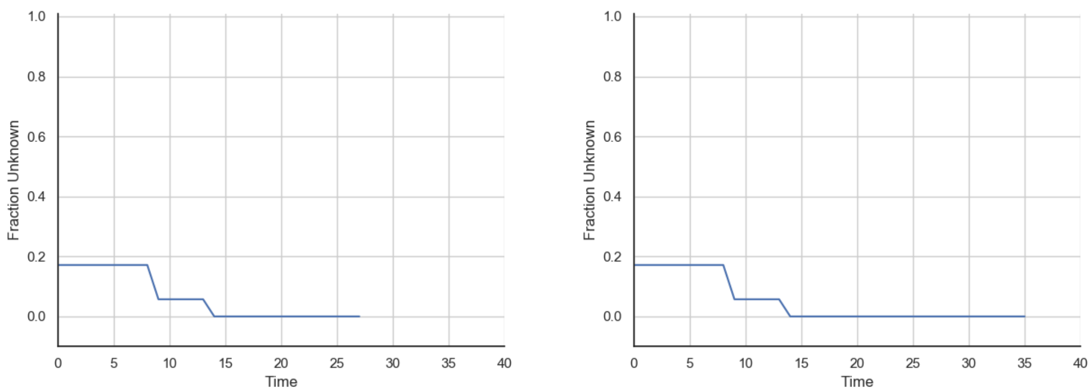
The local layers (Figures 5.13 - 5.15) all illustrate monotonically decreasing trends of the fraction of unknown nodes over time. This suggests that the questions being posed to each design team do not require any additional knowledge entities to be incorporated to be answered; the tools are sufficient to answer the question being asked of them. Note that initially the NAVARCH and OPS teams have a low fraction of unknown nodes in their knowledge structures, while the DIST group has a higher fraction of unknown entities (half of their knowledge structure is initially unknown). This indicates that the OPS and NAVARCH knowledge structures are initially more defined than that of the DIST group. This may provide insight as to which groups will be required to transmit data through global information, and which groups may be required to receive it.

The final fraction of unknown nodes also reveals important insights into the utility of the local knowledge structures in performing design calculations. As was previously



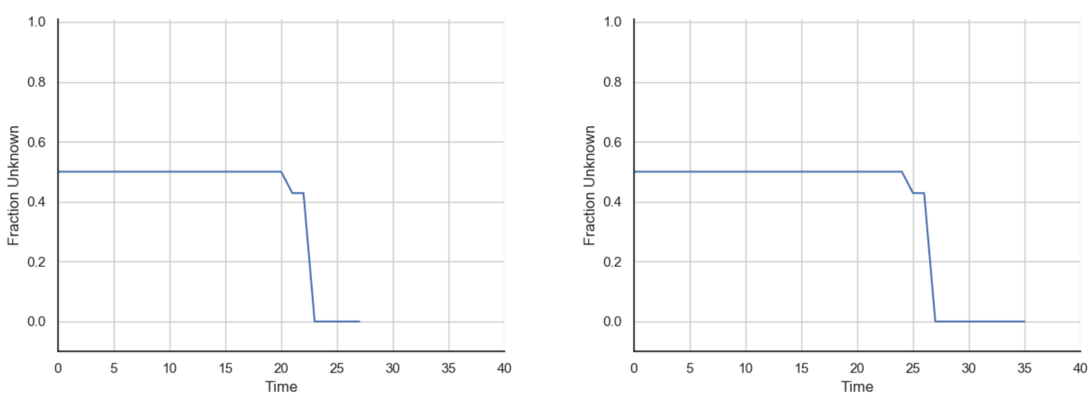
(a) Simple Case (b) Hard Case

Figure 5.13: Fraction of Unknown Nodes - OPS Team



(a) Simple Case (b) Hard Case

Figure 5.14: Fraction of Unknown Nodes - NAVARCH Team



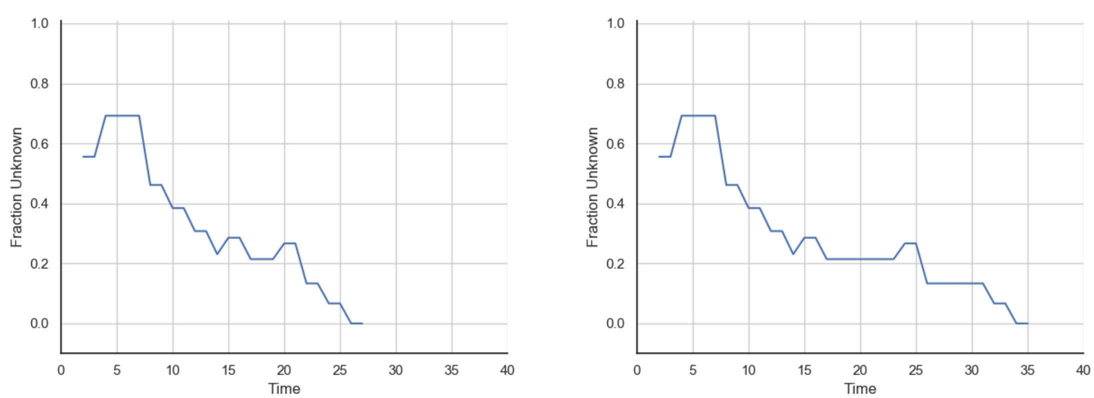
(a) Simple Case (b) Hard Case

Figure 5.15: Fraction of Unknown Nodes - DIST Team

discussed, the ideal observed trend should be a monotonically decreasing fraction of unknown nodes over time. In both the simple and hard cases this trend is observed; however, not all of the final unknown fractions are zero (a zero value indicating all nodes are known). For example, in the simple case, the OPS team contains a portion of their knowledge structure which remains unknown (Figure 5.13a). This spurs the question: how should a non-zero final unknown fraction in a local layer be interpreted? The answer is that teams must conduct calculations according to the target node selected in the global information layer. The translation of a global information target node to the local layer directs the local groups' calculations, and dictates which unknown local entities must become known. If there are still unknown entities in a local structure at the end of the design activity, it means those unknowns were not required to yield results according to the target nodes selected. This explains the final unknown fraction of local OPS nodes in the simple case: the tool was developed to accommodate four vehicles, but the simple case only considers one. This means the locations of the other vehicles are unknown, but are still accounted for in the local knowledge structure.

A high proportion of unknown entities suggest the local knowledge structure is robust to the question, as only a few entities were required to answer the question being posed. Additionally, the presence of unused knowledge entities provides the opportunity for additional information or exogenous factors to be contained in the structure: if they were to arise later in the design process, there would be no need to integrate additional knowledge entities into the local knowledge structure. In this sense, the remaining unknown variables acts as a sort of "buffer" to exogenous factors. If all nodes are known, and a result is determinable, the tool is appropriate for the posed questions but leaves no possibility of accounting for exogenous factors (there exists no "buffer"). To better understand this, consider a late stage design change which requires an additional vehicle to be accommodated by the design for both the simple and hard cases. For the simple case, the addition of an additional vehicle is easy, as the knowledge structure would not need to be changed to accommodate 2 vehicles. As a result of the additional vehicle, more of the local structure would become known, but no revision would be required to integrate additional knowledge entities. However, in the hard case the addition of another vehicle (such that the design must accommodate 5 vehicles) would require the knowledge structure to be revised to handle the change, as all of the structure is already utilized to yield a result. While there certainly exist a number of exogenous factors which may lead to required revisions to both knowledge structures (e.g. multiple refueling locations), the presence of unused knowledge provides a greater likelihood of accounting for future changes.

Unlike the local layers, the fraction of unknown nodes in the global information layer is not monotonically decreasing (Figure 5.16). The observed increase in the fraction of unknown nodes occurs when a discipline translates unknown entities into global information to be determined by other teams. For example, after the initial translation of global knowledge nodes to global information, the introduction of additional unknown entities to global information by the NAVARCH team increases the unknown fraction. The magnitude of the increase is a result of both the number



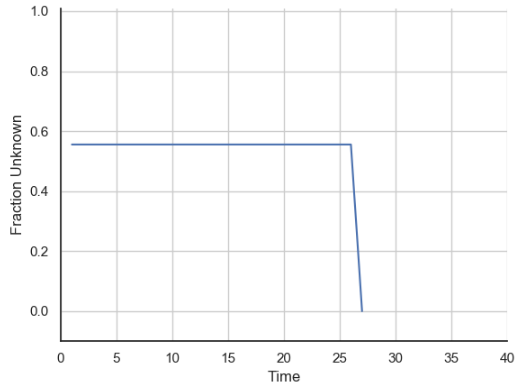
(a) Simple Case (b) Hard Case

Figure 5.16: Fraction of Unknown Nodes - Global Information

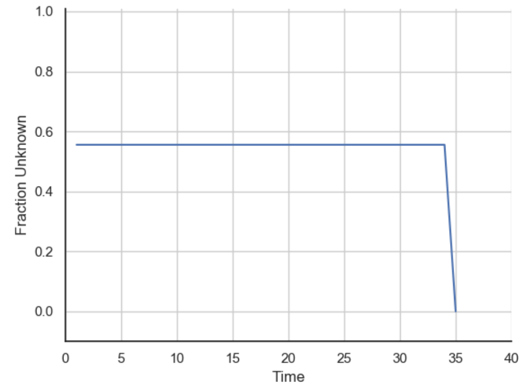
of unknowns added to global information, and the number of nodes in the global information layer. When values are determined by other teams, and the results are translated to the global information layer, these nodes become known, and the fraction of unknown nodes decreases. Both the simple and hard cases exhibit the same general trend of a progression towards zero, although the total time required to reach zero are different across the two. While the trends are the same for the two cases, the differing times are a result of the longer plateau periods in the hard case as compared to the simple case. This indicates that the hard case requires more time in these regions to yield a change in the fraction of unknown nodes than does the simple case. Note that the final unknown fraction across both cases is zero. Since the known and unknown entities from the global knowledge layer are used to populate global information, all unknowns must be determined to successfully translate the global information to global knowledge. Unlike the "buffer" provided by unknowns in the local layers, any remaining unknown global information nodes means either an answer is unable to be calculated by an individual discipline (even after revisions to local knowledge structures), or the information generated is not used to yield global knowledge (meaning the development of the global structure is not aligned with the questions posed by global knowledge). Both cases are problematic, and are indicative of emergent design failures.

The unknown variables in the global knowledge network (Figure 5.17) exhibit a monotonically decreasing trend, with a single stepped decrease to zero from the initial fraction of unknowns. Initially, the global knowledge layer is initialized 9 nodes, of which 4 are known and 5 are unknown - representing the unknown fraction of 0.56. At the final timestep, when the entities from global information are translated to global knowledge, all entities become known and the fraction decreases to 0. This suggests that all global knowledge entities have values which have been determined.

The discussion of the unknown fraction of nodes across the layers have revealed



(a) Simple Case



(b) Hard Case

Figure 5.17: Fraction of Unknown Nodes - Global Knowledge

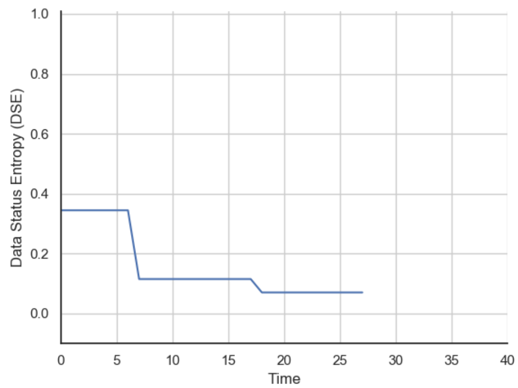
a number of insights about how much of the structures become calculable over time. From an entropy perspective, the trends observed in the DSE (Figures 5.18 - 5.22) mimic those of the unknown fraction plots.

Across all disciplines the DSE decreases in a stepwise manner as the design activity progresses. The stepwise decrease in DSE is a result of previously unknown variables becoming known within their respective knowledge structures as calculations are performed at each local layer. The magnitude of the step is a function of how many unknown variables become known as a result of the calculations being performed. Note that the magnitude of decrease in DSE does not decrease in the same manner as that of TE and illustrates that calculations which make many variables calculable late in the process do not exhibit the same process of diminishing returns. The decreases in DSE suggest that their knowledge structures are becoming more certain over time, as the DSE trends toward zero. Note that zero DSE means the knowledge structure is comprised of data statuses of all zero, or all one (in this case, all knowledge entities are one).

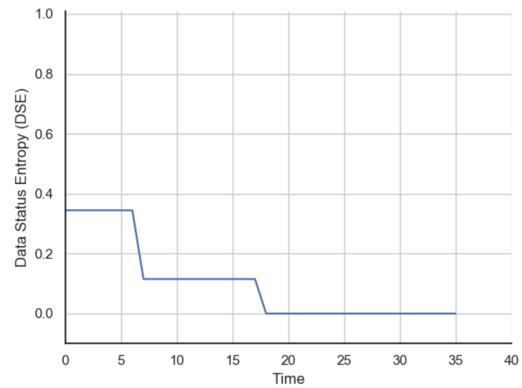
Although the local knowledge structures exhibit similar trends over time, the initial and final values of the DSE time series are significant. The initial DSE indicates the entropy of the known portion of the network. An initial DSE value of one means all nodes are unknown, while a value of zero means all nodes are known. Similar to the fraction unknown plots, non-zero DSE at the end of a successful design activity means not all of the knowledge structure was required to answer the question posed. However, if the residual DSE of a local layer leads to residuals observed in either global information or knowledge, it is indicative that the structure was insufficient to answer the question.

Similar to the local layer DSE plots, the evolution of the global information DSE (Figure 5.21) displays a general trend toward zero. This suggests that the entropy of the known portion of the network is generally decreasing over time (becoming more certain). However, unlike the local layers, the function is not monotonically decreasing. The initial translation of nodes to the global information layer defines the intial DSE which reflects the entropy of the intial known portion translated from global knowledge. Note that the initial portion of the plot remains flat. This is due to the uncertainty associated with the known node remaining constant, while new unknown nodes are added to the structure, as evidenced by Figure 5.16. The observed later-stage increases in DSE correspond to new known parameters being added to global information, which increases entropy. As unknown nodes become known, the DSE decreases. The final DSE being equal to zero indicates that all nodes in the network have become known - illustrating the successful completion of the global information layer from a calculability perspective (no global information nodes remain unknown).

The DSE time series is presented for the global knowledge layer in Figure 5.22. Note the initial DSE, which remains constant throughout the majority of the process for both the simple and hard cases, is identical to the starting value in global information (Figure 5.21). This is because the same initial structure is translated from global knowledge to global information. However, unlike the global information DSE plot, the DSE of the global knowledge layer is marked by a rapid decrease to zero at the end of the process and exhibits no intermediate fluctuations. The final

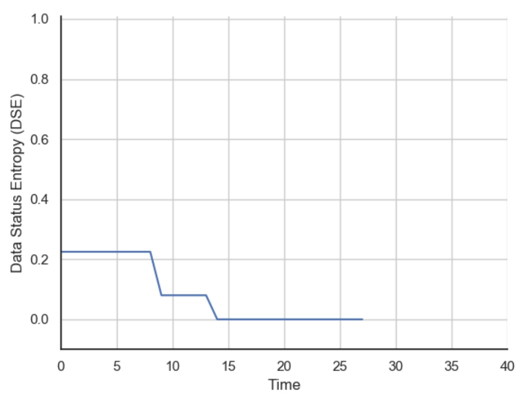


(a) Simple Case

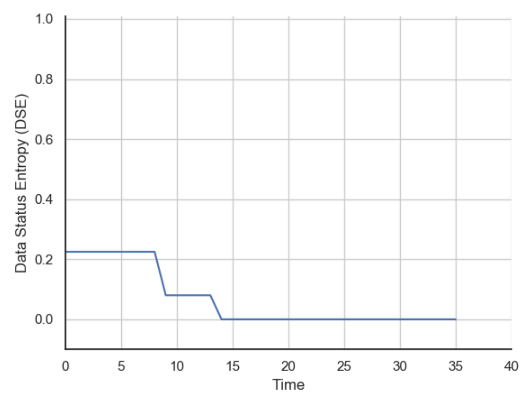


(b) Hard Case

Figure 5.18: Data Status Entropy - OPS Team

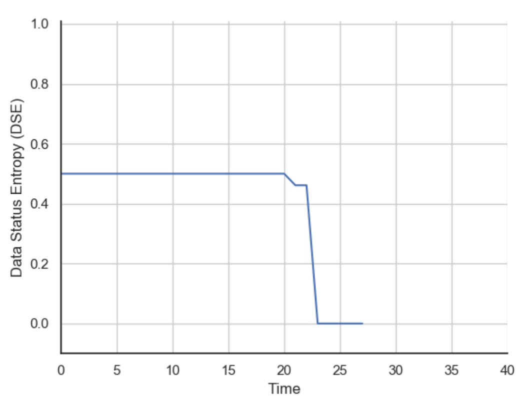


(a) Simple Case

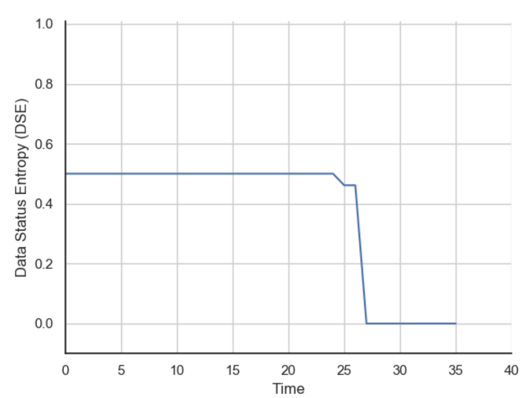


(b) Hard Case

Figure 5.19: Data Status Entropy - NAVARCH Team

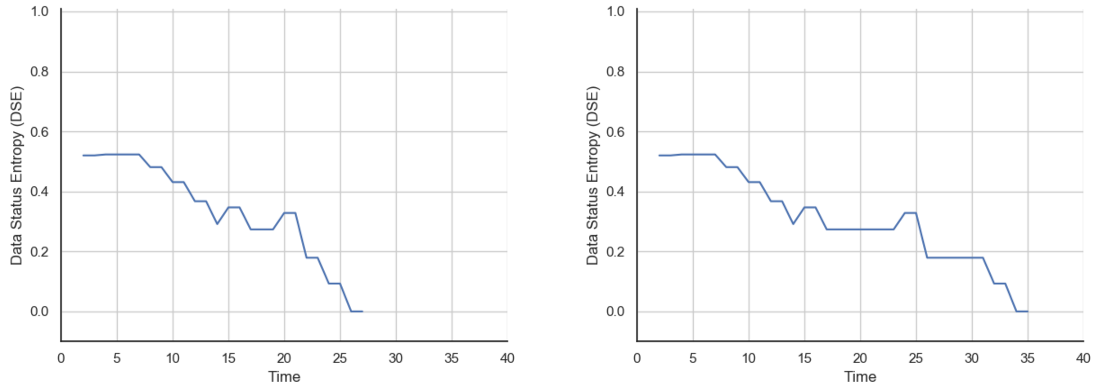


(a) Simple Case



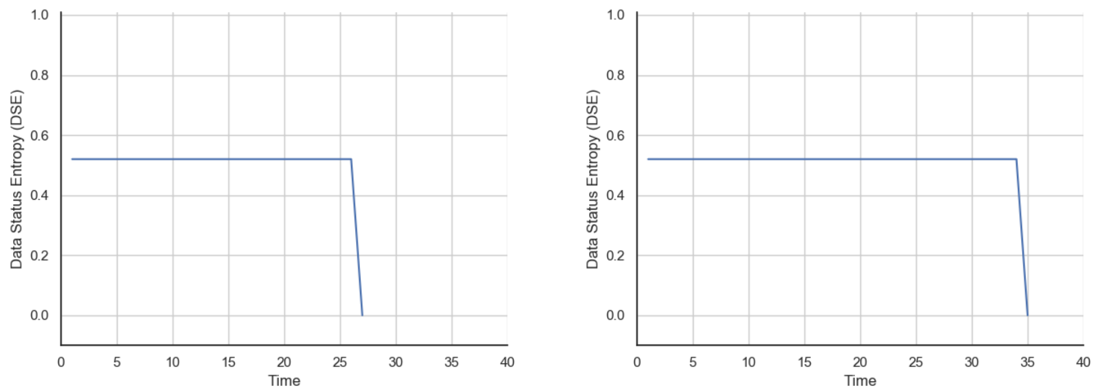
(b) Hard Case

Figure 5.20: Data Status Entropy - DIST Team



(a) Simple Case (b) Hard Case

Figure 5.21: Data Status Entropy - Global Information

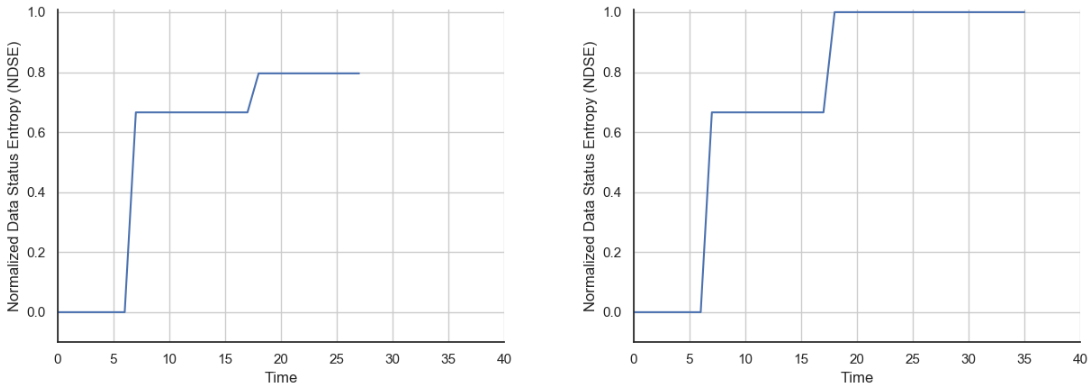


(a) Simple Case (b) Hard Case

Figure 5.22: Data Status Entropy - Global Knowledge

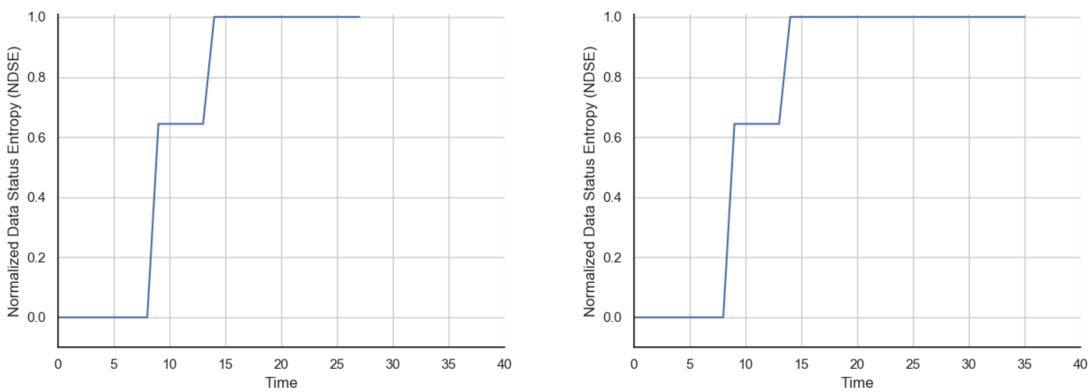
value of zero indicates that all global knowledge entities are known, and the rapid late stage decrease means they only become known at the end of the process in a single step. Thus, the effective development of the global information layer resulted in a calculable knowledge structure for both cases, but not until late in the process. Creating this calculabilty earlier in the process could be attained through a more rapid development of global information.

The unknown fraction and DSE plots have revealed absolute measures of each network's calculability as the design progresses, but there is added value in understanding how the generation of information and knowledge evolves relative to the initial state. To better understand this relationship, the NDSE metric is applied to both the local and global layers for the simple and hard cases.



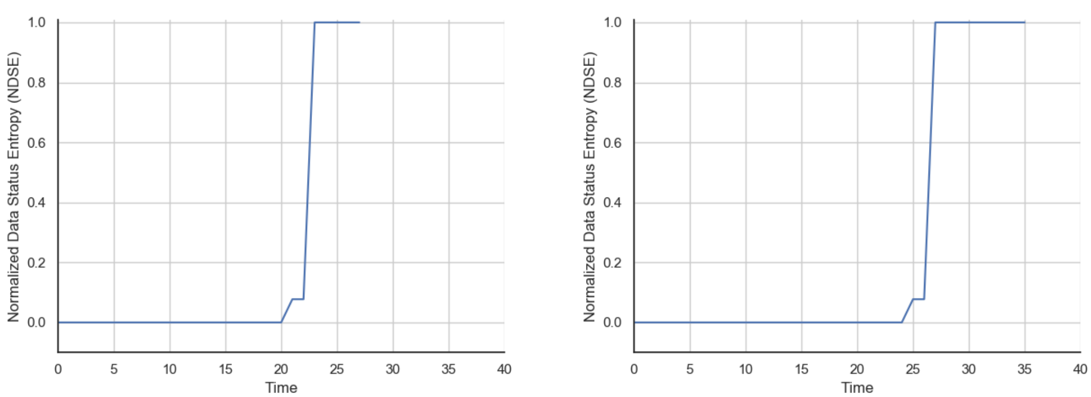
(a) Simple Case (b) Hard Case

Figure 5.23: Normalized Data Status Entropy - OPS Team



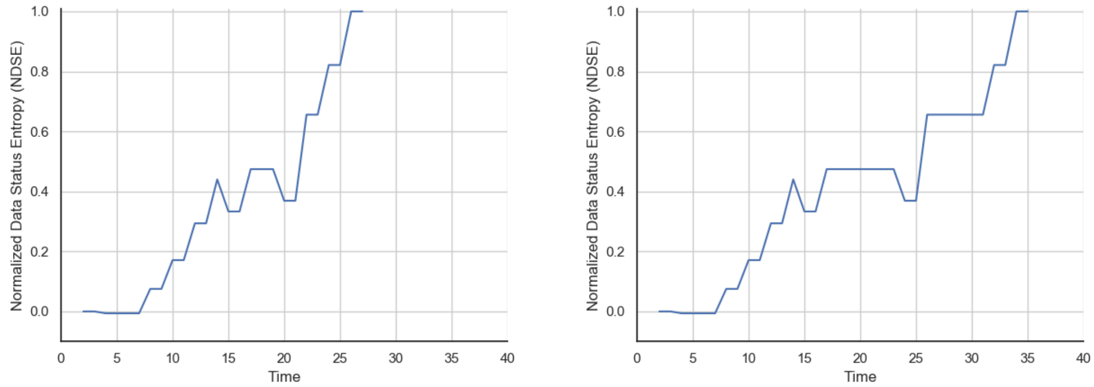
(a) Simple Case (b) Hard Case

Figure 5.24: Normalized Data Status Entropy - NAVARCH Team



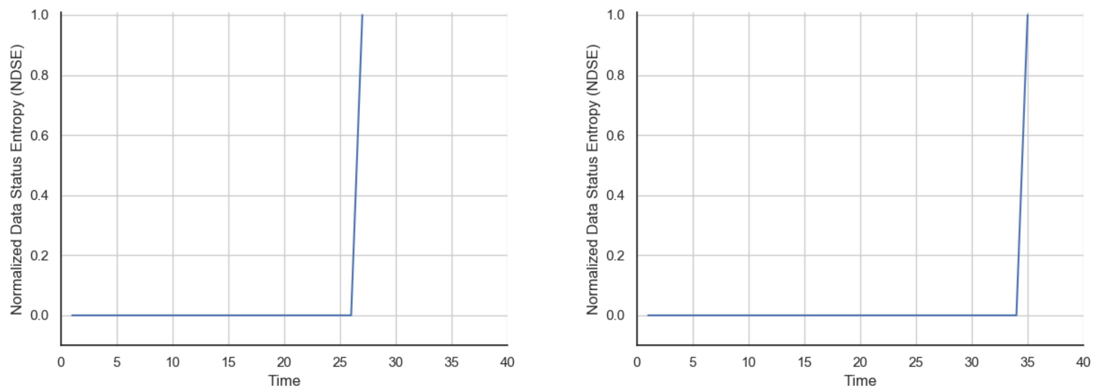
(a) Simple Case (b) Hard Case

Figure 5.25: Normalized Data Status Entropy - DIST Team



(a) Simple Case (b) Hard Case

Figure 5.26: Normalized Data Status Entropy - Global Information



(a) Simple Case (b) Hard Case

Figure 5.27: Normalized Data Status Entropy - Global Knowledge

Tracking the unknown node fraction, DSE, and NDSE over time provides a number of takeaways related to the calculability of the local and global information and knowledge structures over the course of the global integration effort. These relate to conceptual robustness in the following ways:

1. At local levels, oscillations in the unknown fraction, DSE, and NDSE are indicators of design churn in *knowledge development*. If a decrease in any of the metrics is subsequently followed by an increase, this suggests that a calculation was performed that spurred additional unknown entities which themselves need to be determined. These indicators provide evidence of a knowledge structure which is not suited to answer the question being asked of the discipline, and

hence is not conceptually robust by quantifying the increase in unknowns in the structure.

2. Long periods in which the metrics are unchanging are indicators of design churn in *refinement*. These regions highlight areas in which no more of the network is becoming calculable, rather the values themselves are being calculated or negotiated between teams. These regions illustrate areas in which there are changes to data, but there are no changes to the development of calculable entities. Extended periods in which these metrics remain static could provide means to highlight inefficiencies in both calculations and communication pathways, and could hence be used to reduce the time required between design iterations.

3. At the end of a design activity, residual unknown nodes in local knowledge structures suggest that not all of the structure is required to calculate values for the development of the global information. This suggests that the question posed to the disciplines are easily accounted for by the local knowledge structure. These remaining unknown nodes increase the likelihood that exogenous factors will be accounted for by the structure without the need for local knowledge structure revisions. Not requiring revisions to the knowledge structure have been shown to assist in faster rework times (see Chapter IV). While this may suggest that larger and more complex tools should be leveraged to better account for these exogenous factors, this may come at the expense of the utility and applicability of the tool in early stage design. Thus, there exists a trade off between the conceptual robustness and complexity of local knowledge structures, which requires further exploration.

4. Residual unknown nodes in the global information or global knowledge layers could be used as indicators to predict integration failures. Nodes remaining unknown in the global information layer means that no team was able to calculate

a result for that entity, and thus that entity will not be connected with projected edges to the rest of the information structure (see Section 3.2.1). This represents an inability to integrate that information entity with the remainder of the information structure. This could be the result of the sequence in which calculations are performed in the local layers, or a fundamental inability for the teams to conduct calculations. This unknown global information node could thus translate to an inability to integrate an unknown global knowledge entity with the remainder of the global knowledge structure. If the residual unknown global information node was one of the initial global knowledge nodes, this would represent a global knowledge entity that would remain unknown. Additionally, it could lead to an inability to create projections in the global knowledge layer, leading to incorrect or non-robust interdependencies between global knowledge entities.

Utilizing the case study, the entropy metrics presented in this chapter have revealed a number of insights into the dynamics of how unknown entities in information and knowledge structures become known over time. The metrics can be leveraged to understand types of activities which lead to both developmental and refinement design churn, and the impact of conducting rework in terms of the impact on unknown entities within a structure. These metrics focus on the progression of the calculability of structures over time from a binary perspective (either nodes contain data or not), and does not reveal any information about the values associated with the data themselves. Thus, an additional metric, Target Value Entropy, is required to fully understand the change in uncertainties of the values associated with known nodes over time. This additional metric can be used to reveal additional indicators of conceptual robustness.

5.4.3 Target Value Entropy

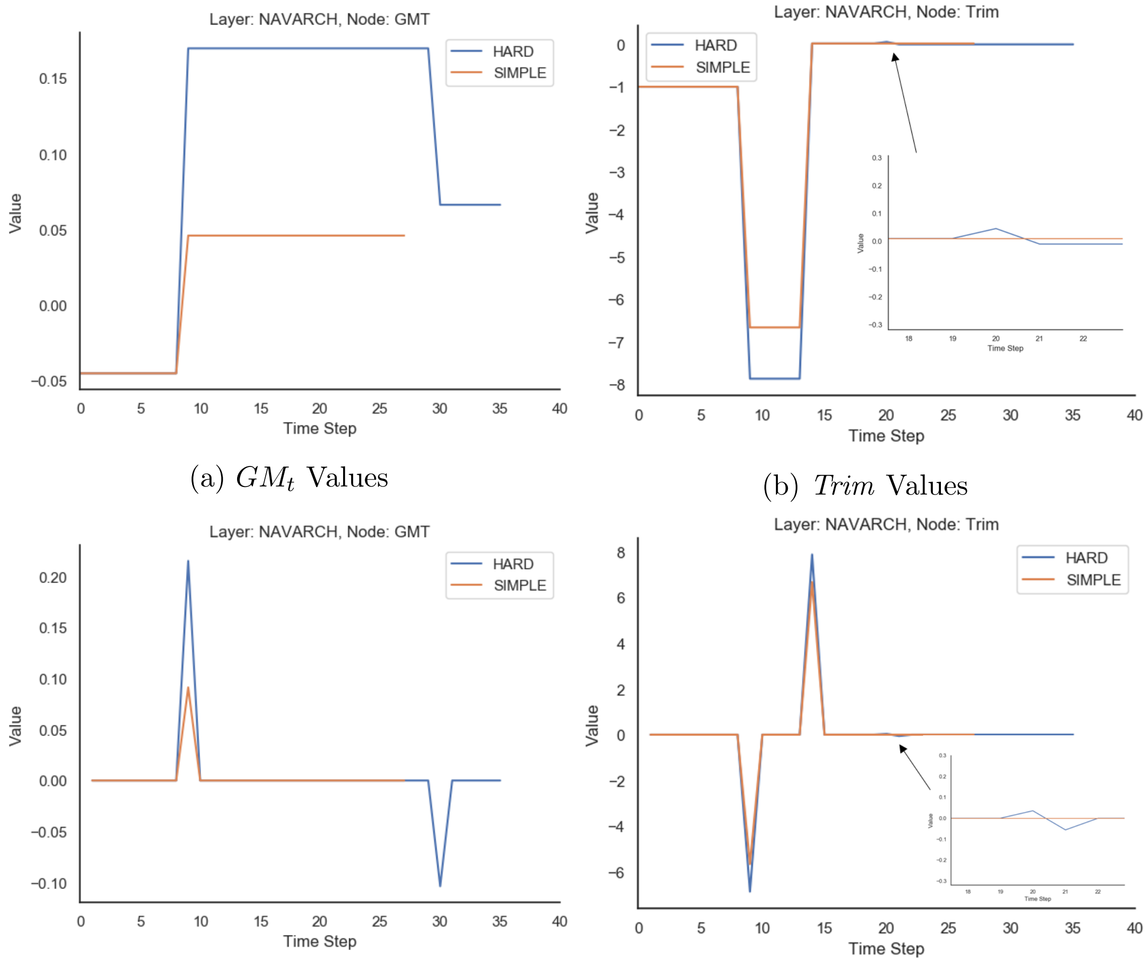
Unlike the aforementioned entropy metrics, Target Value Entropy (TVE) is focused on individual nodes rather than on the entire layer. This presents a different perspective into conceptual robustness by highlighting changes *within* the network rather than *on* the network. By focusing on individual nodes, TVE highlights changes in uncertainties of the history of values within a node over the course of a design activity. Additionally, TVE is used to track intermediate calculations conducted by the optimizer to yield results as it progresses. This enables the ability to track both the changes in calculated values and the process by which these values were calculated. Both final and intermediate calculated values enable not just the changes in uncertainty of input variables to be examined, but also how these changes translate to intermediate and output variables over time in both local and global nodes. Analyzing the changes to dependent nodes allows the unpredictable nature of value changes to be understood, and applied to the context of a greater design goal. The first part of this section focuses on the change in TVE of the calculated values, while the latter part considers the intermediate calculations used to determine the final values.

Calculated Values

The TVE is tracked for each node in the K-I framework over time. For local knowledge layers, node values either represent input values to a calculation or reflect output values as the result of conducting the calculation. For the global information layer, the time history of node values represent the data values communicated between teams over time. The global knowledge structure node values represent the final determined values of the known entities at the conclusion of the design activity. Hence, analyzing the TVE and DTVE trends of input and output values provides a different understanding of the design activity dependent upon the layer being analyzed. Since nodes in the framework are created at different times as the design progresses, some nodes have long time series while others may only contain a single value. Many of the nodes across layers do not exhibit value changes over time, and thus experience no change in TVE. Note that no TVE time series exists for nodes which remain unknown or uncalculated. Similar to the previous metrics, all figures displaying TVE do so as a function of the number of framework timesteps. This section will present a number of examples of what conclusions can be drawn about a design activity using the previously defined simple and hard cases, and validate the developed network evaluation metrics.

To understand the insights that are provided by the value-centric measures of entropy (TVE and DTVE), consider the portion of the design activity related to calculating GM_t and *Trim*. In both the simple and hard cases, determining GM_t marks the first step in the sequence of solving for the unknowns, and solving for *Trim* happens third. For context, both nodes exist in the NAVARCH team's predetermined local knowledge structure, and in the global knowledge layer from time $t = 0$. The global knowledge nodes are communicated to global information at time $t = 1$, which creates the associated GM_t and *Trim* nodes in the global information layer. Both nodes in the global information layer contain no data until calculations are performed and communicated by the NAVARCH team's local knowledge structure. In the global knowledge layer, both nodes contain the final values determined in the design activity, and become populated once the global information structure has been developed. This discussion will consider how the values of GM_t and *Trim* change over time in each layer, and as a result, will provide conclusions that can be drawn about the simple and hard cases.

Before discussing TVE and DTVE, this discussion first focuses on the time histories of GM_t and *Trim* values in local layers. This provides background into how variables have changed over time as a result of the local calculations performed. Figure 5.28 displays the value time series and value differential time series for the NAVARCH,



(a) GM_t Values

(b) *Trim* Values

(c) GM_t Value Differentials

(d) *Trim* Value Differentials

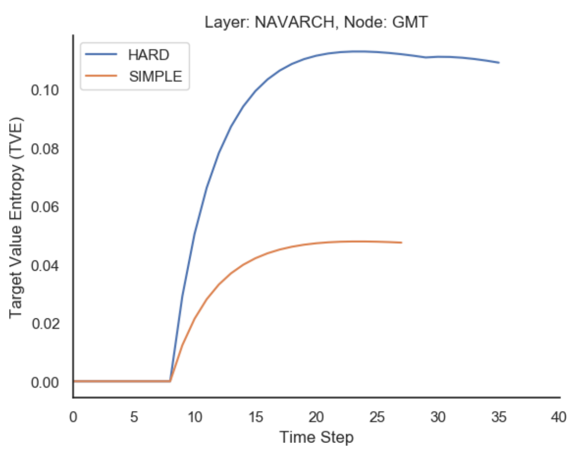
Figure 5.28: Value Time Series and Value Differential Time Series for NAVARCH layer GM_t and *Trim* nodes.

GM_t, and *Trim* nodes. Initially, a portion of the NAVARCH local knowledge structure is uncalculated (see the discussion in Section 5.4.2); however, values are still calculable for both GM_t and *Trim*. Although these values are present initially, they are inaccurate, as not all input parameters have been adequately determined. The initial values are identical for both cases. As supporting node values are determined and integrated into the local knowledge structure, the accuracy of these values increases.

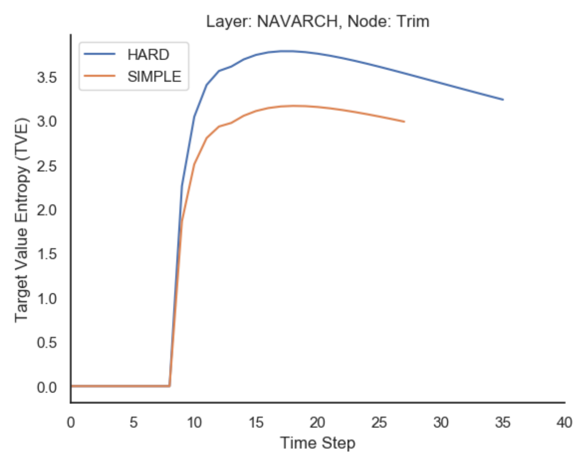
In Figure 5.28a, the change in GM_t value observed at time $t = 9$ across both cases is the result of supporting values being communicated into the local knowledge structure. Specifically, the supporting integrated values are related to the z-location of the vehicles (z_{veh}), the weight of the vehicles (W_{veh}), and the volume of fuel required (V_{fuel}). Given the NAVARCH team's local knowledge structure (Figure 5.2), these are the only unknowns required to accurately determine GM_t, and thus the integration of these values leads to a new calculated value for GM_t in both cases. Note that the calculated GM_t is larger for the hard case than for the simple case, as a result of the larger required volume of fuel being placed low in the vessel. Note that in both cases, the integration of supporting entities related to GM_t at time $t = 9$ also had the adverse affect of changing the value of *Trim* (Figure 5.28b). This large negative change arises from the integration of supporting GM_t nodes, while the supporting *Trim* nodes have not yet been determined. While the hard case exhibits the benefit of a larger calculated GM_t, its impact on *Trim* is more deleterious than that in the simple case.

For the simple case, the initial calculations of GM_t and *Trim* remain unchanged throughout the rest of the design activity, while the hard case exhibits an additional value change for both variables. For the simple case, the initial determination of both node values leads to feasible solutions for other local knowledge layers, and do not require changes to their values. Conversely, the initial values determined by the hard case lead to infeasible options by other groups, and thus are revised through further iteration steps. Note that the hard case's late change in GM_t occurs at timesteps after the simple case has concluded the integration process. The inset figures in Figures 5.28b and 5.28d illustrate a small, late-stage value change in *Trim* as a result of revised calculations of x-locations (x_veh) from the OPS team. While this change is small, it is important given its relation to the global information layer, which will be described later.

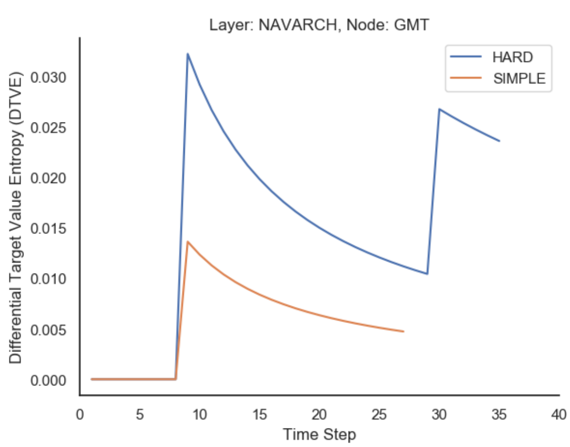
The value time series reveal the history of values a variable took over the course of the design. The differential value time series illustrates the magnitude and directionality of change of a variable's values over time. While both time series provide useful information to designers, considering the associated trends of TVE and DTVE provides insight into uncertainties associated with the variables. Applying TVE to the value time series illustrates the growth of uncertainty given the entire history of values a variable took. Applying DTVE to the differential value time series illustrates the growth of uncertainty based on how a variable changed. The associated TVE and DTVE plots for the NAVARCH GM_t and *Trim* nodes are presented in Figure 5.29, for both the simple and hard cases.



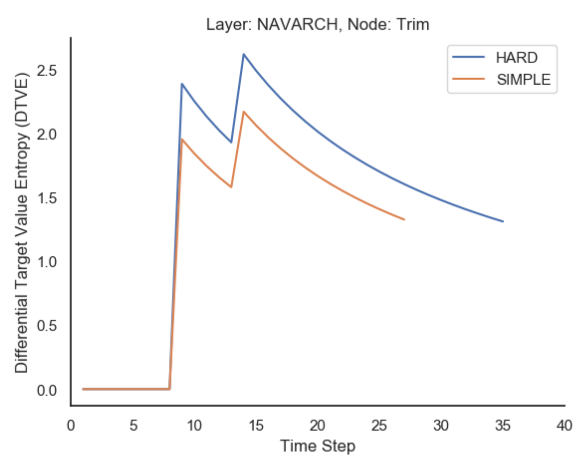
(a) GM_t TVE



(b) *Trim* TVE



(c) GM_t DTVE



(d) *Trim* DTVE

Figure 5.29: NAVARCH *Trim* Time Series

The TVE plots depicted in Figures 5.29a & 5.29b illustrate how the uncertainty related to the history of variable values changed over time. These plots are based on the value time series shown in Figures 5.28a & 5.28b. In both cases the initial TVE remains zero up to time $t = 8$, since the variables value remained constant. Although this value was inaccurate, and based on a knowledge structure which was not completely calculable (as was previously discussed), the uncertainty of the distribution of values is zero. Upon adding the supporting information to calculate GM_t at time $t = 9$, the distribution of observed GM_t values widens, which is accompanied by a growth in entropy. The accompanying change in *Trim* as a result of the information integration results in a growth in *Trim* TVE. The magnitude of entropic growth is a function of the magnitude of the new value, which explains why the hard case exhibits more TVE growth than that of the simple case across both nodes. As a value exists for a longer time, it represents a higher number of observations in the time series. As a result, the distribution of observed values becomes more certain, which leads to a reduction in TVE. However, as the length of the time series increases, introducing a new value decreases the impact on entropy growth. In order for TVE to decrease significantly, the number of observations of the new value (how long it has existed in the time series) must become significantly larger than those of any other value. Hence, this metric provides a holistic view of the values' uncertainty over the entire history, but becomes less sensitive to small value changes as time increases. In accounting for this issue, DTVE is a more useful metric.

The DTVE plots for the NAVARCH GM_t and *Trim* nodes are presented in Figures 5.29c & 5.29d, and are based on the differential value plots shown in Figures 5.28c & 5.28d, respectively. The trends in DTVE are useful measures of understanding the uncertainty related to the change in a variable's value, as the plots demonstrate the added sensitivity to the addition of new values independent of the time they are added. The timesteps which exhibit a value change are accompanied by a rapid growth in DTVE, and the magnitude of the value change is reflected by the magnitude of DTVE increase. Similar to the plots of TVE, the growth in DTVE is larger for the hard case than for the simple case, as the spread of value changes is larger in the former than in the latter. The periods of unchanging values are illustrated by a rapid decrease in DTVE. Note that the small late stage changes in *Trim* (at time $t = 20$ and $t = 21$) still lead to a decrease in both TVE and DTVE entropy. The magnitude of this change is on the order of 10^{-2}, which is very small in comparison to the previously observed changes. This suggests that the small value change still increases the certainty of the value relative to uncertainty created by the previous values.

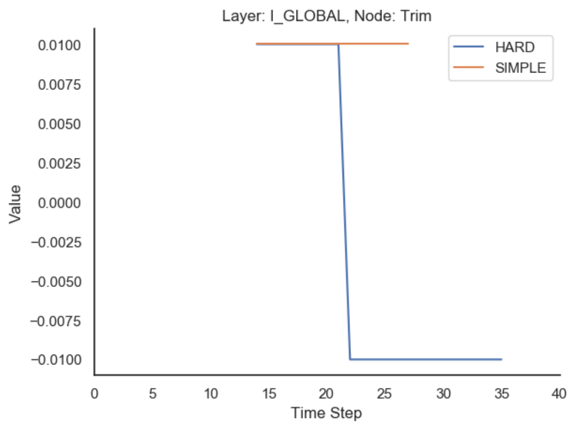
The behavior of the DTVE plots is a result of the way in which the differential time series are composed. Periods of unchanging values are represented by zeros in the differential time series. This centers the distribution of differential values around zero, and any changes will lead to an increase in DTVE. It should be noted, however, that this metric may become less sensitive to identifying increases in uncertainty if changes of the same magnitude are observed at the same frequency as those of unchanging periods. This would create a bimodal distribution of observed differential values, which would lead to fewer increases in DTVE at later timesteps. While this would decrease the efficacy of this metric, the phenomenon would be accounted for in the TVE plot by illustrating a consistent growth. The phenomenon would also be observable by viewing the value time series and differential value time series.

While the trends illustrated in the TVE and DTVE plots provide a means for comparison across the simple and hard cases, the scaling of the entropy growth merits further discussion. The magnitude of TVE and DTVE is dependent on the observed variable value (or change in value), the number of times it is observed, and the sequence in which it is observed. The resulting entropy scale will be dependent on the scale of the variable being considered. Thus, the changes in TVE and DTVE will be unit-dependent. For example, a much larger change in TVE and DTVE will be observed in a variable changing from 100 to 90 than in a variable changing from 1 to 0.9. Although both of these changes represent the same percentage change, the resultant changes in entropy will reflect the scale of the units used. This information is useful to designers by framing the uncertainty values in the context of the units of the variable in question, but limits relative comparisons across variables of different units.

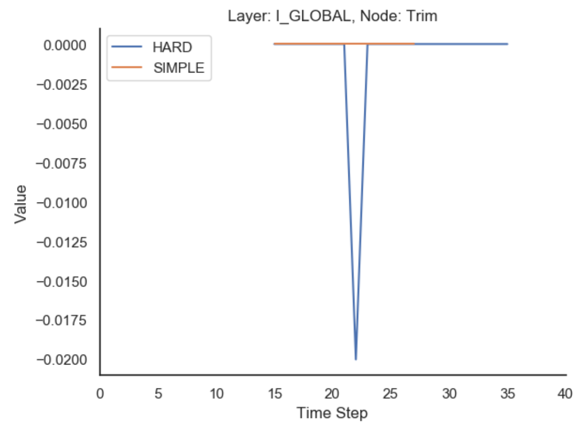
While the magnitudes of entropic growth are indeed useful to designers, it would be convenient to create a normalized version of the metrics to understand uncertainty change across variables independent of the scales of their values. Many measures of entropy are maximized by the uniform distribution - when all outcomes are equally likely. This provides a convenient method to normalize these entropies, and enables relative changes in entropy to be compared across variables. However this is not the case for Cumulative Residual Entropy (CRE), on which TVE and DTVE are based (see Section 2.2.2). Given that the variable values in consideration are often unbounded, the uniform distribution is not finite and does not necessarily represent maximum uncertainty. As a result, the TVE and DTVE are unbounded from above, as both an increase or decrease in value will lead to the same change in uncertainty (both metrics' lower bounds are zero). To date, no generalized maximum entropy metric has been developed for CRE, and thus there are currently no meaningful ways of normalization in the same way as traditional metrics. The development of this normalized metric would be advantageous to understand relative changes in uncertainties across variables. For the time being, understanding entropy metrics in the context of the variable value itself will have to suffice.

As was previously mentioned in the discussion about Figure 5.28, the *Trim* node exhibits two late-stage value changes at times $t = 20$ and $t = 21$ in the hard case. The small magnitude of these changes relative to the previously observed values leads to a decrease in uncertainty in both the TVE and DTVE metrics at a local level. In this case, the late stage *Trim* changes occur as a result of added iteration steps relative to the negotiation of x-location of the vehicles (x_{veh}) between the OPS and NAVARCH teams. When the NAVARCH team calculates a feasible value for *Trim*, they do so by determining a combination of x_{veh} and x-location of the fuel tanks (x_{fuel}). The x_{veh} value is communicated to the OPS team through the global information layer for them to determine a feasible arrangement of individual aircraft locations such that the net center of gravity matches the communicated x_{veh} value. In the simple case, the OPS' task is easy - they simply locate the single aircraft at the center requested by the NAVARCH team. However, the added difficulties of arranging multiple aircraft with spacing constraints to accommodate the requested x_{veh} value emerges in the hard case. If the OPS team is able to find a feasible arrangement of aircraft which meets the requested value, then the x_{veh} and *Trim* values remain unchanged in global information and the NAVARCH team structures. However, if no such arrangement is found, the OPS team determines an arrangement which is as close to the requested value as possible, and communicates the new x_{veh} value back to the NAVARCH team through global information. Upon the integration of the revised x_{veh} value into the NAVARCH structure, the NAVARCH team is required to determine a new value of x_{fuel} to calculate a new feasible *Trim* value. This explains the observed late stage NAVARCH *Trim* changes. The increase in *Trim* value observed at time $t = 20$ is the result of integrating the new x_{veh} value from the OPS team into their calculation, and the decrease in value at time $t = 21$ is the result of determining a new value for x_{fuel} to correct it.

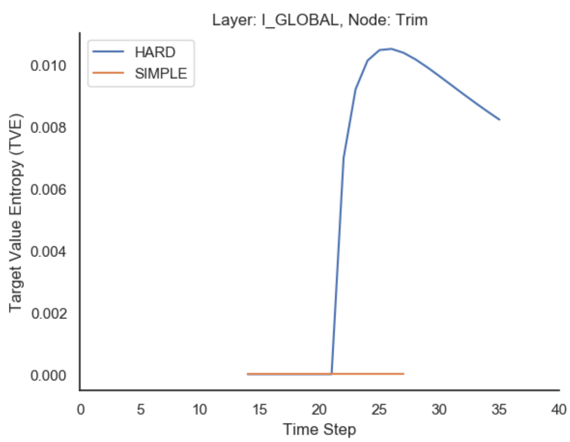
While the entropy plots of the local layer uncover the uncertainty of input and output values within a local knowledge structure, applying TVE and DTVE to the global information layer provides insight into the uncertainty of negotiated values between teams. The TVE and DTVE plots of the aforementioned scenario are presented in Figure 5.30, along with the corresponding global information value and differen-



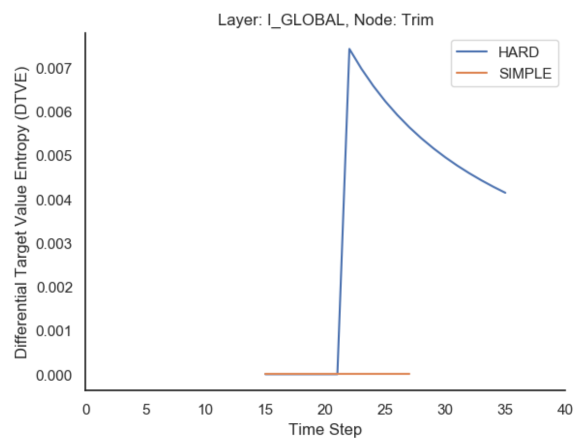
(a) Calculated Values



(b) Calculated Value Differentials



(c) TVE



(d) DTVE

Figure 5.30: Global Information *Trim* Time Series

tial value time series. Similar to the previous plots, the values associated with the nodes are presented for all timesteps since its creation. The initial time series values correspond to the initial communication of the *Trim* value with the NAVARCH team's requested values. The plots initially exhibit zero entropy, as the time series are comprised of the singular value. The plots illustrate a growth in entropy at time $t = 22$, at the conclusion of the process described above. This change is a result of the NAVARCH team conducting their recalculations and communicating the new *Trim* value to global information after integrating the OPS team's new results. Note that the estimate of *Trim* is only changed in global information when it is communicated by the NAVARCH team, and not during the negotiations of the x_{veh} variable.

Analysis of these plots provides a higher-level perspective of the uncertainty growth in global information values as a result of communication between teams. A growth in TVE or DTVE highlights that a global information entity has changed value, and that the distribution of all observed values has become less certain. Similar to the other plots, as a value represents a longer portion of the time series, the distribution of values increases in certainty, and entropy decreases. The extent of entropy growth is a product of the magnitude of the change in value. A small change in communicated value leads to small increases in entropy, while large changes means communicated values are exhibiting large value changes. Thus, large changes in entropy indicate significant value changes in global information values, and could be the result of differing design objectives between teams or misaligned local knowledge structures. In any case, this enables designers to identify potentially problematic design variables in the context of integrating disparate analyses between teams. Global information structures which exhibit zero or small amounts of entropy growth are indicative of a simpler design integration process, as it indicates more agreement about calculated values between teams. This suggests a more conceptually robust process of integrating disparate sources of knowledge toward building a global information structure, with which to create a global knowledge structure.

Based on the results presented in this section, the value-oriented entropy metrics reveal several factors about the robustness of a design approach in relation to the input and output values within and between teams. This provides additional perspectives into the conceptual robustness of integrating disparate sources of knowledge in an integration activity:

1. The value time series of local knowledge entities provide designers with a representation of how variable values evolve over time. The value time series of input values provide designers with a history of decisions made as inputs to a calculation. Tracking the time series of output values enables an increased

understanding of how changes in inputs yield different results, and more importantly provides a means to identify changes in output parameters which would not necessarily be expected. This leads to an increased transparency of inputs and outputs of a given local knowledge structure.

2. The differential value time series provide designers more clarity on the magnitude of changes of local knowledge structure values over time. This provides similar insights as those of the value time series, with the added benefit of explicitly measuring the magnitude and directionality of variable changes over time.

3. Application of TVE to nodes in the framework quantifies the amount of uncertainty in a node's value up to a point in time. Since the growth of TVE is dependent on (1) the variable values, (2) how long they have existed, and (3) the sequence in which they were observed, this metric enables designers to better understand the growth of uncertainty of values as it relates to both the sequence of calculations and the calculation results themselves. At local levels, TVE highlights the uncertainty in the distribution of observed input values or calculated output values. At global levels, TVE highlights the growth of uncertainty of values communicated between teams. This provides the ability to separate uncertainties in local calculation procedures from values communicated between teams for integration purposes. Large growths in TVE at local levels are the result of calculations yielding wildly different results, and thus the metric can be used to identify potential issues in local calculation procedures. Since the TVE metric is applied to all nodes in the local knowledge structure, the integration of a new value (or series of new values) into the local knowledge structure can be used to study the increase in uncertainty associated with all other nodes. Thus, it can also be used to highlight the sensitivity of a local

dependent node's uncertainty based on a change in values of other nodes in the local layer. This is useful in identifying incompatible local knowledge structures or differing local objectives.

4. DTVE highlights the uncertainty related to the magnitude of change of values up to a point in the design process. This metric contains many of the same advantages as TVE related to understanding uncertainty propagation. Since the metric considers the distribution of observed changes in node value, it provides a different context of quantifying uncertainty. Applying the metric at the local and global levels provides different lenses with which to understand the progression of a design activity. Variables with large DTVE growths suggest they are exhibiting unusual value changes, which if unexpected, can be used to immediately identify designers as to the impact of variable value changes elsewhere in the knowledge structure. This mitigates the likelihood of conducting rework and limits future integration failures.

5. Understanding uncertainty propagation through local network layers using TVE or DTVE can be used to help designers understand the interdependencies of inputs and outputs in their tools. From a conceptual robustness perspective, these metrics could be used to flag designers when an unexpected growth in uncertainty occurs in an unforeseen output variable - limiting the probability of future integration failures. The metric can also be used to curtail local rework activities by addressing potential value-centric issues the instant they arise, rather than later in the design process. At global layers, TVE and DTVE can be used to identify potentially highly variable and frequently changing nodes. The identification of these nodes provides extra insight into potentially problematic integration variables earlier on in the design process. Emergent design failures can be mitigated by restructuring communicated variables between nodes (and

hence local knowledge structures) to improve inter-team communication.

This section has thus far been focused on understanding the propagation of value uncertainty and its contribution to understanding conceptual robustness, using TVE and DTVE by considering the inputs and outputs of calculations. While the benefits of this perspective have been outlined, an additional layer of understanding can be gained from delving a layer deeper into the *intermediate* calculations used to relate the inputs to the outputs. The next section will present the additional insights revealed by analyzing the intermediate values generated in yielding a calculation result.

Intermediate Values

The previous section focused on analyzing how node values in both information and global layers change over time. In local layers, node values reflect either inputs to, or results from, a team's calculation. Node values in the global information layer reflect values of any local knowledge entity communicated between teams. While considering the way these values change over time provides a number of insights into conceptual robustness, it does not consider the process by which these value were calculated. The process of calculating output values from input values is a critical component of understanding conceptual robustness and the propagation of uncertainty. This provides an additional level of insight which extends beyond the previous considerations of inputs and outputs over time by focusing on the intermediate values a node takes over the course of its calculation. This additional fidelity will be explored by applying the TVE and DTVE metrics to a node's intermediate values.

The results presented in this discussion of intermediate results focus on the OPS team's process of determining the longitudinal (x) locations of the aircraft along the vessel for the simple and hard cases. Initially, the OPS team utilizes the known quantities of vehicles from the global knowledge layer to determine a total weight estimate of the vehicles. The NAVARCH team utilizes this information to provide a net required center of gravity for the vehicles to ensure that the *Trim* constraint is satisfied. It is then up to the OPS team to determine the individual locations of each vehicle such that the net center of gravity across all of the vehicles matches the value provided by the NAVARCH team, such that the spacing between vehicles of different types satisfies the longitudinal spacing constraint. If no feasible arrangement of vehicles can be determined by the OPS team which exactly matches the NAVARCH requested value, the OPS team must determine an arrangement of vehicles which minimizes the difference between their determined value and the requested value. In this case, the new determined value must be communicated to the NAVARCH team for them to ensure their *Trim* constraint is still met, and if not, they must change other parameters to ensure the *Trim* remains feasible.

The OPS team determines a set of x-locations for the vehicles using two simple optimization procedures which is used to mimic designers searching for a feasible solution. The optimization process associated with exactly matching the requested and calculated values is presented in Equation 5.1. The decision variables correspond to the individual x-locations (x_i) for each vehicle in the set of all 4 vehicles (V), and a binary variable (u_i) which takes a value of 1 if and only if the vehicle needs to be accommodated. Hence, for the hard case $u_i = 1\ \forall i \in V$, and for the simple case there is only one non-zero u_i. The objective function represents the distance between the x-location requested by the NAVARCH team (x_{veh_N}) and the x-location calculated by the OPS team (x_{veh_O}). Note that the requested value from the NAVARCH team (x_{veh_N}) is included as a constant throughout this optimization process. Thus, the objective function is fully defined by the x-value calculated by the OPS team. The first constraint is the method by which the OPS team determines the net x-location of the vehicles considered, where w_i is the unit weight of vehicle i, and n_i is the number of vehicle i. Note that no binary variable is required in this constraint, as

$n_i = 0$ if the vehicle is not included. The second constraint is the spacing constraint, such that the x-locations between vehicle i and vehicle j is at least as large as the minimum spacing distance S. The next three constraints bound the x-location of vehicle i to be along the length of the vessel, and limit the binary constraint to take values of zero and one. The final constraint enforces the objective function value to be zero, by enforcing that the only acceptable calculated x-value be exactly equal to the requested value. Note that all parameters contained in the optimization scheme are represented as nodes in the OPS local knowledge structure (Figure 5.1).

$$
\begin{array}{rrcll}
\min & |x_{veh_N} - x_{veh_O}| & & & \\
\text{subject to} & \frac{\sum_{i \in V} x_i w_i n_i}{\sum_{i \in V} w_i n_i} & = & x_{veh_O} & \\
& |x_i - x_j| u_i u_j & \geq & S u_i u_j & \forall i, j \in V, j \neq i \\
& x_i & \leq & L_{BP} & \forall i \in V \\
& x_i & \geq & 0 & \forall i \in V \\
& u_i & = & \{0, 1\} & \forall i \in V \\
& x_{veh_O} & = & x_{veh_N} &
\end{array}
\tag{5.1}
$$

In the event that the OPS team is unable to exactly match the requested value, the optimization problem changes slightly, and is shown in Equation 5.2. In this formulation, the exact matching constraint is removed, and the optimization problem is allowed to progress by finding a ‘closest possible’ x-location. This step is only employed if the exact matching constraint shown above is violated. These optimization procedures have been separated to better differentiate the dynamics of the simple and hard cases, and to separate the intents of the calculations, as the latter optimization step requires subsequent communication steps between the OPS and NAVARCH teams.

$$
\begin{aligned}
\min \quad & |x_{veh_N} - x_{veh_O}| & & \\
\text{subject to} \quad & \frac{\sum_{i \in V} x_i w_i n_i}{\sum_{i \in V} w_i n_i} &= \;& x_{veh_O} \\
& |x_i - x_j| u_i u_j &\geq \;& S u_i u_j \quad \forall i, j \in V, j \neq i \\
& x_i &\leq \;& L_{BP} \qquad \forall i \in V \\
& x_i &\geq \;& 0 \qquad\quad \forall i \in V \\
& u_i &= \;& \{0, 1\} \quad\; \forall i \in V
\end{aligned}
\tag{5.2}
$$

The consideration of the simulated intermediate calculation steps required by the OPS team to determine a feasible vehicle arrangement presents significant differences between the simple and hard cases. The simple case only requires the first optimization problem to be solved. As the simple case only considers the placement of a single vehicle type, the spacing constraint and exact matching constraint become redundant. As such, determining the single vehicle location requires no intermediate calculations as the optimizer simply selects the location requested by the NAVARCH team. Since this value goes from being unknown to being determined in one step, it exhibits zero TVE and DTVE entropy. This is because there is a one-to-one mapping of the OPS team's input (decision variable) to their output (objective value). As only one value is contained in the time series, the distribution of observed values contains no value-centric uncertainty associated with intermediate calculations. The same is not true for the hard case, which will be the basis of the discussion in this section.

In the hard case, the likelihood that the exact matching constraint is satisfied is small, and as such, there is a high probability that the second optimization scheme will need to be solved. While the spacing constraint becomes redundant in the simple case, it presents significant difficulties for the OPS team in the hard case. The dimensionality of the design space created by the hard case is larger than that of the simple case, and contains a more complicated topography with a large number of local minima. For a provided NAVARCH value, there may be a large number of vehicle arrangements which yield a locally optimal answer. This requires the hard case design space to be explored more thoroughly, to maximize the likelihood of attaining a viable solution. To do so, the OPS team must consider multiple sequences of vehicles along the vessel's length (for 4 vehicles, there are 24 unique sequences), and ensure the vehicles can be arranged in such a way that the spacing constraints are satisfied, while minimizing the difference between the requested value and calculated value.

The above optimization problem was solved using a Nonlinear Generalized Reduced Gradient (NGRG) method. This method is sensitive to finding local minima, so to combat this, the optimization was run using 15 randomly sampled points in the design space. This is akin to the OPS team selecting a set of arrangements of vehicles with which to perform their analysis. The intermediate values of the decision variables (x_i) and calculated net x-location ($x_{veh,O}$) were tracked over each step of the NGRG optimizer to create representative time series. While the NGRG does not guarantee global optimality, the global optimum was determined in the case illustrated in this section. The method is sufficient to generate the data needed to demonstrate the insights which can be uncovered by applying TVE and DTVE to intermediate calculation results.

The value time series and differential value time series are presented in Figures 5.31 and 5.33, respectively. In each figure, the individual vehicle location values are highlighted in blue, while the calculated longitudinal center is shown in orange. The time series are shown as a function of intermediate iteration steps, which correspond to each step in the optimization procedure. The initial timesteps in both plots correspond to the execution steps of the first optimization procedure (Equation 5.1). This first region is characterized by few changes in value, evidenced by the relatively flat value time series and differential time series. This is due to the optimizer failing to return an exact match to the requested value, at which point the optimizer stalls out after 10 iterations. Upon the failure of the first optimization procedure, the optimization problem switches to that shown in Equation 5.2. The value time series exhibit a large number of peaks, which are a result of both the initial randomly sampled points and the the optimizer changing the base variables to search for optimality. The search process must repeat many times to explore the design space. In the differential time series plots, peaks correspond to times where a value changes significantly, and values near zero indicate a variable is being held constant across optimization steps.

An important aspect of the value and differential value time series is that the variability of the inputs is larger than that of the outputs. This is evidenced through an examination of the time series themselves, as well as the associated distributions of the time series presented in Figures 5.32 and 5.34 for the value and differential values, respectively. These histograms represent the distributions of all observed values (and differential values) over the entire course of the optimization procedure. Note that the distributions of the inputs are less peaked than the output, which suggests they are more variable. This result is intuitive for a number of reasons. The optimizer explores the design space through the manipulation of the decision variables, hence to explore more of the design space the decision variables must be swept through a larger range of values. This fact, in conjunction with the combinatorics of the problem, means the extents of the design space must be explored by varying combinations of vehicle values. The variance of the output is reduced, as varying each decision variable contributes a smaller portion to the output value (proportional to the percentage contribution of that vehicle to the total weight of the vehicles). Hence, an extreme value of an individual vehicle will be 'washed out' in the determination of the final value.

Figures 5.31 - 5.34 present all observed values and differential values throughout the optimization procedures. The application of the TVE and DTVE metrics to the distribution of values at each optimization step enables the uncertainty of the intermediate values to be tracked over time. The entropy time series are presented in Figure 5.35 for the input decision variables (blue) and output objective variable (orange).

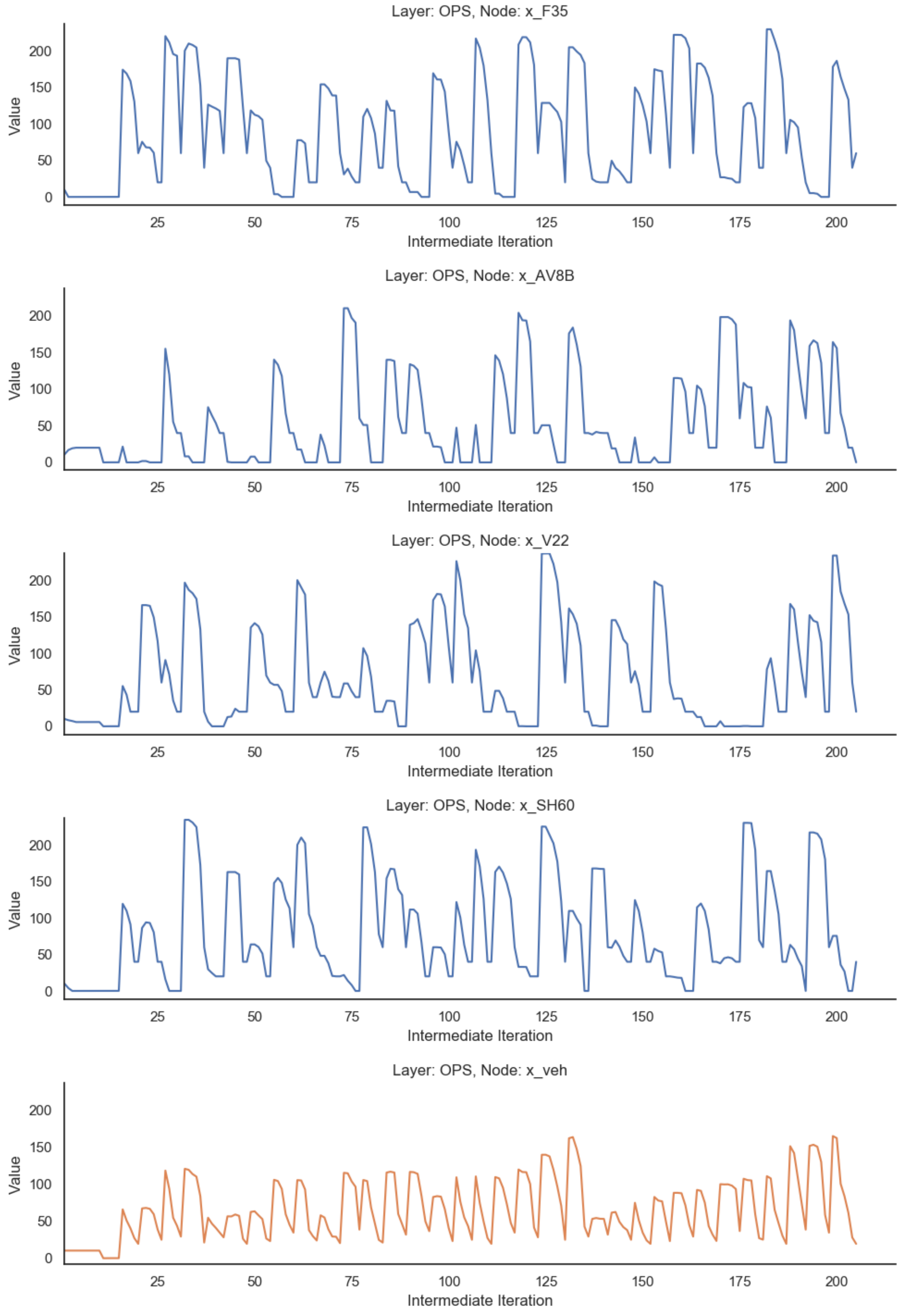


Figure 5.31: Decision Variable (Blue) and Objective (Orange) Intermediate Value Time Series - Hard Case

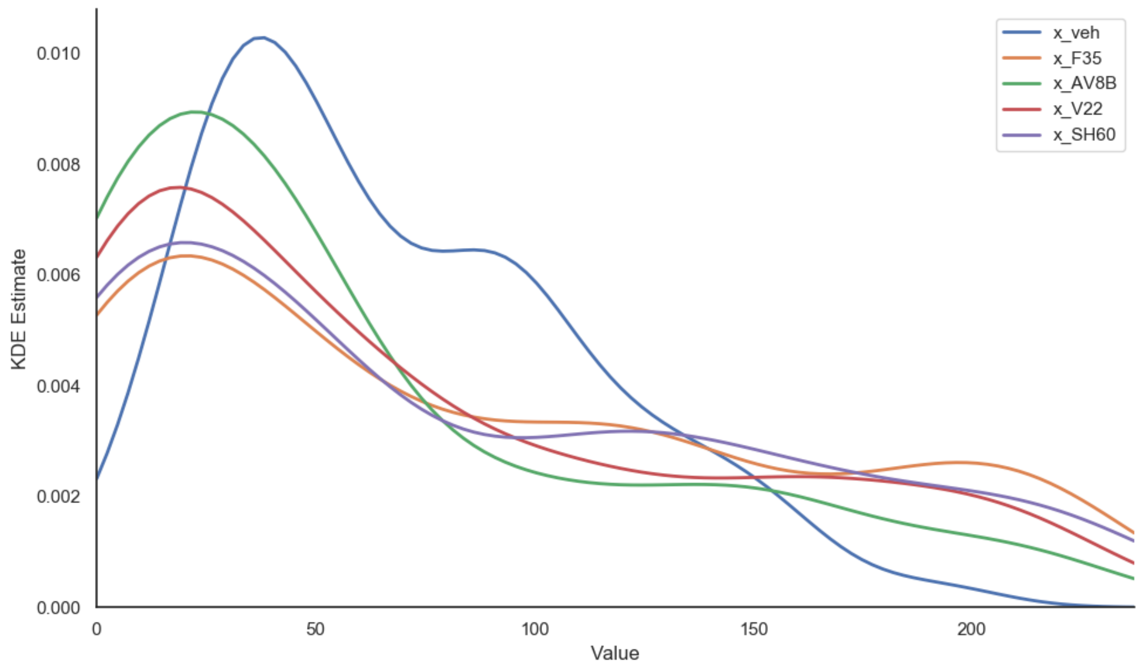


Figure 5.32: Distribution of all Observed Intermediate Values

The plots clearly identify the two regions corresponding to the two optimization procedures. The first optimization procedure is characterized by low TVE and DTVE given the optimizer's failure to converge to a result due to the exact matching constraint. The stalling of the optimizer means the values of the decision variables remain constant for a number of optimization steps, which leads to a decrease in TVE and DTVE. The removal of the exact matching constraint leads to a large growth in TVE and DTVE for both input and output values as the optimizer explores the widened design space by varying the input values. The large initial growth in TVE corresponds to the initial exploration of the design space, when there is a high likelihood that each value observed will be different from those which have been previously considered. This leads to an initial widening of the distribution of observed values. Over time, as more values are determined by the optimizer, the likelihood that a newly observed value is similar to one that has already been considered increases, which leads to the distribution becoming more peaked and results in a decrease in TVE. As a value is

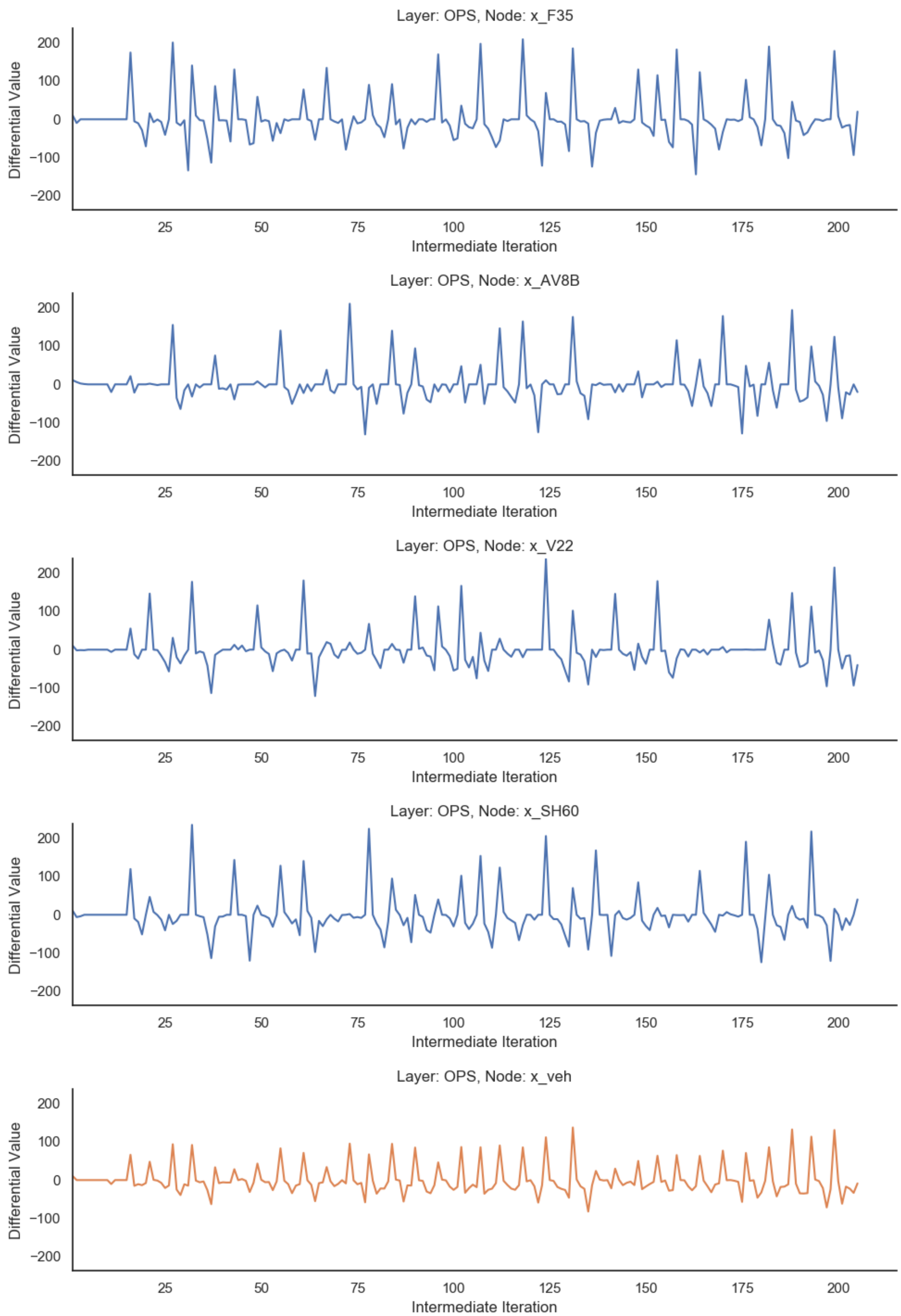


Figure 5.33: Decision Variable (Blue) and Objective (Orange) Intermediate Value Time Series - Hard Case

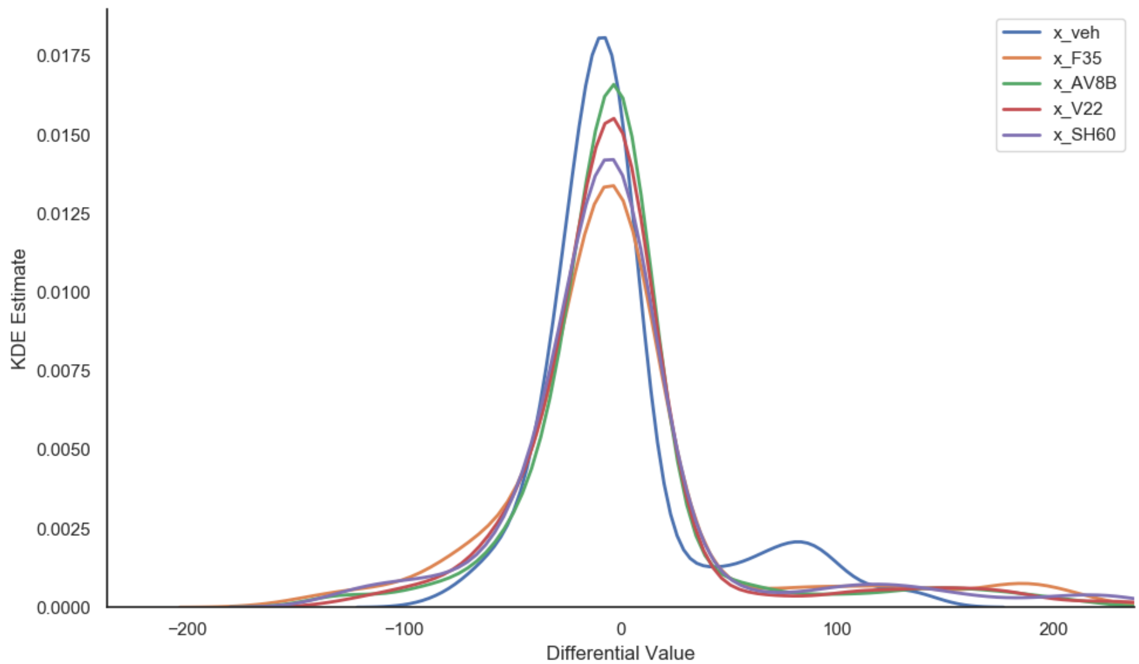
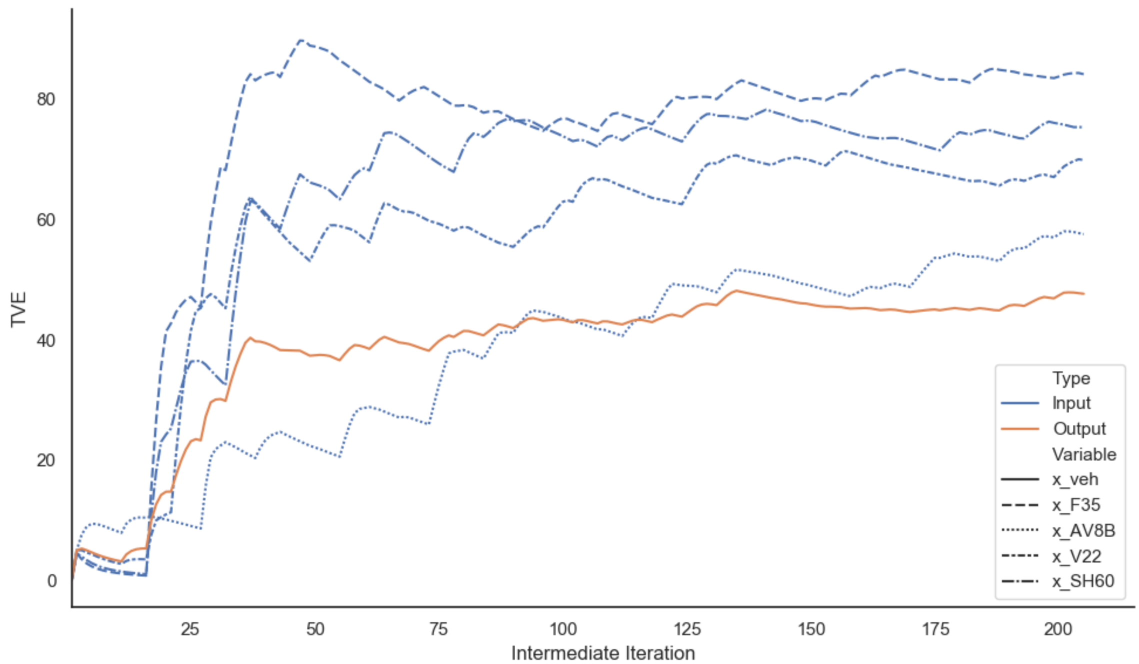


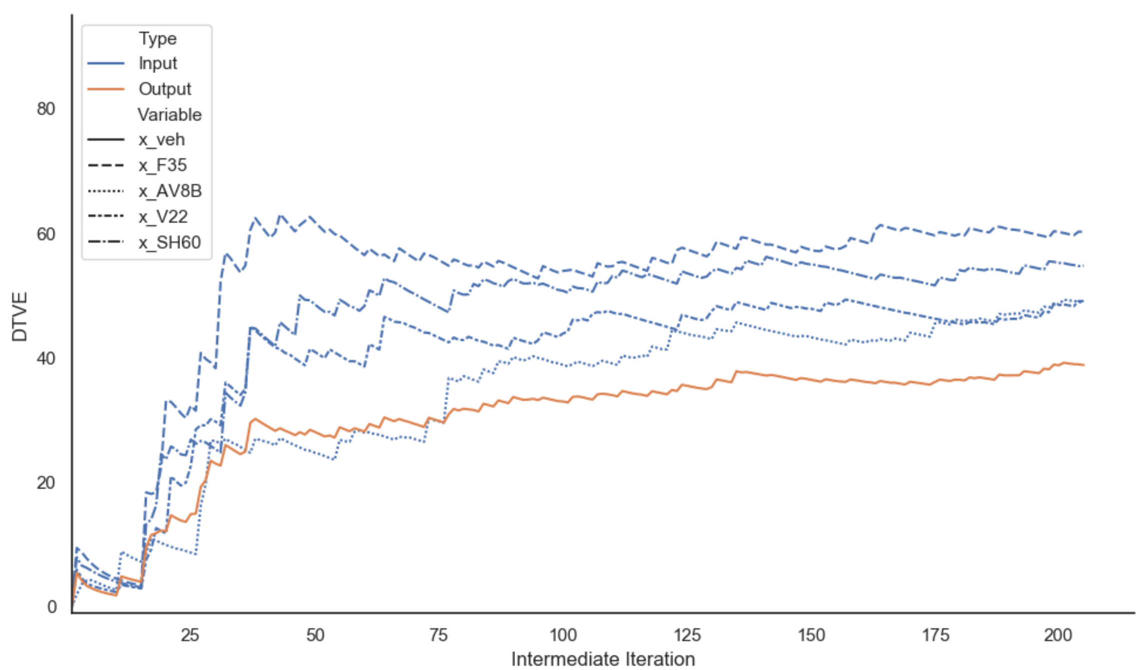
Figure 5.34: Distribution of all observed Intermediate Differential Values

seen more often, the certainty of that parameter's value increases. The final TVE and DTVE values in the time series correspond to the final value and differential value distributions shown in Figures 5.32 and 5.34.

The entropy plots illustrate a number of trends about the relative uncertainties between the considered variables. In general, the increase in entropy of the input values leads to a growth in entropy of the output values, but how does the growth of each input entropy affect the observed output entropy? These results become more intuitive by considering the case and vehicle parameters outlined in Tables 5.2 & 5.5. The combination of the number of vehicles and vehicle unit weights means that the AV-8Bs contribute the largest weight to the net center of gravity with a combined weight of 54.0 tonnes, representing 44.4% of the total vehicle weight. The V-22s, the SH-60s, and the F35-C contribute 30.1 tonnes (24.7%), 24.2 tonnes (19.9%), and 13.3 tonnes (10.9%) to the total weight, respectively. Thus, the locations of the vehicles with a larger contribution to the total weight will have a larger impact on the the net



(a) Target Value Entropy (TVE)



(b) Differential Target Value Entropy (DTVE)

Figure 5.35: Intermediate Calculation Entropy Results - Hard Case

center of gravity. This sheds more light on the trends observed in the TVE and DTVE plots. For the input variables, the lighter vehicles exhibit larger entropies related to their location than do heavier ones. This is because the locations of vehicles with smaller weights are less constrained as they impact the final result less, thus the optimizer is free to consider a wider range of possible x-locations. Conversely, the vehicles with the highest weights will impact the final answer the most, and thus the optimizer considers a smaller range of values to yield the minimal objective value. Hence, the TVE values are least for the x-locations of the AV-8Bs, and are highest for the F-35C. The DTVE plots exhibit the same trends, and are explained by the less drastic changes to the locations of heavier vehicles, and more drastic changes to the locations of lighter vehicles.

A number of conclusions can be drawn about the intermediate calculation results presented in this section. This provides an additional perspective into the conceptual robustness of calculations performed within a local knowledge structure:

1. The application of TVE and DTVE to intermediate calculations extends the idea of target-centric conceptual robustness beyond the input and output values of calculations within a knowledge structure. The dynamics of entropy growth observed in intermediate calculations highlights the growth in uncertainty of the calculation itself, rather than focusing on value changes before or after the calculation is performed.

2. Calculations which require no intermediate computations exhibit no growth in TVE or DTVE. This is due to the single input-output relation of the calculation. Computations requiring iterations will contain intermediate values. Dynamics by which the distribution of values is constructed will lead to a growth in entropy (assuming the values are not all the same across all iterations). A flattening of the distribution will lead to higher TVE, while observing values which are similar to those already observed will reduce TVE.

3. In the presented case, the relative amount of TVE and DTVE growth sheds light on how constrained the variable is. For the lighter vehicles, the optimizer is able to consider a wider range of potential x-locations than for those of heavy vehicles. Comparing the relative entropy growths of a single input and the resulting growth in output provides insight into how impactful that variable is on the uncertainty of the outcome. From a conceptual robustness standpoint, this can be used to determine more effective strategies for conducting intermediate calculations, prioritizing the criticality of variables, and understanding the sources of uncertainties. These will lead to a reduced likelihood of design churn and rework.

4. The length of the TVE and DTVE plots correspond to the number of intermediate iterations required to yield a result. As such, long time series are indicative of more difficult computations. These intermediate processes could be indicative of refinement design churn if they require significant time to compute. This provides designers a means of identifying and quantifying such cases, and enables them to act proactively to improve the design process.

5.5 Managing an Integrated Design Activity

This chapter has presented a case study to demonstrate the utility of the K-I Framework in the context of an integrated design activity. The case study has explored how the local knowledge structures of various design teams are used to develop a global information structure, which is utilized in turn to build a global knowledge structure. In addition to presenting the process, a number of entropy metrics have been presented which can be used to highlight characteristics of the framework's dynamics to inform conceptually robust decisions. To better illustrate how the K-I Framework and entropy metrics can be utilized to increase the conceptual robustness of a design process, this section presents a hypothetical scenario to synthesize the many facets of this chapter, and illustrate how the K-I Framework is useful in managing a successful design activity.

The presented scenario is as follows: Suppose that you are a design manager involved in supporting an Analysis of Alternatives (AoA) study about a new class of ships. You have been provided a Mission Need Statement (MNS) which identifies the need to create a naval platform capable of supporting the launch and recovery of SH-60 Seahawk helicopters. The AoA you have been asked to support considers three sizes of Landing Helicopter Dock (LHD) hullforms - large, medium, and small. It is your task to determine the capability of these ships to support the proposed suite of aircraft. You have been asked to provide a yes or no answer. The results of this AoA will be used to identify the alternative which satisfies the provided requirements at the lowest possible risk, and the selected alternative will progress to the preliminary design stage.

To help you conduct the AoA, you have a team of three engineers who have technical expertise in Naval Architecture, Flight Operations, and Distribution System Design. They will utilize previously developed tools which have been designed to determine the critical parameters about their technical areas. As a skilled design manager, you know the AoA has long lasting implications on the successful design of a product. However, your traditional metrics only focus on the proposed solution, not the process of generating that solution. As such, you decide to implement the K-I Framework as a way of understanding the knowledge generation process in designing the three solutions, rather than just the solutions themselves.

You ask each of your team members to map their tools into network space using the variables of their tools, and the data relations between them. You know that the tools used between groups are implemented in different softwares, so converting them to network space provides an equal means of comparison across the tools. The teams successfully produce their networks, and you add them to the K-I framework to represent each discipline's local knowledge structure.

You create an interface through which each team member can communicate with other members, such that you can track the communications between teams. This database encompasses what information has been requested by a team, and who has provided a value. Additionally, it highlights whether the requested variable has data (data status), and if so, the values of that variable (value). Any engineer can update values of previously existing parameters, or can add new parameters for other engineers to populate. This database represents the global information layer.

As the design manager, you identify a number of global parameters that will be used to compare the alternatives, and determine their feasibility. It is your task to determine how the requirement related to the SH-60 helicopters will affect the these parameters for each of the different hullforms. Thus, you add the unknown global parameters and the known requirements to the global knowledge layer as unconnected nodes.

Having set up your framework, you decide you will track your engineers' progress in designing each of the three alternatives. This will give you a means of comparing not only the final results, but also the risks, uncertainties, and difficulties associated with the development of each solution, as quantified by various entropy metrics.

You let your engineering teams get to work, and provide them a reasonable deadline to complete all of the solutions. To your surprise, the teams converge on solutions for each alternative far ahead of schedule. *Great!* You think to yourself.

You schedule a meeting to present your team's results to the decision makers. In the meeting with the higher-ups, you are told that the mission requirements have changed. The Admiral has decided that mission flexibility should be a priority, and as such, the platforms will now need to be able to accommodate a layout of 4 different aircraft types, rather than just the single layout of helicopters. You are asked to conduct the new analysis, and evaluate each platform's ability to handle the new aircraft layout. They give you a tight deadline to reconduct your work.

You get back to your office, and realize that you first need to determine whether the tools utilized by your engineers are capable of conducting the new analysis. You know that the engineers in your team are able, and that if you ask them, they will tell you anything is possible. However, you want better insight into how well-suited your team's tools are to solving the new problem.

To answer this question, you decide to analyze the K-I framework of the AoA before the requirements change. You start by analyzing the global knowledge structure. You look at the fraction unknown and DSE time series, and find that the final values have both reached zero. This tells you all of the previously unknown global parameters now have values. You expect this, as the engineers were able to yield results pertaining to the integrated design. You look at the structure of the global knowledge network and find that there are edges between all of the nodes, revealing the interdependencies of the global parameters and requirements. This highlights not only the difficulty of the design, but also that the teams have built a fully-connected global information structure through their communication pathways. Hence, the teams appear to have worked together effectively. This gives you increased confidence that your team will be able to tackle the new requirements.

To get more insight, you decide to look at the evolution of global information by looking at the database history, which will tell you more about the communicated values between teams. You find that similar to the global knowledge structure, the DSE time series is zero at the end of the design activity. This tells you that all of the global information entities have values. Additionally, the time series is monotonically decreasing, indicating that your information structure is steadily being populated with data. This makes sense to you, since all of the global knowledge values were determined, but the global information layer provides you more insight about what intermediate entities were used to realize the structure in the global knowledge layer.

You are interested in the growth of the information network over time, to see if there are any potentially problematic areas. By looking at the TE time series, you find that the TE grows quite consistently over time, meaning the size of the information structure is growing over time.

Your conclusions about the analysis of the global knowledge and information layers have confirmed that your teams are communicating well with one another. This suggests that your engineers are working together effectively to not only grow the global information, but grow it in a way that more nodes are becoming known. Although you are confident in your inter-discipline process, you are still unsure about the capabilities of each team as it pertains to solving the new problem.

To answer this, you shift your focus to the local knowledge structures. You view the TE time series and see that they remain constant over time. This tells you the teams have not altered their tools over the course of the design process. You evaluate the DSE of each team's local knowledge structure, and you find that the Distribution and Naval Architecture teams have monotonically decreasing functions, with zero final DSE. This tells you that over time, more and more of their static knowledge structures became known, and at the end of the process, they utilized their entire knowledge structures in the integration activity. *This could be problematic*, you think. *If the teams had to use all aspects of their knowledge structures, now that the requirements have changed, are there new parameters they will need to consider that haven't been integrated into their tools?*

You decide there are two potential issues that the requirements change could pose to the teams' local knowledge structures. The first is that the new problem does not influence the parameters in the local knowledge structures, but will influence their values. This could lead to more iterations between teams, or could pose a new possibility of integration failures. The second case is that the requirements change presents new parameters to be considered that are not contained in the existing local knowledge structure.

While the first issue could lead to refinement design churn and increase the time required to design the vessel, the second case proves to be the bigger issue. The second case requires the engineer to integrate new entities into their local knowledge by revising their tool. Revising their tool could present a significant difficulty. In the best case, only a small portion of the tool (knowledge structure) would need to be restructured to accommodate the new requirements. The revision of the local knowledge structure would not directly contribute to the evolution of global information and knowledge, and would present significant delays to the rest of the process. You would be able to see this through a change in local TE, however it will almost certainly mean you can't make the deadline posed by the higher-ups.

With these concerns in mind, you study the final local knowledge structure: the Operations team. You find that unlike the other two teams, the operations team has a non-zero final DSE. This tells you the team has not utilized their entire structure to answer the question. There are nodes in the network that were not required to yield a result before the requirements changed. You look in more detail at these nodes, and find out they correspond to placeholder parameters for 3 additional vehicles. *Perfect*! This tells you that the requirements change can be accommodated by the Operations team. Their knowledge structure will not need to be revised, and only the values and calculability of the structure will change.

Armed with this knowledge, you make a number of conclusions. First, your team appears to be working well together – they are effectively building a global information structure which properly leads to a global knowledge structure. In analyzing the local knowledge layers, you find that two of your teams have fully utilized their knowledge structures to yield a result. This tells you these tools are 'maxed out', and any additional parameters required will lead to the tool being insufficient to answer the new question. Since the three considered hullforms have not been changed by the requirements, you posit that additional parameters are unlikely to need to be considered. While the requirements change does impact the Operations team's analysis, their tool is set up to accommodate the new considerations. You decide that your engineering team is set up to handle the change in requirements.

Given your newfound insights from analyzing the previous case, you decide you will take a more active role in managing this new case. To do so, you will monitor the K-I Framework and entropy metrics in real time, to ensure that everything runs smoothly. You decide to use the first case as a baseline as it was a similar problem, and ended up with a successful outcome.

You tell your engineers to get to work on the largest hullform. Using the previous case as a benchmark, you monitor the entropy metrics over time, and find they are almost exactly the same as before the requirements change. You notice that the DSE of the OPS team has now decreased to zero – which tells you their tool is now being fully utilized. Over time, the teams are able to determine a solution for the large hullform in the same amount of time as the previous case. *Maybe all of my worries were for nothing. Things look great!*

The team starts on the medium hullform, and initially all of the entropy metrics agree with the previous case. You are monitoring the global information layer, to ensure the development of global information related to the integration activities are progressing well. At first, you witness the growth in TE as the global knowledge entities are translated to global information. As the teams add new entities to global information, TE increases, and as interdependencies are recognized, the TE decreases. At a certain point you observe that the global information TE has plateaued. The global information network is not growing in size, but perhaps is becoming more calculable! You analyze the DSE metric, and observe that DSE has also plateaued. You think to yourself: *The global information network is neither growing nor changing* *in calculability. This tells me no new information is being added to global information, no interdependencies are being determined between the existing information, and no previously unknown information entities are becoming calculable.* The generation of global information appears to have stalled.

While no more of the network is becoming calculable, perhaps the values of previously known nodes are changing. You look at the TVE of the nodes in the global information layer, and compare them to the previous case. You observe that the longitudinal position of the vehicles (x_{veh}) has still not been determined in global information, even though it had been determined much sooner before the requirements changed. Based on the inter-layer edges from this global information node, you find that the Operations team had been requested to provide an estimate, but has not yet done so. You look at the TVE of the x_{veh} node in the Operations team's knowledge structure, and observe a longer period of unchanging TVE. This tells you the time between the inputs and outputs of this calculation have increased in time significantly.

You are concerned that the design is experiencing *refinement design churn.* After all, the time taken to generate results between iterations has increased. You shift focus to the intermediate calculations time series, and find that in the original case, there was no increase in TVE. Now however, there is significant growth!

This growth in entropy highlights a significant change in the design process. Before the requirements change, there was a 1:1 mapping of inputs to outputs, but now the team is experiencing issues in determining a feasible arrangement of the four vehicles due to the increase in dimensionality of the problem. However, this same issue had not happened after the requirements change for the large hullform; in that case, the intermediate TVE remained low as well. You recognize that the first hullform, being much larger and heavier, was less sensitive to the locations of the vehicles. The value requested by the Naval Architecture team enabled the Operations team to match the exact matching constraint, and as such, the Operations team had an easier time allocating a feasible vehicle arrangement – the problem required far fewer intermediate iterations. Now that the proposed hullform has become more sensitive to vehicle locations, the Operations team needs to explore the design space more thoroughly because there are fewer areas to locate the vehicles, and the results are more critical. To combat the issue of a stagnant process, you allocate the Operations team more resources, in the hopes that this speeds up the process.

Shifting your attention back to the global information layer, you observe that the previously static TE and DSE metrics are now progressing smoothly. This indicates to you that communication between teams is occurring once more. The changing TE tells you the information structure is evolving, and the decreasing DSE tells you a larger portion of the global information layer is becoming known. After some time, you once again encounter an extended period where the TE and DSE metrics plateau. *Not again!*

Given your previous approach, you shift to consider the TVE of the global information nodes. You observe that the TVE and DTVE of the required pump power and GM_t are varying greatly. You follow the edges from this node and find they are connected to the Naval Architecture and Distribution teams. Unlike the previous case which in which TVE and DTVE were static, this case tells you that the global information network is neither growing nor becoming more calculable; this is due to values being negotiated between these teams in nodes that are already existing and calculable. The significant growth in TVE and DTVE indicates to you that the communicated values are not similar and are exhibiting growths in uncertainty. You decide to investigate further.

You identify the local required power node in the Distribution team's local knowledge structure. You find that the required power is a function of the fuel tank height (z_{fuel}) and height of the vehicles (z_{veh}). The TVE and DTVE of the vehicle heights is low, and is decreasing. This tells you this value is not changing. However, the entropy of the fuel tank height is increasing! This tells you that this parameter is changing values significantly and is leading to a growth in required power.

You follow the path from the local z_{fuel} parameter through global information to the Naval Architecture team, and find a similar case of entropy growth in the Naval Architect local knowledge node. But based on their structure, this node is an input to their estimate of GM_t! You now see the root of the issue.

These teams have differing objectives in relation to these two variables. The Distribution team aims to locate the fuel tanks high in the vessel to reduce the required power of the pump, while the Naval Architects want to locate the fuel tanks low in the vessel to maintain stability. Through this analysis, you have uncovered that this is the root of the entropy growth. As the vessel has gotten smaller from the previous case, the outputs have become more sensitive to changes in the inputs – hence leading to growths in entropy. To mitigate this problem, you decide to co-locate the disciplines in the same room, to assist the engineers in determining a value that is satisfactory to both parties (concurrent engineering).

After implementing this change, negotiations improve, and you witness the design progressing smoothly once more. After a short time, the global information layer is translated to global knowledge, and a solution has been determined. You note that the design of the medium size hullform took longer and experienced more issues than the large hullform, but was still able to yield a result after the change in requirements.

The teams begin evaluating the final hullform, the smallest of the alternatives. As you are monitoring the K-I Framework, you observe similar entropic trends to that of the medium hullform. Given your managerial changes and having gained an understanding of the issues presented in the medium-hullform case, you observe that the design churn has improved the ability for the Operations team to perform. However, even though you have co-located the Naval Architecture and Distribution teams, you once again witness large growths in TVE of GM_t and required pump power. You come to learn that the teams are unable to agree on a common value, as their feasible regions for the negotiated variables do not overlap. The decrease in vessel size has influenced the ability for the teams to reach a suitable parameter. For the distribution team to make their solution work, the pipes would either have to be far too large to route through the vessel, or the size of the pump would have to be massive. You have witnessed an *integration failure*, and you decide that the smallest hullform would be unable to accommodate the new requirements.

Having managed the evaluation of all three hullforms, you bring your results to the decision makers. You indicate that both the large and medium hullforms can meet the new requirements, while the smallest cannot. Given that both the large and medium hullforms meet the new requirements, the decision makers decide to use cost as a basis for their decision. They decide that because the medium hullform is cheapest, it should be the recommended alternative (Table 5.15).

Given your insights from using the K-I Framework, you are able to communicate to the decision makers that although both solutions satisfy the requirements, the evolution of integrated design information and knowledge presents far more engineering effort required to solve the problem for the medium hullform (Table 5.16). You are able to quantify these difficulties using the entropy metrics, by focusing on each alternative's ability to satisfy a requirement in a very early stage of design. Although the medium hullform has a lower estimated cost, the likelihood of emergent design failures increases as compared to the larger hullform, due to information and knowledge

Table 5.15: Selected Alternative using Traditional Evaluation

	Hullform 1	**Hullform 2**	**Hullform 3**
Size	Large	Medium	Small
Cost	High	Low	Low
New Requirements Met?	Yes	Yes	No

Table 5.16: Selected Alternative using K-I Framework Evaluation

	Hullform 1	**Hullform 2**	**Hullform 3**
Size	Large	Medium	Small
Cost	High	Low	Low
New Requirements Met?	Yes	Yes	No
Design Difficulty	Low	High	N/A

generation issues associated with the increased spatial complexity. This presents a higher possibility of schedule failures and cost overruns. So, although the preliminary cost estimate suggests that the medium hullform may be of lower cost now, the added likelihood of emergent design failures means it could in fact be far more expensive over time.

The utilization of the K-I framework has enabled you to provide decision makers with additional considerations in evaluating the alternatives. You have been able to communicate and quantify differences in the alternatives related not only to the products themselves, but also to the engineering effort which enables the creation of those products. The entropy metrics have enabled you to identify potentially problematic areas of the design and design process in a very early design stage. This mitigates the risk of selecting an inferior alternative which the traditional methods alone would have suggested were dominant. Through the utilization of the K-I framework, you have enabled decision makers to better select the alternative which will be more conceptually robust to future exogenous factors.

CHAPTER VI

Contributions

6.1 Contributions

This thesis has presented a novel perspective of conceptual robustness focused on design knowledge generation. The novel contributions of this research include:

- Creation and implementation of a novel network framework (the K-I Framework) to capture the interplay between information gathering and knowledge generation. The primary contributions of the framework are:
  - Developed a multi-layer network that encapsulates data, information, and knowledge in relation to a design activity.
  - Enabled the ability to track the development of knowledge structures at local and global levels of design.
  - Created a unique method of representing sources of information and knowledge structures using networks.
  - Created a platform to study the dynamics of knowledge generation and knowledge refinement over the course of a design activity in a multi-agent design environment.

  - The ability to utilize networks to gain a knowledge-centric perspective of conceptual robustness.

- Creation of novel information-theoretic network entropy metrics to understand information contained within K-I networks:

  - A new implementation of the PageRank algorithm to study the information content of a network structure (Topological Entropy (TE)).
  - Leveraging Generalized Cumulative Residual Entropy as a novel metric to understand the evolution of target node values over time (Target Value Entropy (TVE)).
  - Creation of a binary entropy measure to study the calculability of a knowledge structure over time (Data Status Entropy (DSE)).

- Developed a new, knowledge-centric perspective of conceptual robustness which focuses on the knowledge structure used to create products:

  - Utilized Target Value Entropy (TVE) to understand path dependencies and predictability of a design approach.
  - Introduced the ability to study and quantify excessive rework using a novel algorithm based on the developed K-I framework.
  - Illustrated the ability to quantify the inability to integrate information sources during rework activities as a result of knowledge structures.

- Demonstrated the utility of the K-I framework using two case studies focused on local and global knowledge structures:

  - The local case study highlights how different knowledge structures are developed using different calculation approaches.

- The global case study reveals the ability to understand conceptual robustness as a result of inter-agent communication in a design activity.

6.2 Future Topics of Interest

While the research presented in this thesis has provided many new concepts toward gaining new insights into conceptual robustness, there is still much to be explored. The focus of this thesis was to develop a flexible platform which could be leveraged to explore many additional aspects as it related to knowledge-centric design. There are a number of additional aspects which could be explored to expand the results presented by the case studies in Chapter IV and Chapter V. These future topics include, but are not limited to, the following:

- *The refinement of existing, and development of new, entropic measures.* The entropy metrics presented in this work consider the growth, calculability, and values contained within knowledge and information structures. While these metrics provide a base by which to study the evolution of layers within the K-I Framework, there are likely other formulations of metrics which provide different insights. For example, there are currently no entropy metrics related to the accuracy or quality of values over time.

- *Entropy normalization methods.* While a number of the proposed metrics can be represented by a distribution which maximizes entropy, no theoretical maximum entropy exists for TVE and DTVE. The development of an appropriate normalization process would better enable comparisons across different variables or design activities. This would be useful in establishing appropriate entropy thresholds of concern for variable changes, such that a designer could be notified of a potential issue across a wide range of activities.

- *Predictive measures of entropy growth.* The proposed framework is able to track entropy growth over time, and use this as a means to mitigate late stage failures by recognizing them early on in the design activity. However, the utility of the framework would be improved by also predicting entropies of future values utilizing existing entropy values. For example, the entropies of a number of inputs coupled with the structures of the networks, could be used to propagate the entropies from inputs to outputs. Predictive measures of entropy growth could be used to better infer strategies for convergence on a solution, and avoid cases of lock-in. This would also improve the ability to recognize potential emergent design failures.

- *Predicting product-centric design outcomes.* The knowledge-centered approach would be improved by relating the various entropy growths to product-centric outcomes. This could be achieved by considering ensembles of product outcomes, and using entropy to understand the evolution of these outcomes throughout the development of local and global knowledge structures. This would translate the impact of knowledge-centric entropy changes to product space and thus enhance the predictive power of knowledge-centric measures. It could also be used to provide new conceptual robustness insights related to knowledge structure development.

- *Exploring the trade off between knowledge structure complexity and robustness.* Further work is required to understand how conceptual robustness is impacted by the complexity of a knowledge structure. The metrics proposed in this thesis quantify a number of aspects relating to complexity, but the relation is not fully understood. This would better inform the applicability of tools in early design stages, and the resiliency of these tools to the presence of exogenous factors.

- *Utilizing entropy metrics to inform knowledge structure development.* This work has validated the ability for the developed entropy metrics to quantify the development of local and global knowledge structures. The work could be extended by utilizing these insights to direct the development of local and global structures, rather than monitoring them as they are developing.

- *Prioritization of inputs for a given knowledge structure.* Utilizing approaches such as percolation theory toward node data statuses would quantify the robustness of knowledge structures to missing data. This would not only provide another perspective of knowledge structure robustness, but could also be used to inform the selection of information structures, and prioritize data sought for calculations.

- *Integration of additional information theoretic metrics into the framework.* The inclusion of additional metrics used in information theory would enable a suite of new analyses to be performed. New, interesting insights would be gained by considering measures such as mutual information, the marginal utility of information, and conditional entropies.

- *Strategies for reducing the computation time of the growing distributions over time.* In order for the K-I Framework to be effectively utilized in real time, additional work is required to improve computation times. One possible improvement is through new strategies for binning growing distributions while maintaining the predictive power of the metrics. The framework could also be improved through the utilization of high performance computing resources.

- Converting the analysis to consider actual time rather than calculation step, which enables the intermediate calculations and input/output values to be combined into the same plot.

- Evaluating global knowledge structure development utilizing the framework in the context of concurrent engineering and set-based design approaches.

- Additional case studies to explore additional exogenous factors.
- Application of the framework to larger, more complex case studies.

APPENDIX

APPENDIX A

Integration Case Study - Design Tools

This appendix outlines the equations utilized by each of the teams in the Integrated Framework Case Study (Chapter V). The equations listed for each team are presented in the form in which they are implemented in each tool. The combination of equations for each team fully defines each local knowledge structure. The equations presented for each team were implemented as Excel spreadsheets, with each variable being represented as an individual cell.

The constraints listed for each team define the feasibility bounds for the variables listed. These bounds are used as inputs to guide the optimization procedures to yield results deemed acceptable by each group. All unlisted variables are unbounded.

A.1 Distribution (DIST) Equations

$$\gamma_{fuel} = \rho_{fuel} * g \tag{A.1}$$

$$P_1 = 0 \tag{A.2}$$

$$P_2 = 0 \tag{A.3}$$

$$V_1 = 0 \tag{A.4}$$

$$V_2 = \frac{vol\_flow\_rate}{\pi(\frac{pipe\_diameter}{2})^2} \tag{A.5}$$

$$system\_head = \frac{P_2 - P_1}{\gamma_{fuel} + (z_{veh} - z_{fuel}) + \frac{V_2^2 - V_1^2}{2g}} \tag{A.6}$$

$$required\_power = \frac{\gamma_{fuel} * vol\_flow\_rate * system\_head}{1000} \tag{A.7}$$

Feasibility Constraints

$$pipe\_diameter \geq 0.1 \tag{A.8}$$

$$pipe\_diameter \leq 1.0 \tag{A.9}$$

$$required\_power \geq 0.0 \tag{A.10}$$

$$required\_power \leq 560.0 \tag{A.11}$$

A.2 Naval Architecture (NAVARCH) Equations

$$V = L_{WL} * B * T * C_B \tag{A.12}$$

$$Disp = \rho_{SW} * V \tag{A.13}$$

$$W_{tot} = W_{DWT} + W_{LS} \tag{A.14}$$

$$W_{DWT} = W_{misc} + W_{veh} + W_{fuel} \tag{A.15}$$

$$W_{fuel} = \frac{\rho_{fuel} * V_{fuel}}{1000} \tag{A.16}$$

$$\% = 100 * \frac{W_{tot} - Disp}{Disp} \tag{A.17}$$

$$x_{DWT} = \frac{W_{misc} * x_{misc} + W_{veh} * x_{veh} + W_{fuel} * x_{fuel}}{W_{DWT}} \tag{A.18}$$

$$z_{DWT} = \frac{W_{misc} * z_{misc} + W_{veh} * z_{veh} + W_{fuel} * z_{fuel}}{W_{DWT}} \tag{A.19}$$

$$x_{tot} = \frac{W_{DWT} * x_{DWT} + W_{LS} * x_{LS}}{W_{tot}} \tag{A.20}$$

$$z_{tot} = \frac{W_{DWT} * z_{DWT} + W_{LS} * z_{LS}}{W_{tot}} \tag{A.21}$$

$$GM_t = BM_t + KB - z_{tot} \tag{A.22}$$

$$GM_L = BM_L + KB - z_{tot} \tag{A.23}$$

$$Trim = \frac{L_{WL}(x_{tot} - LCB)}{GML} \tag{A.24}$$

Feasibility Constraints

$$GM_t \geq 0.0 \tag{A.25}$$

$$|\%| \leq 0.5 \tag{A.26}$$

$$|Trim| \leq 0.01 \tag{A.27}$$

$$x_{fuel} \geq 0.0 \tag{A.28}$$

$$x_{fuel} \leq L_{WL} \tag{A.29}$$

$$z_{fuel} \geq 4.0 \tag{A.30}$$

$$z_{fuel} \leq D + 3.0 \tag{A.31}$$

A.3 Operations (OPS) Equations

$$W_i = n_i * w_i \tag{A.32}$$

$$SFC_i = \frac{fuel\_capacity_i * \rho_{fuel}}{flight\_time_i} \tag{A.33}$$

$$flight\_time_i = 2 * combat\_radius_i * speed_i \tag{A.34}$$

$$V_{fuel,i} = \frac{1}{\rho_{fuel}} * endurance\_days * n_i * sortie\_rate * flight\_time_i * SFC_i \tag{A.35}$$

$$refuel\_time = 60 * (\frac{24}{sortie\_rate} - 3.36 * (\frac{flight\_time_i}{2}) - 4.0) \tag{A.36}$$

$$flow\_rate_i = 60 * \frac{fuel\_capacity_i}{refuel\_time_i} \tag{A.37}$$

$$W_{veh} = \sum_i W_i \tag{A.38}$$

$$x_{veh} = \frac{\sum_i W_i * x_i}{W_{veh}} \tag{A.39}$$

$$z_{veh} = \frac{\sum_i W_i * z_i}{W_{veh}} \tag{A.40}$$

$$V_{fuel} = \sum_i V_{fuel,i} \tag{A.41}$$

$$vol\_flow\_rate = \max(flow\_rate_i) \tag{A.42}$$

Feasibility Constraints

$$|x_i - x_j| \geq 20.0 \quad \forall i, j \in V \tag{A.43}$$

$$x_{veh} \geq 0.0 \tag{A.44}$$

$$x_{veh} \leq L_{WL} \tag{A.45}$$

$$z_{veh} = D + 3.0 \tag{A.46}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

Aftab et al. (2001). *Information Theory: Information Theory and the Digital Age*. Tech. rep. Massachusetts Institute of Technology.

Andrews, David J. (2012). "Art and science in the design of physically large and complex systems". In: *Proceedings of the Royal Society* 468, pp. 891–912.

Bernstein, Joshua I (1998). "Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices". PhD thesis. Massachusetts Institute of Technology.

Box, George E. P., William G. Hunter, and J. Stuart Hunter (1978). *Statistics for Experimenters*. Tech. rep. New York: Wiley.

Braha, Dan and Yaneer Bar-Yam (2007). "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results". In: *Management Science* 53.7, pp. 1127–1145.

Chalfant, Julie (2015). "Early-Stage Design for Electric Ship". In: *Proceedings of the IEEE* 103.12, pp. 2252–2266.

Chang, Tzyy-Shuh and Allen C Ward (1995). *Conceptual Robustness in Simultaneous Engineering: A Formulation in Continuous Spaces*. Tech. rep., pp. 67–85.

Chang, Tzyy-Shuh, Allen C Ward, et al. (1994). *Conceptual Robustness in Simultaneous Engineering: An Extension of Taguchi's Parameter Design*. Tech. rep., pp. 211–222.

Conklin, Jeffrey (2006). *Dialogue Mapping: Building Shared Understanding of Wicked Problems*. West Sussex, England: John Wiley & Sons.

Cornell, J. A. and A. I. Khuri (1987). *Response Surfaces: Designs and Analyses*. Milwaukee, Wisconsin: ASQC Quality Press.

Cross, Nigel (2001). "Chapter 5 - Design Cognition: Results from Protocol and other Empirical Studies of Design Activity". In: *Design Knowing and Learning: Cognition in Design Education*, pp. 79–103.

Dehnad, Khosrow (1989). *Quality Control, Robust Design, and the Taguchi Method*. Springer Science & Business Media.

Goldschmidt, Gabriela and Maya Weil (1998). “Contents and Structure in Design Reasoning”. In: *Design Issues* 14.3, pp. 85–100.

Goodrum, Conner J., Colin P. F. Shields, and David J. Singer (2017). “Investigating the Impact of Distributed System Routing Densities on Vessel Operability”. In: *ICCAS 2017*. September. Singapore: RINA, pp. 26–28.

Goodrum, Conner J., Samantha Taylordean, and David J. Singer (2018). “Understanding Initial Design Spaces in Set-Based Design using Networks and Information Theory”. In: *13th Intenational Maritime Design Conference (IMDC)*. Helsinki, Finland.

Kassel, Ben, Seth Cooper, and Adrian Mackenna (2010). *Rebuilding the NAVSEA Early Stage Ship Design Environment*. Tech. rep. Arlington, VA: American Society of Naval Engineers (ASNE).

Laxton, M (1969). “Design Education in Practice”. In: *Attitudes in Design Education*.

Marine Structures Design Laboratory (2019). *NA 470/475 Educational Toolset*.

NAVSEA (2012). *Ship Design Manager (SDM) and Systems Integration Manager (SIM) Manual*. Tech. rep.

Newman, Mark E.J. (2010). *Networks: An Introduction*. Oxford: Oxford University Press.

Oxford (2019). *Definition of Information. https://www.encyclopedia.com*.

Page, Scott E. (2005). “An Essay on the Existence and Causes of Path Dependence”. In: *University of Michigan, USA*, pp. 1–37.

Polanyi, Michael (2009). *The Tacit Dimension*. University of Chicago Press.

Rao, M. et al. (2004). “Cumulative Residual Entropy: A New Measure of Information”. In: *IEEE Transactions on Information Theory* 50.6, pp. 1220–1228.

Rashevsky, N (1955). *Life, Information Theory, and Topology*. Tech. rep., pp. 229–235.

Shannon, C.E. (1948). “A Mathematical Theory of Communication”. In: *The Bell System Technical Journal* 27.3, pp. 379–423.

Shields, Colin P. F. (2017). “Investigating Emergent Design Failures Using a Knowledge-Action-Decision Framework”. PhD thesis. University of Michigan.

Shields, Colin P. F. and David J. Singer (2017). “Naval Design, Knowledge-Based Complexity, and Emergent Design Failures”. In: *Naval Engineers Journal* 129.4, pp. 75–86.

Singer, David J., Norbert Doerry, and Michael E. Buckley (2009). "What Is Set-Based Design?" In: *Naval Engineers Journal* 121.4, pp. 31–43.

Sypniewski, Michael J. (2019). "A Novel Analysis Framework for Evaluating Predisposition of Design Solutions through the Creation of Hereditary-Amelioration Networks Derived from the Dynamics within an Evolutionary Optimizer". PhD Thesis. University of Michigan.

Webber, Melvin M. and Horst W. J. Rittel (1973). "Dilemmas in a General Theory of Planning". In: *Policy Sciences* 4.December 1969, pp. 155–169.

Yassine, Ali et al. (2003). "Information hiding in product development: the design churn effect". In: *Research in Engineering Design* 14, pp. 145–161.