Marine Design XIII – Kujala & Lu (Eds)
© 2018 Taylor & Francis Group, London, ISBN 978-1-138-34069-5

Understanding initial design spaces in set-based design using networks and information theory

C. Goodrum, S. Taylordean & D.J. Singer
Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, Michigan, USA

ABSTRACT: The US Navy has adopted Set-Based Design (SBD) as the preferred design method for addressing increasingly complex naval products. SBD requires functional groups to create and negotiate design spaces, as well as to manage convergence of the intersection of these regions over time. While there has been significant research conducted on negotiated spaces, little has been done to understand the creation of design spaces from the outset of the SBD process. Understanding initial design space landscapes is crucial to effectively implement SBD. The creation of design spaces is predicated by the design tools used to generate design knowledge. This concept paper investigates the creation of design space information from the outset of the SBD process to understand whether employed tools predicate design spaces, if proper negotiation channels are open between design groups, and if all appropriate parties are involved in set negotiations. Previous research is leveraged to create an integrated design space network, based on representative design tools. This network is analyzed using information theory to determine inherent hierarchical network structures, which provides insight to the landscape of design spaces using the limited information present in early design stages.

1 INTRODUCTION

1.1 *Motivation*

Set-Based Design (SBD) is a convergent design method that is better suited to tackle complex engineering design problems as compared to traditional methods (Singer 2003, Bernstein 1998, Sobek 1997, Liker et al. 1996, Ward et al. 1995). While traditional design methods have been used to successfully design ships in the past, they are inadequate to handle the increasing complexity of modern naval design problems. This is because traditional methods, based on the design spiral approach (Evans 1959), attempt to develop a 'best' design satisfying all design requirements at each design iteration. While this method often produces converged designs, it falls apart in the face of complex design landscapes, where external, internal, and temporal factors mean the design problem can never be fully understood (Shields 2017a). This is due to both the path-dependent nature of generating design knowledge, and time and budget limitations (Singer et al. 2009, Shields 2017a).

On the other hand, SBD requires the creation of broad sets of design parameters which are used to define tradeoff information as it relates to the design. These sets are developed concurrently, and are gradually narrowed with increasing information fidelity until a more globally optimal design point is revealed. This removes the requirement to search for 'the best' feasible designs at each point in the process and instead aims at discarding designs which are infeasible or dominated. SBD delays decision making until later design stages when there is more information to make an informed decision. This process has enabled more flexibility in adapting to design changes, and removed constraints early on in the design process which often predicate solutions (Singer et al. 2009).

The benefits of applying SBD to complex design activities have led to the US Navy utilizing the SBD method for a number of activities, including the ship-to-shore-connector and requirements evaluation (Mebane et al. 2011, Doerry & Fireman 2009). SBD has been well advocated for in regard to future naval ship designs (Keane et al. 2006, Kassel et al. 2010, Eccles 2010, Doerry 2009, Sullivan 2008).

There have been a number of challenges to widespread adoption of SBD in the US Navy. One major problem has been transitioning design tools that were successful in the traditional framework to enable success in SBD. Historically, there was a large push in traditional design approaches to remove the human from the design process through the use of automated codes. Improved computational power spurred the creation of large integrated toolsets and synthesis models. These have increasingly been used to select designs, instead of being used as a design tools (McKenney et al. 2012), and are ripe sources of design bias (Sypniewski & Singer 2017). These tools have had extensive investment, research, and time dedicated to their creation, and are not likely to be replaced. As such, it is difficult to fit many tools which were effective in traditional point-based design approaches into the new SBD framework. The structure of the analysis tools and widespread removal of the designer from the design process has drastically affected the way design space negotiations are made and has had huge implications of the success of SBD.

To date, an extensive amount of research has been conducted on the way designs are negotiated between design groups (Parsons et al. 1999, Singer 2003, Gray 2011, Cuneo 2013), but little has been done to understand the role these tools play on structuring the SBD process. Significant information can be gained by mapping the relations between variables used in design tools to a network framework (Parker 2014). This framework provides insight to the design process, and uncovers variables important for internal and negotiated design spaces. It also provides a novel way of understanding the impact the approach to an engineering analysis has on the ability to negotiate between design groups.

This paper outlines a novel framework of investigating design tools and how they relate to the design method. The paper introduces the framework, how it can be used to analyze a design process, and how it can be used to enable SBD. Section 2 familiarizes the reader with the network terminology used throughout the paper. Section 3 describes how the variable network was created, and Section 4 outlines how hierarchical network structures can be elicited through the created network. Section 5 presents a representative case study to demonstrate the method, and the results are presented in Section 6. Section 7 expands the discussion to the impacts this method has on structuring existing tools to be applicable for SBD.

1.2 *Related research*

Significant work has been done on understanding the impact design tools have on shaping the naval design process (McKenney 2013, Parker 2014, Parker & Singer 2015). McKenney (2013) identifies that the overreliance on design tools leads designers to treat tools as 'black boxes' and can cause a misunderstanding of the entire design problem. Additionally, he highlights the need for tools that facilitate the exchange of information. Parker (2014) utilizes a multi-partite network framework to elucidate the structure of design tools and inspired much of the work conducted in this paper. Parker defines three node types in his formulation: variables, functions, and disciplines. Variables are related to one another through the function layers, and are prescribed a design discipline dependent on the nature of the function. Parker's approach enables a mapping of variable interactions through functions, to design disciplines which led to a larger understanding the structure of naval design.

One shortcoming of the existing research is the requirement to prescribe what functions lend themselves to which disciplines ahead of time. While in some cases these allocations are obvious, many times functional memberships are not totally discrete. The framework presented in this paper enables the mapping of variables to disciplines organically through the structure of the network itself. The proposed method enables the disciplines to be uncovered without the need to define them a-priori. This research also provides insight as to how groups should be organized to maximize the efficiency of information flow, and provides insight as to which variables should be used to negotiate between groups.

2 NETWORK TERMINOLOGY

Networks are utilized throughout this work to uncover the structure of design tools and a design process. *Networks* (or *graphs*) are powerful and versatile tools which abstract representations of a system using points (*nodes*) and lines (*edges*). Nodes represent entities, and edges represent relationships between them. Edges can be directed or undirected, depending on the nature of the relationship. Networks' ability to represent and utilize information of varying fidelity makes them powerful tools to study design. Abstracting design tools to network space allows the structure of the tool to be studied independent of how it is implemented. The study of how these nodes and edges relate to one another enables the study of 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 the paper. For an in-depth study of networks, see Newman (2010).

A key metric used to understand a network's structure is the degree distribution. A node's *degree* is the number of edges connected to it. For directed graphs, additional information can be gained by considering *in-degree* and *out-degree*, which measure the number of inbound and outbound connections a node has, respectively. These are node-centric measures of structure, and are the basis for more global structural perspectives, such as *centrality* and *communities*.

Additional information about the network structure can be gained by considering paths through the network. A *path* is a sequence of nodes such that every consecutive node in the sequence is connected by an edge. A *random walk* is a path across the network created by taking repeated random steps between nodes by moving along connected edges. Beginning at an initial vertex, an edge is selected at random to create the next step in the path, and the process is repeated. Random walks allow for nodes and edges to appear multiple times the sequence defined by the path, which is an important distinction applied to the analysis in this paper.

In order to deduce additional information about the way networks are structured, *community detection* algorithms are often employed. This refers to the division of the network into groups, or *communities*, according to the pattern of edges in the network such that there are a large number of connections within a group and relatively few connections between communities. A large number of community detection algorithms exist all of which attempt to classify nodes dependent on their relations to other nodes in the network.

3 THE VARIABLE NETWORK

To map the structure of the design tools, a *variable network* is created. The variables added to the network are any parameters which are involved in a calculation in a design tool (length, beam, etc.). The variables used to generate design information will be finite, and hence can be mapped to a network. This network represents variables as nodes. Edges represent functional (mathematical) relationships between variables. These relations are represented as directed edges. A directed edge is drawn from variable A to variable B if and only if A is used to calculate B. That is to say, directed edges point to dependent variables from the variables used to calculate them. An example is shown in Figure 1.

By formulating the network in this way, purely independent variables (inputs) have zero in-degree and non-zero out-degree. Conversely, outputs will contain zero out-degree and non-zero in-degree. Nodes with non-zero in and out degrees are intermediate dependent variables. Also note that in this representation, no nodes should have both zero in and out degrees, as this would imply variables exist which have no effect on any other variable. In general, the graph should also be acyclic. This is intuitive, as any directed cycle would result in a circular variable references which cannot occur. This is not to say there are no cycles in the design process, since we know the point-based design approach is iterative. Instead, this variable network is representative of a single loop around the design spiral, or through a single iteration of design information. Each subsequent iteration of a design will affect the independent variables (inputs), which will in turn propagate through the network to alter the dependent variables.

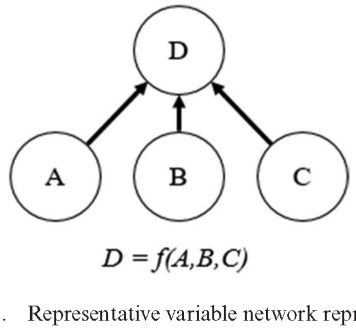

Figure 1. Representative variable network representing a variable D as a function of variables A, B and C.

Using this network generation process, the complete variable network can be built by including all variables used in a design tool, by evaluating the equations used to generate the design information. Note that the resulting network does not explicitly represent the mathematical function relating the variables, it simply represents that variables are functionally related. This is advantageous in generating the network, especially in the limit of large networks, which can arise as software and toolkits become more complicated. For small tools, it may be viable to map variables manually, however for larger software packages, the network generation tool could be automated by tracing data references.

The construction of the variable network in this manner yields a number of benefits. Firstly, it is quite straightforward to create. Also, it provides transparency to the way a design tool and design process is structured from an information perspective. As design can be viewed as the act of generating knowledge for decision making over time (Parker 2014), this approach provides a framework that enables quantitative analyses. The framework is flexible enough to enable the generation of significant insight to the design process, independent of the implemented design tools used, by using various network-centric metrics and algorithms. This information can be utilized to better align tools with the selected design approach.

4 HIERARCHICAL NETWORK STRUCTURES

As discussed in Section 2, identifying communities within a network not only provides additional information about the network's structure, but aims to uncover implicit relations within the network by studying the connections of nodes. Traditional community detection approaches are based on modularity maximization which aim to find natural divisions of a network into groups of nodes which have many edges within the group, and few between groups. Many algorithms aim to do so by solely considering the structure of the network, while others determine divisions using the way information flows through the network (Newman 2010). The flow of information, as well as the information itself can reveal a lot about a network's structure and hence community structures. The first algorithm applying information theory to community detection in networks was posed by Rosvall & Bergstrom (2008), and was given the name *InfoMap*.

InfoMap reveals inherent hierarchical structures implicit within networks. The authors apply information theory to a network by conducting a random walk on the network long enough to visit all nodes at least once. Over the course of this random walk, certain nodes are more likely to be visited than others due to the network structure. Then, Huffman codes (Huffman 1952) are used to assign each node a unique binary bit string, the length of which is based on the frequency which a node is visited in the random walk. In this way, a random walk can be quantified by a single binary string based on the nodes encountered. Then, for each possible community division of the network, the task is to find the shortest length of the bit description of this random walk. This will be the most efficient partition of the network from an information-transmission perspective. Making use of Shannon's source coding theorem (Shannon 1948), the average number L of bits per step in the walk is equal to the entropy of the random walk, which is used to derive the map equation, defined as:

$$L = qH(Q) + \sum_{g} p_g H(P_g) \tag{1}$$

where q is the fraction of time that a random walk spends hopping between groups, p_g is the fraction of time the random walk spends within and exiting group g, Q represents a sequence of entry label nodes (nodes in which a random walk enters a group), and P_g represents the sequence of the rest of the nodes in group g through which the random walk passes. The function $H(x)$ corresponds to Shannon's information entropy and is defined as:

$$H(x) = -\sum_{i} x_i \log_2 x_i \tag{2}$$

Thus, $H(Q)$ accounts for the entropy of the sequence of groups that a random walk passes through, and $H(P_g)$ is the entropy of the nodes traversed within group g. This method has been shown to effectively partition graphs into communities, and has been extended to determine multiple hierarchical partitions within networks (Rosvall & Bergstrom 2009).

5 CASE STUDY

A representative case study was created to demonstrate the creation and analysis of the variable network. The case study is based on the design activities of University of Michigan (UofM) Naval Architecture and Marine Engineering senior undergraduates in completing their capstone ship design course. Students utilize a number of in-house design tools to conduct a preliminary point-design of a containership, given a set of design requirements. Many decades of students have utilized these tools for their designs, so they are well verified and validated. Studying these point-based tools enables the structure of traditional design tools to be understood by studying the structure of the associated variable networks in the context of an integrated design activity.

The tools considered in this case study include the UofM Cost Prediction, Midship Section design, and Weights I spreadsheets, the UofM Propeller Optimization Program (POP) (based on Parsons (1975) and Oosurveld (1975)) and UofM Powering Prediction Program (PPP) (based on Holtrop & Mennen (1982) and Holtrop (1984)). These tools are used to determine the ship's cost, structure, weight, propeller design, and powering requirements, respectively. While there are additional tools used by students, the tools considered in this case study provided enough variables to demonstrate the method. Larger scale implementations of the method are left for future work.

These preliminary design tools are based on a combination of regression analyses, experimental data, and theory, which are implemented through the form of C++ codes, Excel spreadsheets, and Fortran codes. The differences in implementations of analysis codes are akin to implemented design tools in industry used by disparate design teams. The different analysis methods are included to highlight the flexibility of this approach, as results are not predicated on the implemented analysis tool, but rather focus on the structure of combined variables as a method of yielding a solution.

Each of these tools were dissected to determine the equations and variables used in their analyses. Due to the limited size and numbers of the tools utilized, it was practical to study the structure of each software manually. For each considered tool, each utilized variable was recorded, in addition to the dependencies between variables. Careful attention was paid to ensuring all variables were

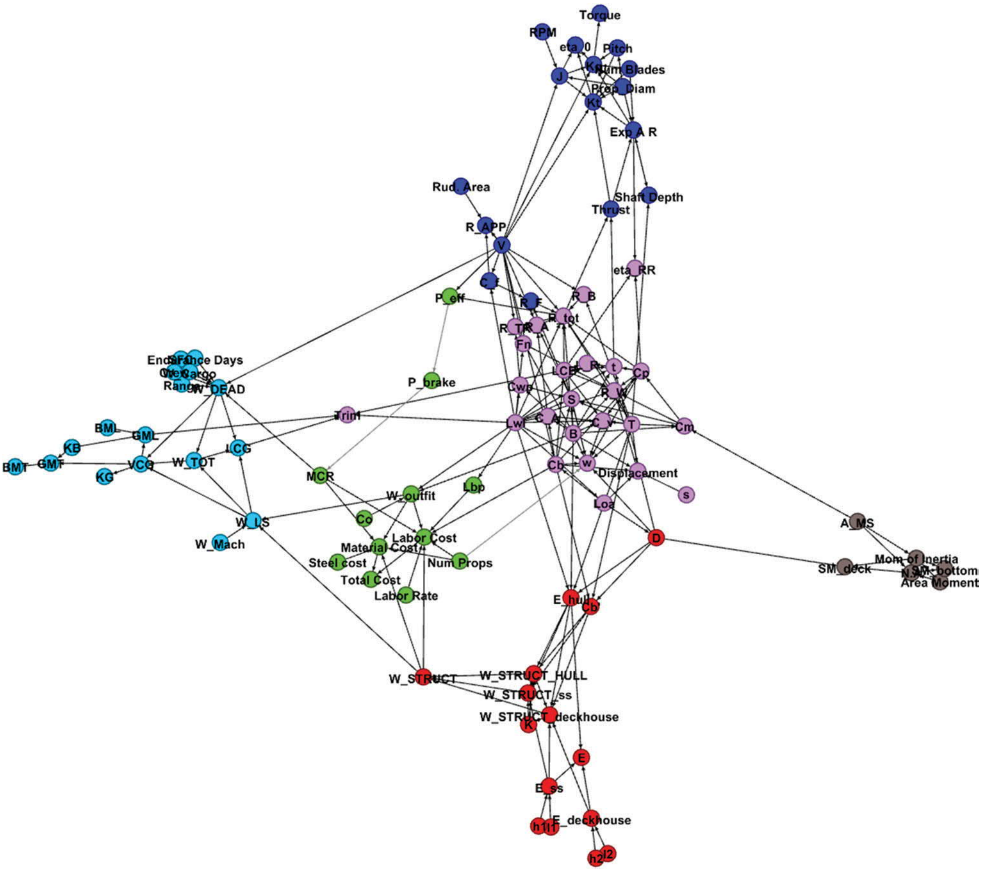

Figure 2. Variable network created using UofM design tools for preliminary containership design. The resulting graph visualization illustrates clear clustering. Nodes are colored according to the Louvain algorithm and represent different identified communities.

accounted for, and accounting for various notations between tools. These relations were used to create the variable network shown in Figure 2 using the process described in Section 3. The created network contains 92 nodes, and 209 edges.

6 RESULTS

6.1 *Variable network interpretations*

The network shown in Figure 2 provides interesting insights as to the structure of the design analysis tools. The network has been arranged as a force-directed graph, which aims to minimize the number of crossing edges and locates similar nodes closer together, and less similar nodes farther apart. Given this visualization the structure of the network reveals a number of variable clusters. Interestingly, these clusters correspond to their associated design discipline. The Louvain algorithm (Blondel 2008) is applied to identify communities, which are shown in different colors. Also notable is that the resulting variable network is indeed acyclic.

The standard modularity maximization approach confirms the communities apparent through the network visualization. The identified clusters are as follows: dark blue nodes represent powering and propeller design, light blue nodes represent overall weight and stability, green represents cost, red represent structural weight, and grey represents midship structural design. These clusters were expected, as they correspond to the five design tools considered in the case study. Interestingly, a sixth cluster emerges in the center of the graph, which is connected to all other clusters. This does not appear to coincide with any specific tool, and is shown in pink. Many of the variables in this cluster correspond to the hullform shape (L,B,T, Displacement, etc.) and classic naval architecture parameters. Not surprisingly, this cluster also contains a large proportion of the independent variables (nodes with in-degree of zero) as these entail the design requirements.

From this consideration a number of insights can be drawn about the way the design process is structured. First, the size of the communities (in terms of the number of variables within groups) provides insight to both the fidelity and complexity of the discipline's analysis method. Groups with larger numbers of variables are likely to be of higher fidelity, and those with a large number of edges are likely to be more complex tools. Additionally, the community structure suggests that the naval architecture specialty is central to the design activity, as it is the only community with connections to all other design disciplines. It also illustrates communities which are dissimilar—in this case the midship section analysis is not directly tied to the stability calculations, however they are related through structural weight and through the hullform parameters.

The directionality of these links is also important to consider. The naval architecture discipline has a large number of independent variables (variables with in-degree zero), and as a result, the central cluster contains a large number edges pointing to other clusters. This provides insight into the initial flow of information, as traditionally these are variables which must be decided early on in order to progress the design. The directionality of the flows provides context for the sequence in which design information is generated. The paths from independent variables through the network reveal the order in which decisions must be made in order to design the vessel. However, if these variables are fixed (through requirements for example), they will immensely reduce the design flexibility by limiting the number of negotiation pathways through the network. This quantifies a main advantage of SBD—by not fixing these variables in early design stages, information is able to flow much more readily through the network. This will be further explored in Section 7.

6.2 *Exploring hierarchical network structures*

While the detailed structure of the variable network provides valuable insight into the way design disciplines are connected, and the way the tools used predicate design spaces—additional insights can be gained through an analysis of the hierarchical network structure. The map equation was applied to the representative variable network using the *InfoMap* algorithm created by Edler & Rosvall, and is displayed in Figure 3.

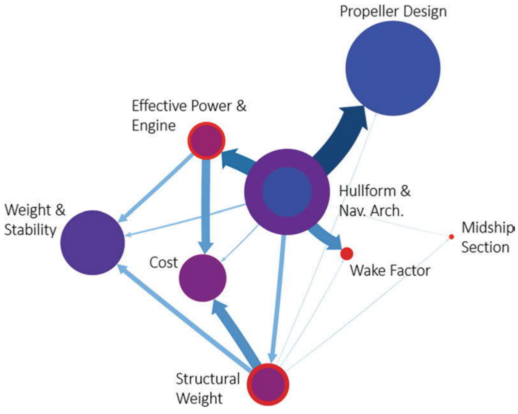

Figure 3. Hierarchical structure of the case study variable network using the *InfoMap* algorithm to identify design disciplines.

In Figure 3, nodes now represent identified design communities, and edges represent the flow of information. The node size corresponds to the time the random walk spent in the community, and the size of the edges represent the per step probability of moving between modules. Hence, the edges provide information between how likely information is to be transmitted on any given step of the design process between groups. The ratio of the inner to outer circles in each node represent the ratio of flow within and out of the community, respectively. Note that the node size of communities which are only pointed to (have no outflow of information) represent the inner circles of other nodes.

The communities identified by the map equation are very similar to those identified using the modularity maximization approach. The information-theoretic approach highlights the same broad communities, and elicits two additional communities. One community corresponds to the hullform wake factor, which arises from the high dependence on hullform parameters (as it is estimated using regression data) including Depth (D), which is important to the hull weight estimates. The other additional community arises from *InfoMap* separating the powering and propeller communities, which was merged in the modularity approach.

The edges in Figure 3 also provide valuable information about the flow of information between design communities. While the edges in Figure 2 represent variable dependencies, the edges in Figure 3 represent dependencies on design information. For example, the large flow from the naval architecture module to propeller design module suggests that the propeller discipline requires extensive information about the hull shape before the propeller can be designed. This dependency is intuitive and arises as a result of the point-based design structure, in which there exists a required sequence of analyses to be conducted to generate design information.

The directions of the flows reflect the central role the naval architecture discipline has on the other design areas, as all flows are directed out of the community. This means decisions made by the naval architect will have widespread impacts to all other design groups through the large amount of information flow. It also means that the majority of the negotiated design variables in this design approach will be naval architecture related—pertaining to the hullform. Conversely, the cost and stability communities have zero out-degree, which illustrate their analyses are predicated on decisions made by other groups. This is an interesting result, as stability and cost are disciplines which often emerge as design failures which are only obvious in late design stages (Cavas 2007, Shields 2017b). The emergence of design failures is a result of the number of dependencies on other groups' design decisions and the path by which they are generated as they require information from other groups before they can be calculated. They often are characterized by necessary rework, failure to create a converged design, or through increased design effort (Braha & Bar-Yam, 2007).

Perhaps the most insight to be gained from Figure 3 lies not in considering the existing edges, but those which do not exist. The absence of an edge between two groups which are seemingly closely related suggests a communication pathway does not exist, when it likely should. For example, consider the propeller and powering modules—these are highly interrelated design efforts in practice—engine selection and propeller optimization are highly dependent on one another. However, given the structure of the variable network, they are primarily related through the ship's speed (V), which is an independent variable (with in-degree zero) and resides within the naval architecture module. Given the directions of the flows, the two groups' analyses are both predicated on information transmitted from the naval architecture group. This type of insight can be used to reveal the underlying sequence of the design process, but could also be extended to analyze the effects of fixing requirements (in the form of variables) on the efficiency and rate of information transfer between groups.

7 IMPLICATIONS FOR SET-BASED DESIGN

The variable network outlined in Section 3 and the associated hierarchical representation outlined in Section 4 have presented the framework of studying a design activity. The case study presented and analyzed in Sections 5–6 have presented the types of insights which can be gained from such an analysis. However, the question still remains: what implications does this framework have on structuring a SBD design activity?

The first step in the SBD process, identified by Sobek et al. (1999), is mapping the design spaces. This is conducted to define feasibility, explore tradeoffs, and communicate sets of possibilities. Determining the design spaces from the outset of a project is integral to the success of the design activity. Herein lies many of the key contributions of the variable network approach and hierarchical structure analyses. The key contributions of the framework as they pertain to SBD are summarized below. The proposed framework:

- Maps initial design spaces, and generates information about the design process.
- Utilizes information flows to quantify design spaces and communication pathways between groups to make informed decisions about structuring design disciplines.
- Evaluates the structure of the design tools and processes independent of the implemented medium.
- Enables design tools to be better aligned with the selected design approach.

The variable network lends itself to a number of key insights which enable SBD. Clearly the network enhances the transparency of design tools, and how that fits into the larger design process. Illustrating functional dependencies between variables from the outset of the design activity enables designers to understand the design approach and understand tradeoffs. Additionally, the size of the resulting variable network provides context as to the size, scope, and complexity of the design activity. The network decouples the structure of the design problem from the medium through which it is implemented. This reduces the confusion between a design tool and design method, which is a key role in reducing designer understanding (McKenney 2013). The method aims to better incorporate the decision maker into the design process, and reduce overreliance on design tools, by opening the 'black box'.

The generation of the variable network can be conducted in parallel, which enables different functional groups to work concurrently. By doing so, it also enables functional groups to uncover potentially unknown interdependencies between variables which are crucial considerations from a SBD perspective. These unknown interdependencies are crucial to identify early in the SBD process, to ensure adequate communication pathways are opened for set negotiations. These interdependencies can exist as resident pathogens in the design system or design process, and can lead to emergent failures over time

(Leveson 2004, Perrow 2011). Identifying these interdependencies early on assist in front-loading the design process to identify issues when there is maximal design freedom, which is directly aligned with the SBD mantra.

By analyzing the hierarchical structure of the variable network, inter-variable dependencies can be used to determine inter-group dependencies. This scalability, based on the optimal flow of information, further bridges the gap between the structure of design tools to the design approach. This provides insight as to how design tools can be potentially re-aligned to better enable SBD, by classifying variables into design spaces based on the network structure. For example, variables (nodes) only connected to other nodes within their own group represent *internal design spaces*. Alternatively, variables pointing to nodes in different groups, but only pointed to by nodes within their cluster would be classified as *broadcast spaces*. Finally, *negotiated spaces* arise from nodes with zero in-degree (independent variables) which are those which must be negotiated to generate solutions. These classifications of design spaces would be relatively straightforward to determine from the network based on structure alone and are directly tailored from the design tools and approach being used. This is crucial as novel design activities will have unique and distinct design spaces dependent on both the nature of the design activity and the tools being utilized, and the method incorporates both.

In addition to classifying variables into design spaces, the network can be used to evaluate communication pathways between functional groups using the network's implicit hierarchical structure. Intra-group connections in the variable network, such as those shown in Figure 2, provide context about specific variable negotiations, while connections in the hierarchical structure (Figure 3) provide a higher-level context about the quantity and directionality of information flows between groups. The process of viewing the macro-level information flows between groups enables missing communication pathways to be identified, and enables the variable network to be restructured to solve these issues. The impact of changing the relationship between variables is immediately measurable by observing the way the hierarchical structure of the network changes. Restructuring the variable network to manage communication pathways between groups can be used to tailor design tools to better enable SBD.

The method is extendable to elicit information about potentially problematic areas of a design approach. A number of issues associated with structuring a SBD activity identified by Singer (2003) include: clearly defined design variables, balancing of workloads across design disciplines, and agent information overload. Many of these issues can be addressed using the variable network framework and resulting hierarchical structure. By clearly defining variables in a mathematical sense, the subjectivity of variables is eliminated. Analyzing optimal information flows in the hierarchical variable network provides insight to the balance of design tool utilization, which can be used to better balance workflows. These information flows can also be analyzed from a variable perspective to provide insight into which variables or groups are exhibiting information overload.

Information overload may be identified through information transmission metrics, and by determining variables with edges to a large number of disciplines. Highly connected and congested nodes may suggest that variable refinement is required, which may be able to adjust the tool to be better utilized in the SBD framework. As an example, consider the representative variable networks shown in Figure 4. In the network on the left, total weight (W_{TOT}) is expressed as a function of machinery weight (W_M), outfitting weight (W_O), structural weight (W_{STR}) and deadweight (W_{DWT}). In this case, W_{TOT} has edges to three other design disciplines (*D1*, *D2*, and *D3*). If information overload were an issue at this node, variable refinement may be applied to restructure the communication pathways. The network on the right illustrates variable refinement whereby an additional intermediate variable representing lightship weight (W_{LS}) has been added. Note that both cases would yield the same answer given the same inputs, yet the networks have different structures. In this example, the addition of the W_{LS} variable reduces the information overload in W_{TOT} by enabling negotiations with *D1* through W_{LS}. In this example, this would be a relatively minor change to an implemented tool, yet this restructuring process could present significant challenges in more complicated tools.

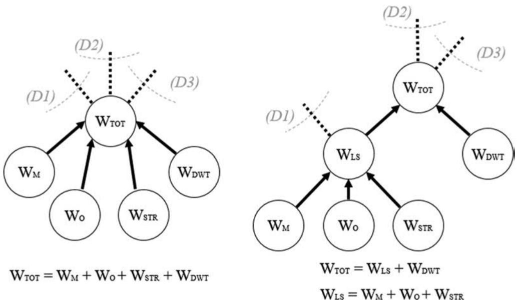

Figure 4. Two variable network structures representing the same function, with different communication pathways for negotiated design spaces.

8 CONCLUSIONS

This paper presents a method for determining the structure of a design approach simply by enumerating the variables considered in a design effort, and by tracking their dependencies through a set of design tools. The structure of the resulting variable network is not dependent on the actual form of the variable functions, nor is it dependent on the medium through which the analysis is conducted. The network provides a number of insights to a design effort. First, community detection can be used to determine functional groups (the design tools) without defining them a-priori, and allows for additional groups to emerge (such as in the case of the naval architecture group) which may not exhibit themselves through other design tools. By considering the hierarchical network structure, the flow of information within and between functional groups can be determined, which not only highlights inter-discipline dependencies, but quantifies the amount of information flow required between groups. This information provides insight to address potential issues earlier in the design process, as well as to identify the applicability of existing design tools to be integrated into the SBD framework.

The presented case study highlights the importance of the role of the naval architect in the design activity. In the traditional point-based approach, the decisions that the naval architect makes early on has ramifications in all other design disciplines, and allows for emergent design failures in groups with no out-degree. If these tools were to be utilized in a SBD framework, the connectivity of the naval architecture discipline to all other design disciplines shifts its role from not only generating design information, but also to acting as an integration manager. Given the structure of the considered variable network, the naval architect would be responsible for both determining variables and enabling the negotiations between a large number of design disciplines. The absence of communication pathways between seemingly related fields may help tailor the approach of the design tools to be more effective in relation to negotiated design spaces and SBD overall.

Significant research remains to be completed in this domain. Further work includes:

- Determining ideal information-flow structures of design disciplines in SBD.
- Studying dynamic network generation over the course of a design to incorporate increasing information fidelity.
- Time dependent analyses of information flows and routings, expanded to consider decisions made and rates of design convergence.
- Expanding the *InfoMap* analysis to allow partial variable memberships across different groups.
- Considering weighted edges between variables.
- Determining the impact of fixed requirements on design flexibility, information flow, and time to design.

While the case study considered a relatively simple ship design, the authors believe the method is scalable to incorporate more complicated designs, and flexible enough to be applied across a wide range of industries and engineering design efforts outside of the naval realm.

ACKNOWLEDGEMENTS

We would like to thank Kelly Cooper from the Office of Naval Research (ONR) for providing support for this project. This work was funded under grant number N00014-15-1-2752. We would also like to thank Michael Sypniewski for his insightful comments and helpful review of the paper.

REFERENCES

Bernstein, Joshua I. (1998). Design Methods in the Aerospace Industry: Looking for Evidence of Set-Based Practices. *Master of Science Thesis, Massachusetts Institute of Technology.*

Blondel, V.D., Guillaume, J.L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment, 2008(10), P10008.*

Braha, D., and Y. Bar-Yam (2007), The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. *Management Science, 53 (7), 1127–1145.*

Cavas, Christopher P. (2007). Is New U.S. Destroyer Unstable? Experts Doubt Radical Hull; Navy Says All Is Well. Retrieved from https://www.defensenews.com/naval/2015/12/03/instability-questions-about-zumwalt-destroyer-are-nothing-new/.

Cuneo, B.J. (2013). Development of a Human Centric Multidisciplinary Design Optimization Method Using Fuzzy Logic Systems and Controllers. *Ph.D. Dissertation. University of Michigan.*

Doerry, N., & Steding, M. (2009). Ship Design Manager Manual. *Naval Sea Systems Command.*

Doerry, N. & Fireman, H. (2009), Fleet Capabilities-Based Assessment. *Naval Engineers Journal, 121: 107–116.*

Eccles, T.J. (2010). Ship Design and Analysis Tool Goals *(COMNAVSEA memo 9000 Ser 05T/015). Washington, D.C.*

Edler, D & Rosvall, M. The MapEquation software package, available online at http://www.mapequation.org.

Evans, J.H. (1959). Basic Design Concepts. *Naval Engineers Journal, Vol. 21, Nov.*

Gray, A.W. (2011). Enhancement of Set-Based Design Practices Via Introduction of Uncertainty Through the Use of Interval Type-2 Modeling and General Type-2 Fuzzy Logic Agent Based Methods. *Ph.D. Dissertation. University of* Michigan.

Holtrop, J. (1984). A statistical re-analysis of resistance and propulsion data. *Int Shipbuild Prog, 31, 272–276.*

Holtrop, J., & Mennen, G.G. (1982). An approximate power prediction method.

Huffman, D.A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE, 40(9), 1098–1101.*

Keane, R., Fireman, H. & Billingsley, D. (2006), Leading a Sea Change in Naval Ship Design: Toward Collaborative Product Development. *2005 SNAME Maritime Technology Conference and Expo and Ship Production Symposium, Houston, TX*

Leveson, N. (2004). A New Accident Model for Engineering Safer Systems. *Safety Science (42).*

Liker, J.K., Sobek, D.K., Ward, A.C., & Cristiano, J.J. (1996). Involving Suppliers in Product Development in the United States and Japan: Evidence for Set-Based Concurrent Engineering. *IEEE Transactions on Engineering Management, 43(2), pp. 165–178.*

McKenney, T.A., Buckley, M.E., & Singer, D.J. (2012). The Practical Case for Set-Based Design in Naval Architecture. *Paper presented at the International Marine Design Conference, Glasgow, UK.*

McKenney, T.A. (2013). An Early-Stage Set-Based Design Reduction Decision Support Framework Utilizing Design Space Mapping and a Graph Theoretic Markov Decision Process Formulation. *PhD thesis, University of Michigan.*

Mebane, W.L., Carlson, C.M., Dowd, C., Singer, D.J., & Buckley, M.E. (2011). Set-based design and the ship to shore connector. *Naval Engineers Journal, 123(3), 79–92.*

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

Oosterveld, M.W.C., & Oossanen, P.V. (1975). Further computer-analyzed data of the Wageningen B-screw series. *Int Shipbuild Progr 22, 251–62.*

Parker, M.C. & Singer, D.J. (2015). Analyzing the dynamic behavior of marine design tools using network theory. *Ocean Engineering 106, 227–237.*

Parker, M.C. (2014). A Contextual Multipartite Network Approach to Comprehending the Structure of Naval Design. *Ph.D. thesis, University of Michigan.*

Parsons, M.G. (1975). Optimization methods for use in computer-aided ship design *(No. 13).*

Parsons, M.G., Singer, D.J. & Sauter, J.A. (1999), A Hybrid Agent Approach for Set Based Conceptual Design. *10th International Conference on Computer Applications in Shipbuilding (ICCAS), Cambridge, Vol. 2, pp.207–221.*

Perrow, C. (2011). Normal accidents: Living with high risk technologies. *Princeton university press.*

Rosvall, M., & Bergstrom, C.T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences, 105(4), 1118–1123.*

Rosvall, M., Axelsson, D., & Bergstrom, C.T. (2009). The map equation. *Eur. Phys. J. Special Topics, 178, 13–23.*

Sullivan, P.E. (2008). Ship Design and Analysis Tool Goals *(COMNAVSEA memo 9000 Ser 05D/047). Washington, D.C.*

Shannon, C.E. (1948), A Mathematical Theory of Communication. *Bell System Technical Journal, 27: 379–423.*

Shields, C.P.F. & Singer, D.J. (2017a). Naval Design, Knowledge-Based Complexity, and Emergent Design Failures. *Naval Engineers Journal, 129(4), 75–86.*

Shields, C.P.F. (2017b). Investigating Emergent Design Failures Using a Knowledge-Action-Decision Framework. *Ph.D. Thesis. University of Michigan.*

Singer, D.J., Doerry, N., & Buckley, M.E. (2009). What Is Set-Based Design? *Naval Engineers Journal, 121(4), 31–43.*

Singer, D.J. (2003), A Hybrid Agent Approach for Set-Based Conceptual Ship Design through the Use of a Fuzzy Logic Agent to Facilitate Communications and Negotiation. *Ph.D. Dissertation. University of Michigan, 2003.*

Sobek, D (1997). Principles that Shape Product Development Systems: A Toyota-Chrysler Comparison. *PhD thesis, University of Michigan.*

Sobek, D. & Ward, A.C. & Liker, J. (1999). Toyota's Principles of Set-Based Concurrent Engineering. *Sloan Management Review. 40.*

Sypniewski, M.J. & Singer, D.J. (2017). A Framework for Asessing the Effects of Bias in Early-Stage Ship Design. *International Conference on Computer Applications in Shipbuilding (ICCAS). Singapore.*

Ward, A.C., Liker, J.K., Christiano, J.J., & Sobek, D.K. (1995). The Second Toyota Paradox: How Delaying Decisions Can Make Better Cars Faster. *Sloan Management Review, 36(2), pp. 43–61.*