Using and Interpreting the Bayesian Optimization Algorithm to Improve Early Stage Design of Marine Structures.

by

Thomas E. Devine

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
2016

Doctoral Committee:

Assistant Professor Matthew D. Collette, Chair
Professor Charles R. Doering
Assistant Professor David J. Singer
Professor Armin W. Troesch

For the loving family and friends who have always believed in me

ACKNOWLEDGMENTS

First and foremost I must thank my chair and advisor, Dr. Matthew Collete. I remain incredibly fortunate to have worked with him these past 4 and a half years. He has allowed me to grow both as a professional and as a person and his guidance, leadership, and mentoring will always be appreciated. His steady hand has led to some of the most positive experiences of my life and laid the foundation for my future.

Beyond Matt, my entire committee, Dr. David Singer, Dr. Armin Troesch and Dr. Charles Doering have enhanced my education and guided me through this process and I remain indebted to them. My time at the University of Michigan has allowed me to develop skills I'll use for the rest of my life I will always cherish the education I received here. I must thank my fellow Ph.D. Students and labmates, Dr. Jiandao Zhu, Dr. Dylan Temple, Dr. Yan Liu, Dr. Mark Groden, and Lauren Claus. You have helped me to produce my best work, acting as the best sounding boards and critics a researcher and writer could hope for. I would also like to thank the entire staff and faculty in the NA&ME for their patience understanding and encouragement as well.

Before Michigan I had excellent mentors at my previous schools, Virginia Tech and MAST, as well as in my professional societies SNAME and ASNE, all of who led me to this opportunity. Friends, teachers peers and colleagues there have always provided quality advice as well as provoking challenges to help me improve myself. I have so many valuable and unique experiences as a result of these groups and i will cherish the memories and lessons learned from them.

Lastly I must thank my family. I can't imagine my parents' private reaction a week after leaving NJ, hearing I was planning on spending 3+ years in the midwest. They provided unfailing love and support, seeking only my best interest throughout the entire process. My brothers have been a steadfast source of support and lightheartedness when times here got tough or trying, and I'll always love them. This document and degree is the result of so many individuals endevouring to help me and I remain indebted to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Using and Interpreting the Bayesian Optimization Algorithm to Improve Early Stage Design of Marine Structures.

by

Thomas E. Devine

Chair: Matthew Collette

Early stage naval structural design continues to advance as designers seek to improve the quality and speed of the design process. The early stages of design produce preliminary dimensions or scantlings which control the cost and structural performance of a vessel. Increased complexity in the evaluation of structural response has led to a need for efficient algorithms well suited to solving structural design specific optimization problems. As problem sizes increase, existing optimizers can become slow or inaccurate. The Bayesian Optimization Algorithm (BOA) is presented as one solution to efficiently solve problems in the structural design optimization process.

The Bayesian optimization algorithm is an Estimation of Distribution (EDA) algorithm that uses a statistical sample of potential design solutions to create and train a Bayesian network (BN). The application of BNs is well suited for nearly decomposable problem composition which closely matches rules based structural design evaluation. This makes the BOA well suited to solve complex early stage structural optimization problems.

Additionally, the learning processes used to create and train the BNs can be analyzed and interpreted to capture design knowledge. This return of knowledge to the

designer helps to improve designer intuition and model synthesis in the face of more complex and intricate models. The BNs are thus analyzed to augment design problem understanding and explore trade-offs within the design space. The result matches a paradigm shift in early stage optimization of naval structures. Designers gain better understanding of critical design variables and their interactions as compared to the previous focus on the single most optimal solution. This leads to efficient simulations which rapidly explore design spaces, document critical design variable relationships and enable the designer to create better early stage design solutions.

CHAPTER 1

Introduction

1.1 Research Overview

With the size scale and complexity of modern ships, design and optimization of ship specific systems continue to be interesting and challenging fields of research. One particular sub-system, marine structures, has advanced rapidly through the application of computational simulation in the analysis and evaluation of structural response. Larger, more complex, high fidelity simulations more accurately model structural response to allow the designer to test a wider variety of designs. These approaches have begun to strain the limits of computational simulations, becoming more computationally expensive. This is particularly cumbersome during early stage design where primary scantlings are subject to uncertainty and experimentation. Without efficient design, these new evaluation tools can become prohibitive.

This improved capacity and accuracy also continues to challenge designers, particularly when making design selection decisions. The increased complexity of designs has obscured some of the designer knowledge that remains critically important when making tradeoff decisions. These tradeoff decisions largely control the performance and the cost of the vessel, and are made at an early stage, locking in these characteristics. Good decisions generate good designs, while poor decisions can plague a design as it matures through the design cycle. Designers thus need tools to better understand the complex design spaces they are exploring and making decisions in. These tools should be simplistic, allowing for interpretation and implementation of change based on observations and conclusions.

To respond to this challenge, an improved optimization approach is needed. This approach should efficiently utilize structural fitness function evaluations such that these computationally extensive simulations are smartly used. At the same time, a tool that also captures, collates, and presents unknown design space relationships in a simple and understandable manner can help to counter the loss of designer intuition in these complex spaces.

The problem marine structural design is well described as a nearly decomposable design problem when evaluated through rule based strength constraints. Evaluation of strength constraints largely relies on small subsets of the total design variable vector, and these subsets are often conditionally related. This format presents opportunities to leverage specific, heuristic processes to solve the problem more specifically. In particular the conditional dependencies associated with Bayesian networks (BNs) closely match this structure. For this reason the Bayesian Optimization Algorithm (BOA) was hypothesized to be highly efficient when solving marine structural design problems. The associated networks allow the optimizer to quickly solve the local structural constraints in a rule based approach faster than existing heuristic optimizers, thus requiring less total fitness function evaluations.

While the BOA conducts optimization, it creates models of the design variable interaction, and stores this knowledge in the BNs. The structure of these networks may then be analyzed to determine if the network edges and conditional probability tables may be correlated to the design space. If this is true, the question then becomes: can the network be analyzed to produce simple digestible relationships to make designers better informed? Through the use of simple network metrics, the BNs are analyzed to create concrete design relationships. These relationships are then related to the problem fitness function. This improves the designer understanding of the design space and ultimately delivers a better design.

1.2 Research Contributions

This work produced specific novel contributions in the field of early stage marine structural design and optimization:

1. The existing nonlinear relationship between problem size and required number of fitness function evaluations is explored and bounded. This demonstrates the need for improved optimization approaches and introduces the potential impact of the BOA.

2. The BOA is adapted and applied to a large and complex engineering specific problem. The algorithm is improved moving from binary representations of variables to discrete representations. The improved algorithm successfully solves this difficult case study.

3. After successfully solving this initial problem, the algorithm's efficiency is compared to the existing state-of-the-art algorithm, highlighting the differences between the heuristic approaches and the resulting responses of the optimization simulations.

4. The new implementation of the algorithm presents discrete bin values on the variables, allowing the networks to be analyzed for design relationships. This analysis was used in conjunction with the known objective function to explore and document the relationships formed by the BOA.

5. The knowledge gathered through network analysis is then leveraged in two case study demonstrations. These simulations show how the tool was used to improve the results of early stage designs.

CHAPTER 2

Background

2.1 Overview

This chapter seeks to explore the modern marine structural design problem, and the use of optimization to solve it. This review will bound the present scope of work done, and point out potential spaces for advancement in key areas, primarily in the efficiency of the optimization process. Based on the presented classification of the design problem type, the Bayesian Optimization Algorithm is introduced, and its previous application is discussed. The BOA is then presented as a potential solution to efficiently solve nearly decomposable structural problems.

2.1.1 Naval Design and Optimization

Modern naval vessel are spectacles of engineering, inspiring audiences with their capability and complexity. The science and engineering applied to these massive designs continue to push the boundaries of feasibility and understanding, seeking to add effectiveness while mitigating risk, as noted by Peri and Campana (2003). This pure engineering work is then exposed to modern political and socio-economic climates, forcing cost to become a major limiting factor in ship design. The result is a highly complex, continuously evolving problem with external design pressures. The most famous modern example of this is the DDG 1000, a US naval vessel designed as the

Figure 2.1: DDG 51 at left and DDG 1000 at right Crucchiola (2015) Reilly (2003)

Principle Characteristics	DDG 51	DDG 1000
Length(m)	154.0	182.9
Beam (m)	20.0	24.6
Draught (m)	9.4	8.4
Displacement (metric tons)	6900	14,798

Table 2.1: Principle Characteristics of the DDG 51 and its replacement DDG 1000. Growth in a majority of categories are a small indication of the massive leaps in complexity of modern ship design.

next generation Destroyer. In table 2.1, the principle characteristics are compared between the DDG 1000 and its legacy parent, the DDG 51 are shown. Most of the physical dimensions demonstrate growth, and in navy ships which are typically both weight and volume limited, this speaks to the increase in design complexity across a variety of ship systems. Internal systems also took massive leaps forward as well, replacing aging systems for state of the art systems. Even the external appearance of the vessels demonstrate the extraordinary changes that occurred between the two ship classes. Negotiating these changes falls to the designer, and as described by a report to congress in O'Rourke (2009) this is a nontrivial task connected to billions of dollars. The ship design process is often described as "Wicked'" by some naval architectsAndrews (2012). Wicked problems as first described in Rittel and Webber (1973) have 10 unique challenges that require designers to negotiate unusual or difficult problems. Often this requires designers to fill gaps caused by incomplete information with modeling and assumptions. This difficulty is compounded by the scale of the product, making large scale modeling and prototyping infeasbile. Instead, engineering models and analysis provide the lion's share of design space information for designers to use. As a result, ship design has historically been product oriented or driven, with a single point solution, or multiple competing objectives being the focus and output of the design process. Through repeated approaches to this process a common knowledge-base is developed consisting of appropriate or ideal parameters governing equations and modeling techniques. This knowledge base is then called upon to solve future design problems, both extrapolating or interpolating, where appropriate, to best estimate effective solutions. The failure of this approach is that it is inherently reactive.

In other words, what can the designer do when the proposed design solution falls outside the established conventional knowledge set? By relying on standardized techniques and assumptions, design configurations and solutions that exist sufficiently outside conventional design envelopes become risky. The set of experiences, models, and designs leading to the conclusions about the design space may not describe the new portion of the design space. The response of the designers is to apply tools to the problem to probe, explore, and understand the design space such that it may be mapped and utilized in the design process. This creates a pattern, where designers, when encountering the scenario, test the design space with the new information and record the results to be added to the knowledge base. This is most frequently done through optimization, as designers are concerned with solutions that are 'good'.

The problem with this approach is the tedious nature of the cycle. Every time a new space needs to be explored, the design must calibrate an optimizer and run it. From these simulations, the designer, over time and repetition, builds understanding of the solution space. The intense computational effort is required, with hundreds of potential nuances to be learned. Also as the models become more and more complex, the computational requirements grow, making computer simulation potentially intractable. Additionally, research from Lewis (2012) describes the the problem of design synthesis for large and complex problems, similar to ship design. They highlight the need for simplicity in responce to complexity: *Complexity and simplicity should co-exist in the design of large-scale engineering systems. Their elegant interplay will better allow engineers and managers to design, develop, and manage such systems.* At a certain point the human brain struggles to grasp the entirety of the design and in particular higher order effects that may not be explicitly clear focusing on representation of artifact in design. Dym (1994) These consequences from naval design process evolution require innovation and a potential paradigm shift to enable designers to again produce high quality solutions while creating simple representations of designs.

In light of this complex problem, new approaches to design of various ship systems should seek to accomplish 3 major things:

1. **Optimization approaches should take advantage of problem type and solution modeling to most efficiently explore the design space and locate design solutions.** In the design of each ship system, ranging from its hydrodynamics to its structural scantlings to its power generation and distribution systems, varying physics based models are created to model the phenomena occurring. The model and disciplines have forms which may be best exploited for efficiency.

2. **Due to modern levels of complexity, and massive generation of data, design and optimization processes should 'learn' and create simplistic relationships to aide designer understanding.** To make designers more effective any model simulation or process should seek to utilize data created and capture lessons from it. If this data can then be analyzed and presented to designers in an understandable or digestable manner, designers become that much more capable.

3. **The designer, armed with new knowledge and insights into trade-offs and relationships within the design space should then be able to affect the design space to confirm potential hypothesizes and lean optimizations to highly fit and robust solutions** The changes to the design and optimization process are ultimately done for the improvement of designer understanding. The final step in a new optimization paradigm is the re-insertion of designer intent, whether it be to explore contingencies, confirm theories, or test 'what-if' scenarios. This ability to experiment and improve has been noted in social sciences to help child comprehension when designing to learn. Hmelo et al. (2000) This feedback loop improves the quality of the design and prevents designers from falling into the old paradigm which was much more reactive.

With these changes, the designer goes from a reactive position, where they potentially lack understanding or intuition to a proactive position, where lessons learned from optimization simulations are absorbed and may be applied tested or improved. The bulk of changes needed to accomplish this paradigm shift exist in the optimization process, and with a clearer understanding these changes can be addressed.

2.1.2 Optimization techniques

As previously discussed, optimization has become a foundation of design exploration evaluation and selection. All optimizations are founded around 3 simple parts, a design variable vector, $\mathbf{X}$, which contains the pertinent design variables the designer seeks to explore and/or select, an objective function vector $f(\mathbf{X})$ which maps the design variable from its local space to one or more solution spaces as specified by the system model(s), and a constraint function vector $g(\mathbf{X})$ which enforces model constraints on the space (in either design variable space or objective function space). The optimizer, though some mechanism autonomously transforms the design variable vector seeking to improve the resulting objective function value until a specified convergence criteria is reached. For a more comprehensive review of basic optimization see Deb (2004).

Approaches to solve design problems are quite numerous, and these approaches are often tailor made to problem specific structures such that the design algorithms are incredibly efficient, robust and/or accurate. The largest divergence in the approach to optimization is the method in which changes to the design variable vector are made. The largest and most developed branch of which is gradient based optimization. Snyman (2005) When using a gradient based optimizer, a mathematical gradient is used to inform the change in design variable vector. In concept this leads along the path of steepest descent and with necessary and sufficient conditions guarantees the optimal function value is selected. In ship design this often appears in mathematically dominated systems such as hydrodynamic optimization where the model governing equations are smooth and well defined leading to an achievable gradient.

For other systems the form of the problem or design variable vector may be ill suited to obtain true or numerical gradients. Instead these optimizations are guided by heuristic approaches. Heuristic techniques trade potential accuracy or completeness for speed and efficiency. Where a positive definite hessian guarantees a globally optimal solution, the heuristic optimization can only assure a robust estimate, which for the early stage of design is often more than acceptable. Heuristic techniques can take on a wide variety of forms, as reviewed by Lee and El-Sharkawi (2008). They might mimic natural processes or impose a series of logic queries to a design problem, often probing with a population of candidate design pools. Heuristic techniques often have incredible search power as the robustness needed to ensure a highly fit solution drives the optimizer to search vast swaths of the design space. It is a shame that few heuristic optimizers create models of such spaces while searching, as these models would be critical tools to reintroducing design knowledge to the designer. In particular some networks show promise to combine a large amount of data from the populations and present the information in an easily digested manner. For this reason Bayesian networks may potentially be adapted to accomplish the design process innovation needed for naval design.

2.1.3 Bayesian Networks

A Bayesian network is a graphical method of representing data. The graph G is defined by a vector of nodes B and edges E. It is widely popular in multiple applications particularly in the fields of decision making Nielsen and JENSEN (2009) and modeling uncertainty Daly et al. (2011). Newman (2010) gives excellent instruction on the use of networks in modeling.

To introduce the concept, a simple example is presented. Consider for a moment the probability of a student receiving a letter grade A-F on an exam. Without any further information one might assume the probability follows a normal distribution centered around a mean score, as many professors prefer to create their test to fit this profile. So without any information, a probabilistic assumption may be made regarding the students likely grade. At the same time consider the probability a student spent N hrs studying for the exam. Again with potential modeling and parameter assumptions, a distribution may be created to fit student study habits. Finally consider the number of hours a student sleeps per night in the same manner, with a distribution modeling college student sleep pattern. In this scenario, 3 nodes of a simple Bayesian network have been created. The nodes must also be connectes through a series of edges to show conditional dependence.

With the 3 nodes, test grade, hours studied and hours of sleep, there are natural ways to associate these quantities. From experience, the student that studies more is more likely to score higher, and likewise one who fails to study sufficiently is more likely to receive a poor grade. Similarly, a well rested student is likely to perform better than a sleep deprived student. The test grade is said to be conditionally dependent on both hours studied and on hours of sleep. This relationship implies that edges exist between these nodes in the network. To summarize this graphically:

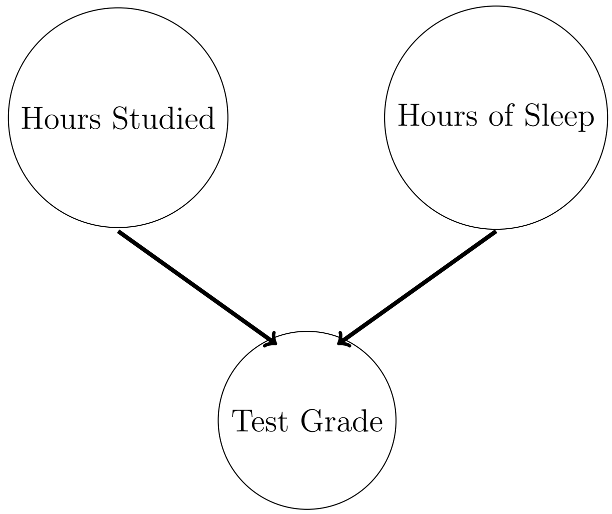


Figure 2.2: Using Student test grades as a simplistic example, a basic Bayesian network is described and constructed

To abstract these concepts any quantity may be represented by a node. Nodes are connected by a series of directed edges or arcs. These arcs indicate the flow of condition dependence, as governed by Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

A and B correspond to nodes within the Bayesian network. By using evidence about nodes A and B and an assumed relationship between the two, state probabilities can be statistically determined. Nodes in a Bayesian network differ from other types of networks in that each node has an associated Conditional Probability Table (CPT). These CPTs govern the statistical distribution of the various nodes within the network.

These features of a Bayesian network supply tremendous statistical power. This power can be used in multiple ways, either using statistical samples to train and create the network, or using the network and distributions to create a sample matching

Sample Point	**A**	**B**	**C**	**D**
1	1	1	0	1
2	0	1	0	1
3	1	1	1	1
4	1	0	1	0
5	0	1	0	0
6	0	0	1	1
7	0	1	0	1
8	1	0	0	0
9	0	0	0	1
10	0	1	1	1

Table 2.2: Sample binary population for 4 noded example network in figure 2.3

the model. Suppose that for the given network in figure 2.3, a data set of population members was given in table 2.2. CPTs for nodes A, B, C, and D (table 2.3) would then match the statistical sample. Should this sample be deleted or discarded, through

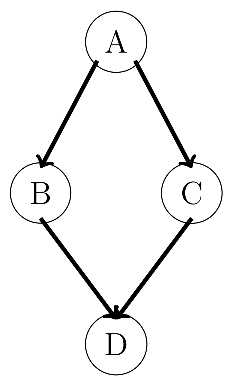


Figure 2.3: Example Bayesian network with nodes A,B,C, and D and edges AB,AC,BD,and CD

enough statistical sampling of the trained network, a nearly identical sample could be replicated. This is incredibly powerful as the network encodes and stores information on a given sample population. This perfectly matches the heuristic techniques previously discussed, which expend a massive amount of computational resource simulating populations to control in optimization techniques. As noted, those heuristic techniques fail to capture the problem information provided by the samples, instead driving towards a singular solution set. A Bayesian network is a potential solution

A	
P(A=0)	P(A=1)
0.6	0.4

(a) Conditional Probability Table for node A

	B	
	P(B=0)	P(B=1)
P(A)=0	0.33	0.66
P(A)=1	0.5	0.5

(b) Conditional Probability Table for node B

	C	
	P(C=0)	P(C=1)
P(A)=0	0.66	0.33
P(A)=1	0.5	0.5

(c) Conditional Probability Table for node C

		D	
		P(D=0)	P(D=1)
P(B) = 0	P(C)=0	0.5	0.5
	P(C) = 1	0.5	0.5
P(B) = 1	P(C)=0	0.25	0.75
	P(C)=1	0	1

(d) Conditional Probability Table for node D

Table 2.3: Conditional Probability Tables for all nodes

to the knowledge capture step so critical for a new paradigm shift in design. This is what the BOA does, and is thus well suited for future optimization.

2.1.4 The Bayesian Optimization Algorithm (BOA)

The Bayesian Optimization Algorithm belongs to a class of heuristic optimizers called estimation of distribution algorithms(EDAs) Larraaga and Lozano (2002). These algorithms use probabilistic information about a population of individuals to attempt to select locations of highly fit candidate designs. The BOA uses a Bayesian network to capture store and refine these probabilities. Lima et al. (2011) makes specific reference to the BOA as a capable optimizer for problems that are decomposable in composition.

The algorithm itself has multiple distinct stages with varying mechanisms within each stage. The algorithm is looped generationally with the same steps occurring in every generation. Figure 2.4 shows pseudocode required to accomplish a basic optimization using the algorithm. In sections 2.1.4.1 to 2.1.4.4 the five critical operators required in each generation are discussed.

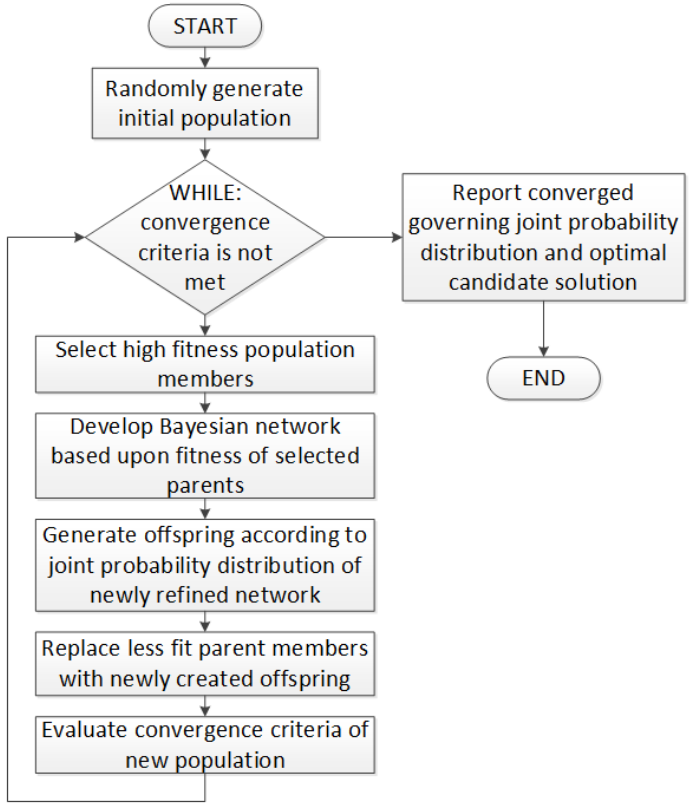


Figure 2.4: BOA Pseudocode Flowchart.

1. Selection
2. Network Structure Learning
3. Network Parameter Learning
4. Offspring Generation
5. Evaluation

As with many heuristic optimizers, there are a variety of operators that can be used in each stage to affect the performance of the BOA. Each section will discuss in more detail the theory and process in conducting a specific step of the BOA.

2.1.4.1 Selection

Most heuristic optimizers use a selection operator in place of a mathematical gradient used by gradient based solvers to guide the advancement of the optimizer. As mentioned, heuristic optimizers are varied, and thus the selection operators are widely varied. For heuristic optimizers using populations, like the BOA, PSOs, GAs, etc. the goal of the selection operator is to pick design points with high fitness and preserve them for use in replication/data training/propagation. The BOA requires a highly fit subset of the total population to be selected for data training of the network. To accomplish this many techniques exist, such as tournament selection, roulette wheel selection, ordered cuts, elitism advancement, or a combination of strategies. This operator will balance the tradeoff of the average fitness level with the diversity of the population selected for training. Though examined for the GA, Goldberg and Deb (1991) provides excellent insight into navigating this tradeoff. Upon completion of selection, the BOA will have isolated a training subset to be used in steps 2 and 3 of the generation optimization.

2.1.4.2 Network Structure Learning

Once the training data set has been determined, network structure must be learned. Part of the reason the BOA is so successful in optimizing nearly decomposable problems is the intelligent learning of a network structure to represent the associated problem and constraints. Each generation requires this learning step, as the training data changes from generation to generation. The increase in fitness is captured in the modifications made by the learning operator. Many approaches exist to modelling network structure, with intent often driving approach selection. Daly et al. (2011) provides a recent survey of popular choices of learning operators. All approaches seek to take the existing structure of the network and refine it to best fit the data used for training. The model may have pre-existing edges or it may simply have no edges at all. A machine learning approach is required to refine the network. For engineering optimization, the refinement of the network is in itself an optimization problem within an optimization problem. There are many ways of arranging network structure Singh and Valtorta (1995), with new methodologies continually created and modified Moore and Wong (2003). One of the most commonly applied approaches is a score based evaluation of the conditional independence of the installed edges. Among the scores, a Bayesian Dirichlet (BD) metric is simplistic and often used Cooper and Herskovits (1992). The BD metric evaluates the conditional independence of connection within a network, multiplying the product of observed data points and summing these products to produce a score.

$$\begin{aligned} BD(B) = P(B) \prod_{i=0}^{n-1} \prod_{\pi_{x_i}} & \frac{\Gamma(m'(\pi_{x_i}))}{\Gamma(m'(\pi_{x_i}) + m(\pi_{x_i}))} \\ & \prod_{x_i} \frac{\Gamma(m'(x_i, \pi_{x_i}) + m(x_i, \pi_{x_i}))}{\Gamma(m'(x_i, \pi_{x_i}))} \end{aligned} \tag{2.2}$$

The products run over all states of x_i and all states of the parents π_{x_i}. m denotes the observed number of data points within the set D for the given observed structure B. m' considers prior information of the data set statistics. Heckerman simplified the BD metric to create the K2 metricHeckerman et al. (1995).

$$K2(x_i, \pi_{x_i}) = \prod_{j=1}^{q_i} \frac{(r_{x_i} - 1)!}{(N_{ij} + r_{x_i} - 1)!} \prod_{k=1}^{r_{x_i}} \alpha_{ijk}! \tag{2.3}$$

q_i is the set of bins of the parent π_{x_i} being evaluated, and r_i is the set of bins of x_i being evaluated. N_{ij} is the number of observed data points, D with parent state j and α_{ijk} is the number of data points with parent state j and nodal state k. The K2 score is assumed to correlate to the goodness of the network representation, with a high K2 score more appropriately correlated to the given data set.

Multiple techniques exist to add edges to a graph such that the resulting network best represents a given data set. The easiest of which to implement is the local greedy algorithm (LGA). This algorithm arranges edges in a primitive fashion seeking to alter edges individually that make the greatest positive impact on the BD marginal likelihood of the data set. In this manner the optimizer can and often does lead the network refinement into globally sub-optimal representations. In fact determining the truly optimal representation is a computationally difficult task Chickering (1996), proven to be NP-complete.

The LGA operates with 3 essential edge manipulations:

1. Add Edge
2. Remove Edge
3. Reverse Edge

To control the complexity of the network and to limit computational time requirements, a maximum incoming edge limit is imposed on each node. The structure learning process terminates when the network no longer has a valid edge operation that improves the K2 score. In this way a termination criteria is established. Additionally, the graph must remain acyclic regardless of operation. Also, each node within the network is limited in the number of parents it may take, to control the computational expense of the algorithm search. Finally to end the local search, only positive BD metric increases are considered. Therefore the operator will terminate if the only operation will either undo the previous edge operation or will violate the two previously stated criteria.

2.1.4.3 Network Parameter Learning

Though the structure is critically important, the bin probability values of the CPTs will ultimately guide the generation of new points. Bins with high probability values will accordingly be selected more frequently than bins without. Once a structure has been selected, the discrete bin values are set to best represent the statistical sample of the potential designs. Heckerman (1998) provides excellent reference on the parameter learning process. The selection of data points as well as the statistical size of the sample are critically important to the quality of the Bayesian network in representing the design problem. An individual bin value for a given bin i of variable x may be stated as:

$$P(x_i) = \frac{\sum_1^n x_i}{\sum_1^n \pi_{x_i}} \tag{2.4}$$

This formulation will have the ability to set certain bin values to zero if the selected population subset contains no instances of a given bins variable representation. If the bin value is set to zero, it will never be further propagated in future generations. This may be desirable for certain implementations of the BOA, while others desire to maintain the ability to re-introduce the bin at a future point.

2.1.4.4 Population Generation

Once the Bayesian network has been re-learned to incorporate the most recent and elite data, the unselected training points from the previous generation must be discarded in favor of potentially more highly fit points. The selection of new points is accomplished using the joint distribution of the new network. Returning to figure 2.3 root to leaf propagation is used, selecting states of each node according to their local conditional distribution. In this figure node A is the root of the network, and its probability, $P(A)$ is known. For the other nodes, only the conditional probabilities are known, $P(B|A)$, $P(C|A)$ and $P(D|B,C)$. The final result is a population that follows the entire joint of the network. Statistically, this may also be described.

$$P(A,B,C,D) = P(D|B,C)P(C|A)P(B|A)P(A) \tag{2.5}$$

In this example, the state of node A will be selected first. This will allow the probability values of nodes B and C to be known explicitly. Nodes B and C will then be selected, allowing the final node, node D to be selected. In mathematical notation, the probability of node D may be expressed by:

$$P(D) = \frac{P(A, B, C, D)}{P(A, B, C)} \tag{2.6}$$

The algorithm will generate points to replace the discarded less fit solutions, restoring the candidate pool the the selected simulation population size.

2.1.5 Marine Structural Design

Initial use of computational optimization in the design of ship scantlings began with computationally efficient optimization algorithms Hughes et al. (1980). Designers recognized the potential for computer exploration of design spaces and the number of computational approaches exploded as documented in Hrnlein (1987). This increase in computational power led to a large body of work analyzing the response of structural elements to create methods of evaluation for design. While examining the residual strength of damage structures, Ghose et al. (1994) identifies three main types of failure, yielding, buckling, and fracture. It makes a demarcation between global and local failure of the structure, with strategies and tolerances for each level. A general approach of decomposing the structure into elements, and analyzing those more tractable elements was adopted. This led to the use of basic substructures, such as the stiffened panel or the fatigue detail, and promoted the understanding of failure modes effecting those substructures. Paik et al. (2005) provides an excellent overview of the various failure mode assessment methodologies for the stiffened panel, benchmarking across multiple engineering disciplines. Wirsching and Chen (1988) examines the effect of probabilistic safety approaches for welded fatigue details within ships. In both cases, characteristic substructures are identified and evaluated using simplified models.

At the same time, Designers continually clashed headlong into computational limits. As modern finite element analysis advanced, engineers devised problem specific approaches to leverage existing information and techniques. In one such scenario, existing designs were perturbed in response to design changes, rather than conduct full optimization(Kim and Bernitsas (1988)). A chapter within Kamat (1993) discussed the adaptation of problems to be used in linear programming, leveraging massive speed increases. As computational power progressed the techniques changed. Rigo (2001) probed least cost solutions in a design space using constrained nonlinear optimization to optimize midship scantlings. The authors noted: *Preliminary design is the most relevant and the least expensive time to modify design scantling and to compare different alternatives. Unfortunately, it is often too early for efficient use of many commercial software systems, such as FEM.* This is a critical observation: efficiency is king in early stage design. Clearly this portion of the design space allows the designers freedom to modify the design with lasting implications for the selected objective function.

The common thread throughout these developments is a need for speed. Engineers seemingly take one of two approaches, either adapt the design problem to use an existing, fast, robust optimizer, or create a new technique to take advantage of problem specific structure. In the field of aerospace Simpson et al. (2001) present a kriging model aimed to improve the computational speed of multidisciplinary optimization, targeting engineering design which relies heavily on complex computer analyses. The kriging model adapts the objective function to more quickly and inexpensively represent the design solution space. Engineers continued to tackle greater scope and more complexity in design problem formulations. As a result, the ship structural design community has expanded the use of heuristic optimization algorithms to solve complex structural design problems.

Heuristic approaches trade some of the mathematical rigor reserved for other optimization routines and as a result gain potential speed increases. Heuristic often mimic existing processes. Diez, Diez and Peri (2010), used a particle swarm optimizer to solve a robust optimization problem formulated for a bulk carrier. In the field of multi-disciplinary optimization (MDO) Peri, Peri D (2003), solved a complex hydrodynamic optimization using Genetic algorithms. Heuristic optimizers in particular have had numerous successes improving design objectives in the marine design field such as the genetic algorithm (GA) Sekulski (2010), Jang et al. (1996), Zhu et al. (2012) or the particle swarm optimizer(PSO) Pinto et al. (2007), Hart and Vlahopoulos (2010). Often these optimizers are applied where problems are computationally hard to solve, allowing designers to find optimal solutions where other methods might collapse.

With this in mind, existing potential gains may still be made in early stage approaches to structural optimization. Speed remains a critical aspect, and with the popularization of parallel computing approaches, speed becomes difficult to measure. Ultimately the bottleneck is the optimization process is the evaluaton of designs with high quality methods. Niche design optimizers, as described by Fieldsend (2013), continue to be developed to address problem specific intricacies that allow the optimizer to be more efficient at solving particular problems. New approaches to efficient early stage design need to reduce the number of fitness function evaluations needed to reach high quality optimal solutions.

By identifying marine structural design as a niche, optimizers may be selected and improved to best solve problems within the niche. For example, Klanac and Jelovica (2009) showed excellent results adapting the NSGA-II algorithm proposed by Deb et al. (2002) and applying its vectorized form to marine structural design. Ehlers (2012) matched the rapid objective function improvement of a PSO to computationally expensive nonlinear FEA to optimize high strength steel structures in LNG carriers. Clearly incremental gains will continue to be made in efficient optimization of niche specific problem. Therefore, a description of the topology that characterizes structural optimization, using failure mode evaluation should be used to match optimizers to niches and then exploited.

An understanding of the localized response of substructure elements suggests there is an ordered approach that can be taken to the optimization of marine structures. Marine structural design problems may be described as nearly-decomposable in composition due to the evaluation of failure modes. Decomposability is a measure of the coupling or interactions between design variables in the evaluation of the design. A highly decomposable design may effectively be split into many smaller parts. Each individual part of the design has limited interaction with other parts. The smaller parts are easier to solve and optimizers that exploit this to gain advantage will likely perform well when solving problems. The work done to examine substructure evaluation matches this pattern. Failure modes of scantling elements suggests high decomposability in the evaluation of structures. However in juxtaposition to a true decomposable problem, marine structural evaluation retains a degree of non-decomposability. Certain evaluations require the entire design variable vector representing the considered scantling set. Therefore a nearly decomposable design problem shares characteristics with both topologies. A majority of the design evaluations are reducible in required variable set size. A small number of evaluations however require the entire design variable vector to determine adequacy.

The Bayesian Optimization Algorithm became interesting as an optimizer because it was used to efficiently solve purely decomposable problems. Pelikan et al. (2002) demonstrated that for an entirely decomposable problem, the BOA required fewer function evaluations than those of a comparable Single Objective Genetic Algorithm (SOGA) at large problem sizes. This conclusion suggests that the BOA should also be an ideal candidate to apply to marine structural design.

From a designer standpoint, the early stages of design are critical in guiding the final solution to an optimal point. At this stage, an optimizer is used to search the design space and identify candidate solutions within the space that are promising. This early stage period contains one of the most critical tradeoffs for a designer, flexibility at the cost of reliable problem information. This flexibility allows designers to compare alternative designs identified by the optimization that should be further developed. Work has been done into quantifying freedom and flexibility in design as well as information certainty Simpson et al. (1998) in early stage design.
Many methods exist to deal with the lack of information. Approaches like set-based design seek to delay design decisions while more reliable information becomes available Singer et al. (2009). Another approach is to accept the uncertainty and design for it, using techniques like RBDO Tsompanakis et al. (2008). Other methods seek simply to design robustly in the face of uncertainty Diez and Peri (2010) utilizing probabilistic distributions in design decision making processes. A novel approach was proposed to innovate while optimizing (innovization) Deb et al. (2014) seeking to concurrently learn about the problem while optimizing it. This would seem to combine the best of both worlds, simultaneously exploring the design space with the flexibility of early stage design as well as adding information in a rigorous orderly approach. Future optimizers should approach the problem with the goal of learning while optimizing, producing both optimal solutions within the design space but also problem specific information. Again the BOA is promising in this regard. The Bayesian networks developed by the optimizer capture problem information. With an approach to interpret the network, this knowledge can be incorporated back into the design.

2.2 Summary

With an innovative approach to ship system level design, a new paradigm can be applied to engineering optimization. Within the system level discipline optimizers are selected for inherent efficiency when solving specific topological forms of the design problem. Focusing specifically on the design of marine structural scantlings, the nearly decomposable nature of failure mode evaluation and cost estimation provides great potential for efficiency improvements. To realize this improvement, Bayesian networks are applied to model the conditionally dependent structural elements within the design. The Bayesian networks constructed by the BOA record problem specific information often discarded by other optimizers, and provide a product to learn about the design and optimization process.

The Subsequent chapters will focus on documenting the exploration of these areas.

- Simulate and document the effort and efficiency of the BOA as compared to another heuristic approach when designing early stage marine structures.
- Explore and analyze Bayesian networks produced by the algorithm and connect the model to existing models of structural interaction as dictated by the objective function and constraints.
- Demonstrate how with the new process the designer can impact change or explore using knowledge gained to better effect the quality of the design solution.

These results can then be evaluated to determine the effectiveness of this new paradigm.

CHAPTER 3

Efficiency

3.1 Overview

To effectively create high quality designs in the early stages of the design process, efficient optimization approaches are needed. The added modeling accuracy in structural mechanics and evaluation continue to improve fidelity require additional computational expenses to resolve. As a result, reducing the number of fitness function evaluations becomes incredibly valuable especially in the time constrained epoch of early stage design. The BOA may be able to efficiently solve rule based early stage structural design optimizations due to the the nearly decomposable nature of the problem. The algorithm creates Bayesian networks which are well suited to represent and resolve conditional dependencies existing between scantling values. To explore the potential efficiency of the BOA, a Single Objective Genetic Algorithm (SOGA) is used as a benchmark. After developing a knowledge base on the problem and the response optimizers, results bound and define the performance of the BOA in solving large multi-section optimizations.

3.2 Theoretical Justification

A variety of approaches can be used to improve the efficiency of optimizers, ranging from algorithm techniques to problem specific changes. As discussed in chapter 2 the critical design problem characteristic in rule based early stage structural design is a nearly decomposable composition. A majority of the fitness function evaluations and failure mode constraint evaluations require only subsets of the entire design variable vector to evaluate. Efficient optimization algorithms can leverage this problem specific composition to more quickly solve the problem. The BOA is hypothesized to solve this problem well by using Bayesian networks to resolve conditional dependencies and probabilities associated with structural tradeoffs. Where other optimizers may try to merge, blend, slice, or otherwise use existing design points to best resolve constraint boundaries, the BOA identifies that the response of one variable is often conditionally dependent on another variable and it is counter productive to try to blend results. This gives the BOA a decided theoretical advantage when solving fully decomposable problems and this is hypothesized to extend to nearly decomposable problems, such as marine design problems.

To explore and confirm these expectations a case study needs to be developed, with a nearly decomposable nature. This case study will then be solved and documented to better understand the behaviour and performance of the BOA when solving marine structural design problems.

3.3 Case Study

3.3.1 Designing the Chief Case Study

A case study was developed to examine efficient optimization of nearly decomposable structural design problems (portions appear in Devine and Collette (2013)). The case study needed to have the explicit element of decomposability, with the ability to control the amount of decomposability in the problem. Additionally, the design would likely exist in early stage design where lack of prior designs exist, and designers need quick, accurate answers to explore the space and make tradeoffs. For this reason the US Navy TCraft was selected and used as a parent hull.

The TCraft is a surface effect ship (SES). Its hull forms an air cushion that creates lift for the vessel to ride upon. The air cushion pocket is created by a combination of solid structures on the port and starboard edges of the vessel, and flexible seals at the bow and stern. Therefore the midship section profile is unusually formed, having structure extending into the water to ”seal” the cushion. As this vessel, when on cushion, relies on less buoyant lift and more pressure lift, its structures are subjected to different loadings in a variety of zones. Additionally, the hypothetical design relies on aluminum structures rather than the more traditional steel. Herrington and Latorre (1998) give a more complete overview of design of marine structural panels.

Figure 3.1: Artistic rendering of a TCraft concept configuration. Note the black bow skirt and rigid structural sidewalls of the Surface effect ship, creating an air cushion pocket mar (2010)

The SES hullform was used as a reference and a baseline geometry, figure 3.3, was created to mimic a potential half section at midships. Further assumptions were made regarding potential ship particulars (table 3.1). These principle characteristic assumptions are used to size the required rule based structural constraints using of Shipping (2014). For this simulation a constant transverse frame spacing of 2 m was used. The level of detail is simple and the granularity matches that of early stage design, where the BOA and associated BN has excellent potential to positively

Principle Characteristic	Value
Length	100 m
Beam	20 m
Vehicle Deck Height	4m
Sustained Speed	20 kts
Displacement	1000 tons

Table 3.1: Relevant Assumed Principle Characteristics of idealized Tcraft

impact the final design solution positively.

3.3.1.1 Design Variables

At the earliest stages of design, the most critical assessment is the sizing of the primary scantlings and estimation of design cost.

The problem geometry was formatted into a series of grillages G_0 to G_5 shown in figure 3.3. These grillages are a series of identical TPanels arranged in series to create the primary load carrying structure of the vessel.

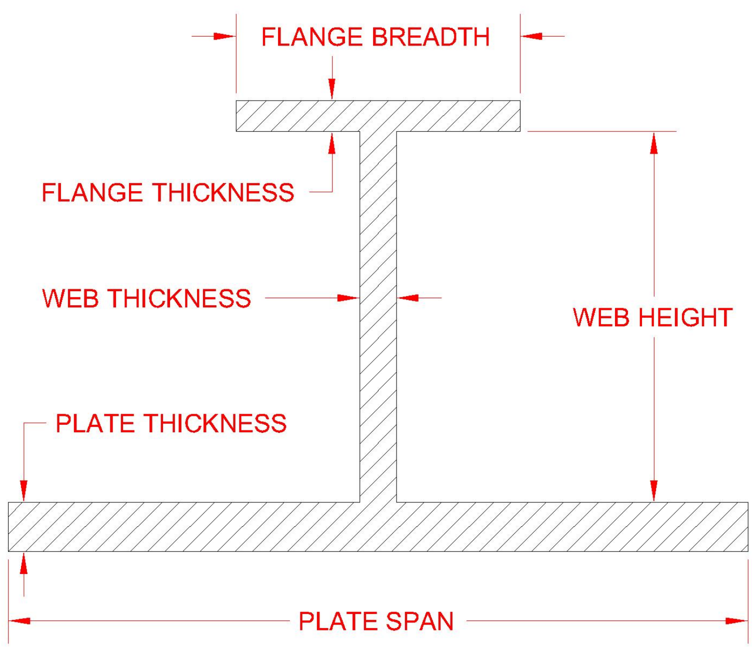


Figure 3.2: Representative geometry of a basic stiffened T panel. In the case study being considered, the various stiffener dimensions are controlled by a single stiffener selected from a catalogue, as denoted by S_i.

Stiffener Number	**Web Thickness**	**Web Height**	**Flange Thickness**	**Flange Breadth**
1	3	45.5	4.5	30
2	3.5	54.9	5.1	35
3	4	63.8	6.1	40
4	4.5	73.8	6.2	45
5	5	93.6	6.4	50
6	5.5	113.3	7.7	55
7	6	132.2	8.7	60
8	6.5	159.7	10.3	65

Table 3.2: Discrete Stiffener Catalogue

A T-Panel may be further decomposed into its principle parts, the hull plating P_t and the stiffener S. The dimensions of the stiffener has previously been entirely determined by the optimizer. Often, the use of real value variables brings question the true optimality of the as-built. Shipyards use only a standard catalogue of stiffener shapes and dimensions to build vessels, and often these structural profiles do not match the dimensions the optimizer specifies unless the design variable is constrained to only existing shapes. Therefore, the stiffener variable S was described in this optimization as one of 8 predefined stiffeners from a catalogue. This specifed all information on both the web and the flange for a given entry in the catalogue. The extent of P_t and the spacing of the stiffeners can further be controlled by the number of stiffeners N placed on a given grillage geometry. A grillage that is 10 m wide can have spacings of 5m, 3.33m, 2.5m, etc. with N = 2,3,4 etc. Therefore the simplistic early stage definition of a grillage requires only 3 variables, P_t, S, and N, to define the structure.

$$\mathbf{x} = (P_t, S, N)$$

These grillages were arranged to mimic a potential early stage configuration of a possible TCraft hull. This configuration is rudimentary but captures critical features of the TCraft. The midship section shows a large wet deck area, with room for a complete vehicle deck. Additionally the hard chined side walls exist extending into the water. Figure 3.3 shows the dimensions of the half section as defined by the centerline. This model is then used to calculate cost.

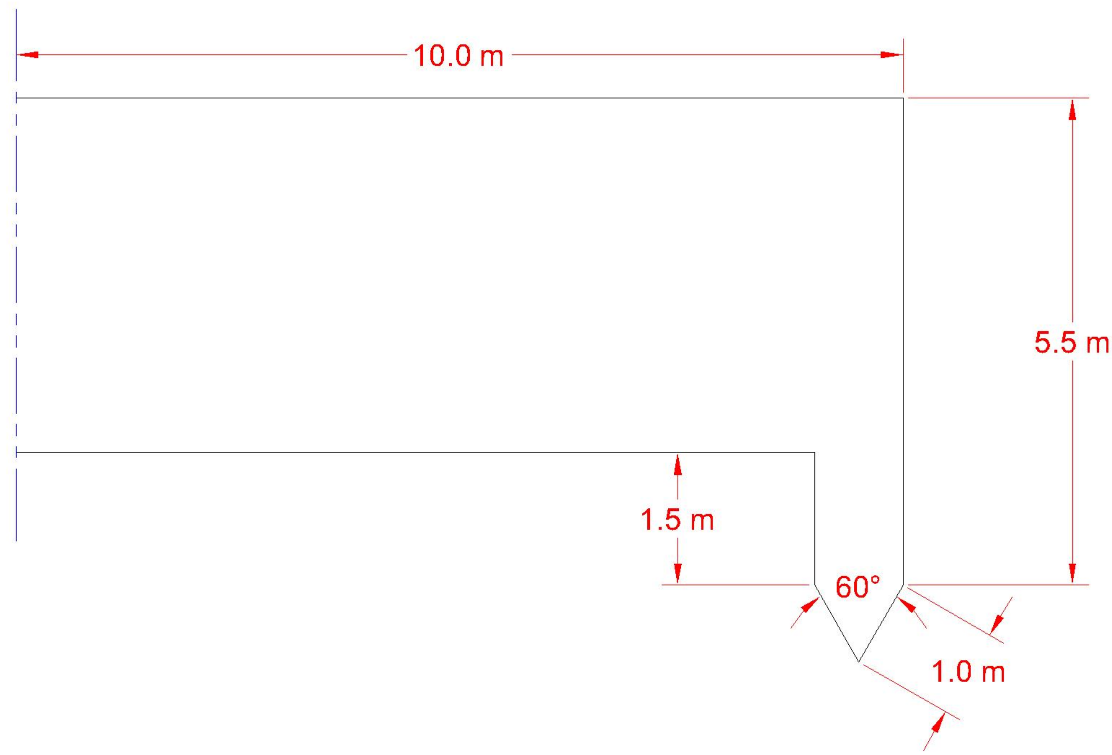


Figure 3.3: Case study geometry created to model the TCraft

3.3.1.2 Objective Function

Using the defined grillages the cost of the structure may be assessed and a principle output for decision making. Cost of production has become a major focus of a financially constrained US Navy. The cost of a structure can be evaluated in a number of way, but is primarily weight or volume based in early stage assessments. As a result, the costing model as proposed by Rahman and Caldwell (1992) will be

Cost Model Parameter	**Value**
Material Cost ($ ton)	2000.0
Material Density (ton/m^3)	2.72
Stiffener Cost Coefficient	1.05
Labor Rate ($ /hr)	27.0
Weld Rate (m/hr)	1.2
Fabrication Rate (m/hr)	1.5
Electric Utilization Rate (m/hr)	0.9

Table 3.3: Assumed Cost model Parameters

used to estimate the cost of the TCraft.

$$C_T = C_P + C_s + C_w + C_E + C_I \tag{3.1}$$

Where

$$C_P = 2000 * 2.72 * P_{t_i} * L * B$$

$$C_S = 1.05 * 2000 * 2.72 * A_{S_I} * N_i * L$$

$$C_w = \frac{27.0 * 2 * N_I * L}{1.2}$$

$$C_I = \frac{27.0 * 2 * N_I * L}{1.5}$$

$$C_E = \frac{27.0 * 2 * N_I * L}{0.9}$$

Assumed model parameters are given in table 3.4. It considers the cost of the plate at a present value of military grade aluminum. In the model, stiffeners cost slightly more, currently set at 105% of the material cost, to capture the added effort needed to shape the structural element. Production costs are then added based on the length of preparation and welding needed to bond the stiffener to the plate. Labor hours, welding expendables, and stiffener-plate surface preparation are all added based on the length of welding required. This leads the cost function to favor larger plate thicknesses and a smaller number of total stiffeners where possible in the structure.

Functional Location	Stiffener K4 Value	Allowable Stress
Bottom Shell	0.0021	$0.9\sigma_{yield}$
Wet Deck	0.0018	$0.9\sigma_{yield}$
Strength Deck	0.0018	$0.6\sigma_{yield}$
Side Shell	0.0021	$0.9\sigma_{yield}$

Table 3.4: Assumed Cost model Parameters

The optimization will seek to minimize the cost while satisfying required strength constraints.

3.3.1.3 Constraints

To balance the desire of the optimizer to remove material from the structure, structural constraints are applied based on the ABS High Speed Naval Craft Rules (HSNC). The selected rules are a subset of the total required rules, selected to specifically create a nearly decomposable problem for testing. Purely decomposable constraints assess strength and local failure modes of the individual T Panels. The local failure of a stiffened T Panel is decoupled from other surrounding T Panels, and only the 3 variables defining the TCraft are needed for assessment.

In this design problem, each local T Panel is checked for 2 primary failure modes, adequate plate thickness 4.2, and adequate T Panel moment of inertia 4.3.

$$g_1 : p_t \geqq s\sqrt{\frac{P_d k}{1000\sigma_a}} \tag{3.2}$$

$$g_2 : I_{stiff} \geqq 260\frac{P_d s l^3}{K_4 E} \tag{3.3}$$

The required plate thickness is specified by the spacing of stiffeners in each local grillage, the material properties used, and the functional location of the grillage within the vessel. Functional location 3.4 is defined by the ABS guide, and controls the allowable stress and stiffener K4 value (controls required stiffener moment of inertia) expected in the T Panel. These location dependent parameters lead to a rules based

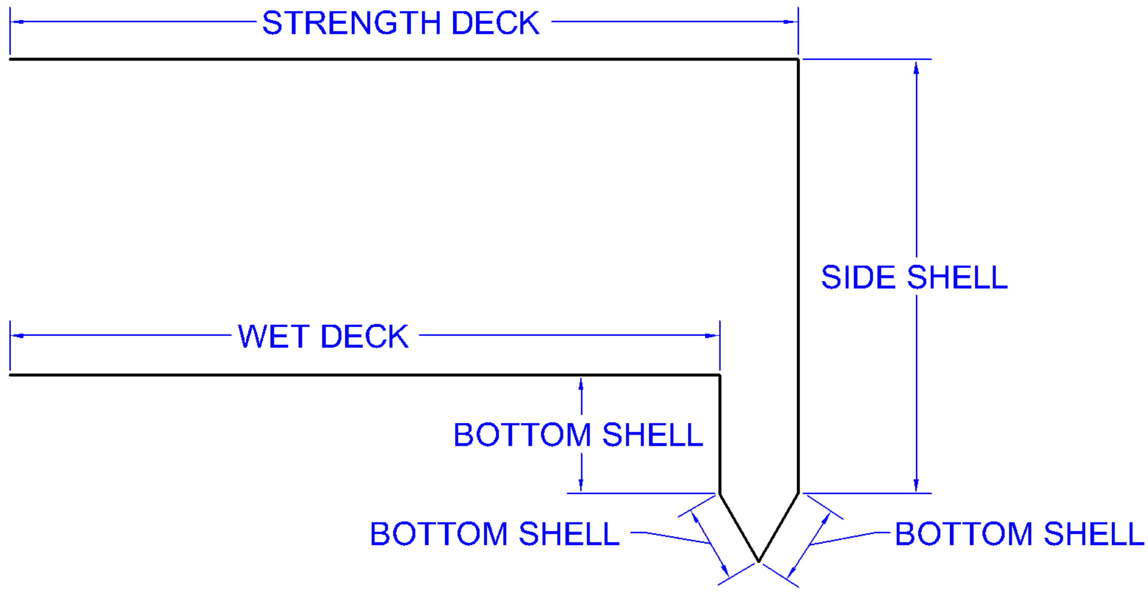


Figure 3.4: Local Functional Areas Definitions

constraint on the minimum plate thickness required. The required moment of inertia of the stiffener is similarly constrained. The geometry of the panel as defined by the design variable vector must meet a minimum neutral axis moment of inertia value again specified by a series of parameters. The required local moment of inertia is slightly more complex as it also requires the local stiffener shape, S in addition to P_t and N.

These constraints are a small cross section of the total rule based constraints placed on the vessel but they drive primary principle dimensions. Their effects are explicit enough to model, and yet complex enough to be difficult to easily comprehend. They are also highly decomposable in nature as each grillage within the structure has its own pair of plate thickness and stiffener moment of inertia constraints. It remains to add a non-decomposable constraint, which requires the entire design variable vector to evaluate constraint satisfaction.

To do this, the ABS HSNC rules are applied to determine a required section moment of inertia 4.4 for the design in question.

$$g_3 : I_{Section} \geqq I_{req} \tag{3.4}$$

The moment of inertia calculation about the neutral axis requires each variable within the entire design variable vector to be evaluated, as each controls material within the half section. For this vessel, the neutral axis moment of inertia was required to be 12 m^4 as required by the ABS standards, but due to the simple representation of the structure, that limit was relaxed to 6 m^4 Again this is a highly common rules based approach to structural design and leaves the case study example open to added change or increased complexity. Future work can focus on expanding scope or accuracy while modeling optimizer performance.

To assess constraint violation, external penalty functions were applied to make infeasible designs less desirable to the optimizer.

$$C_{net} = C_T + \sum_{i=1}^{n} R_k p_i^2 \tag{3.5}$$

where p_i is the normalized penalty violation and R_k is a multiplicative scaling constant. These penalty function warp the design solution space near beyond the constraint limit, such that infeasible designs are unpreferred. The constraint handling approach is one of the more basic but stable approaches. Thus the optimization may be stated as:

$$\text{Minimize: } C_{net}(\mathbf{x})$$
$$\text{Subject to: } \mathbf{g}(\mathbf{x})$$

This formulation was used in multiple studies for its simplicity and flexibility..

3.3.1.4 Effort Comparison Technique

One of the difficulties in comparing heuristic optimization techniques, both with other heuristic optimizers, and with other techniques in general is a lack of defined framework or process. Different approaches have areas of strength and weakness and it is challenging to select a like comparison that reveals performance results of the optimizers. Computational time serves little to enhance arguments as to which optimizer is more efficient as processor times and parallelization can yield a variety of run times. Instead, a common element to each simulation is the evaluation of objective function required. Barring the use of machine learning as a substitute to a true objective function, each design within the optimization approach must be evaluated, and provide a standard to compare to. As was discussed in chapter 2, these fitness functions have become more expensive and efficient design is a highly coveted aspect of optimization approaches.

Another challenge specific to heuristic optimization is the inherent randomness of the sample and optimization process. This means that should the initial design "guesses" be high quality, the optimization simulation is far more likely to reach the globally optimal solution and do so in a fast manner. Based on pure chance, a single point solution could produce the ideal design solution and the data would not reflect the true work of the optimizer but rather the fortuitous selection.. For this reason, means of data subsets are needed to show the intent/effect of the optimizer on the population. Therefore, a threshold strategy was applied to determine where the simulation "converges":

- Establish a baseline optimal solution for a single section optimization. This solution will be treated as a true optimal, since without enumeration it is impossible to guarantee global optimality.

- Create an acceptable offset threshold for convergence based on a percentage of

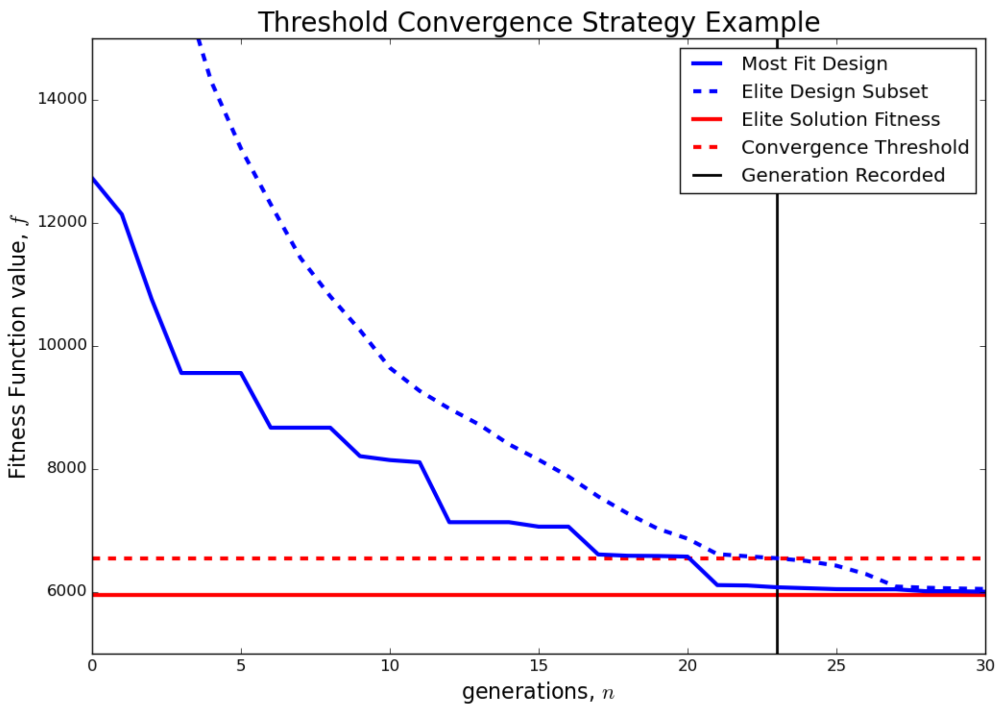


Figure 3.5: Example Simulation with Threshold Strategy Shown

the true global optimal score. This threshold ensures that the design reaches a value close enough to the true optimal.

- Sample the populations of each heuristic population-based algorithm selecting a subset of the most fit solutions. Use the mean value of this subset and compare it to the threshold value established.

- If/when the mean value of the subset selection is lower than the threshold value established, the simulation has acceptably converged to the true global optimal. The generation number is recorded if this fitness is reached and the number of objective function evaluations may be totalled.

For these simulations, the elite subset was assumed to be 20% of the total population. The mean of this subset was required to be less than or equal to 110 % of the assumed global optimal value. There exists debate as to whether the threshold will effect the convergence results. These concerns are legitimate, but ultimately only affect the specific values of the simulation means. The overall behavior of the algorithms should remain constant, and this overall trend is the critical result. The study of the optimizer comparison is focused on efficiency not optimality (as both the BOA and SOGA have been proven to reach the optimal solution at properly tuned heuristic parameters).

This process is simulation intensive and a sampling approach is therefore needed. The following procedures were used when sampling the data:

1. Select initial population size m, and simulate 5 random seed simulations.

2. If all 5 random seeds converge within the acceptable threshold, simulate 20 additional random seeds.

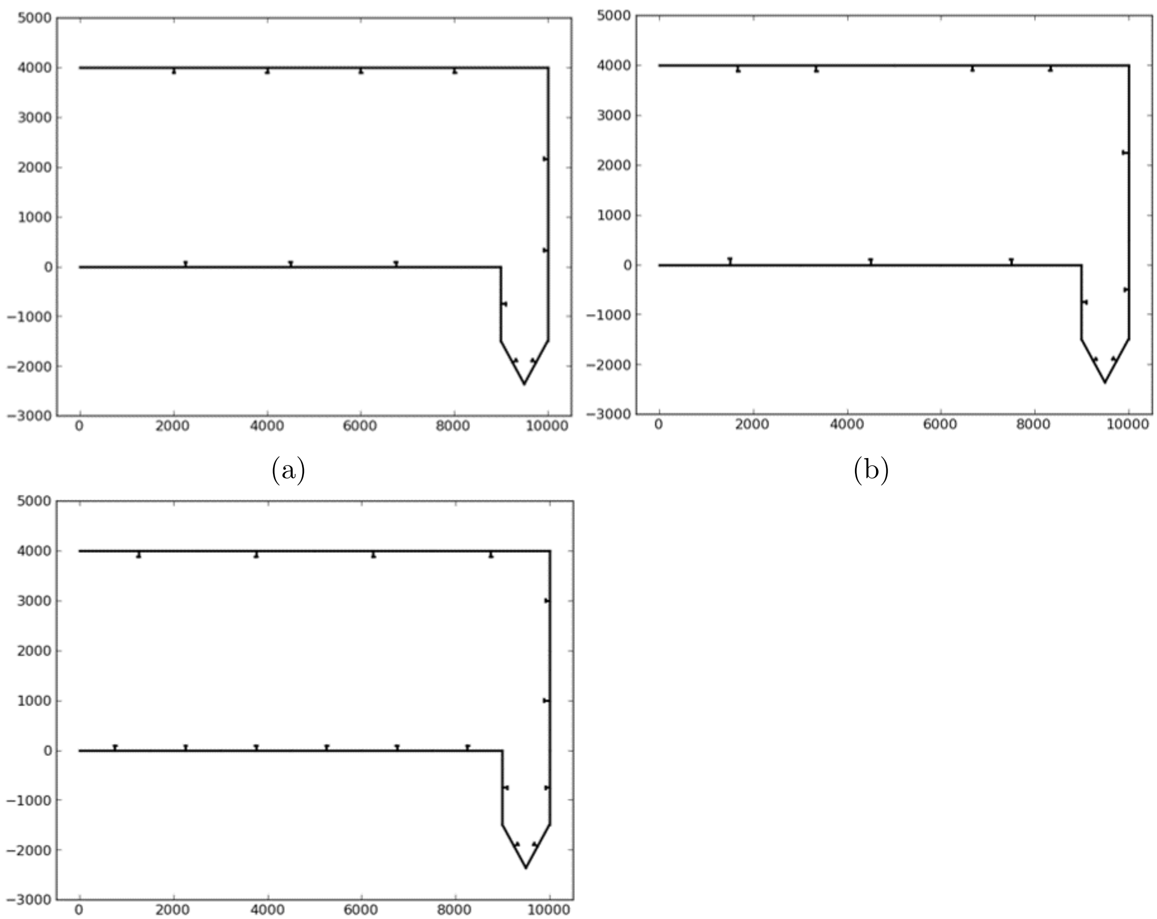
3. If 2 or less simulations out of 25 ($< 10\%$) fail to converge within the threshold, lower the population size. Conversely if 3 or more simulations fail to converge, increase the population size.

4. All population size changes are made in 20% increments. For example an initial population of 100 members would be increased to 120 or decreased to 80 members.

5. When the limit of population size is determined, reduce the population increment size to 10% of the original population to refine the exact threshold limit. For example, if a 100 member simulation fails to converge, but a 120 member simulation converges, conduct the 110 member simulation to determine the correct population size. If 110 member simulation converge, use the results, else use the results of the 120 member simulations.

3.3.2 Initial Optimization Results

To assess the efficiency of the BOA an initial study was conducted on a single section design of the TCraft. After many consecutive runs with populations well above the threshold range, the minimum cost optimal design was determined. The total cost of a 100 m structure was determined to be 150,090 USD. It is important to note that this cost is a preliminary value, and its USD value is likely irrelevant in a realistic vessel due to the simplicity of the structures within the case study. This value is useful in relative terms however as it allows for comparison of random seed simulations. The topology of the design is shown in figure 3.6a.



(a) (b) (c)

Figure 3.6: Resulting structural configurations for the 6 grillage (figure 3.6a), 10 grillage (figure 3.6b) and 16 grillages (figure 3.6c) discretizations of the single section problem.

To create a range of problem sizes to test, the single section was then discretized into 10 grillage and 16 grillage designs, by subdividing the large grillages within the existing design. These results had similar costs to produce, the 10 grillage solution costing 151,900 USD and the 16 grillage solution costing 210,640 USD to produce. As large grillages were subdivided, the solutions became unnaturally constrained by the requirement to have at least 1 stiffener in each grillage. This division strategy adds unnecessary stiffeners to the desgin and can be observed when comparing figures 3.6a and 3.6c.

3.3.2.1 SOGA

To begin the SOGA was run on the problem. Previous optimization experience has led to the conclusion that Genetic Algorithms struggle as problem sizes become large. It remained to be seen however if the existing problem size was large enough to challenge the optimizer. The SOGA used was binary coded. As with most heuristic optimizers, a variety of parameters were set to produce good results. The SOGA was run with a Crossover rate of 99% and a mutation rate of 1%. These parameters could likely be altered to produce small improvements in optimizer performance, but in general these parameters provide a fast and stable simulation for comparison.

One point of contention was the number of crossover points to use within the problem. If the optimizer was given each variable location within the binary vector, it would be more efficient. However this would seem to unfairly promote the SOGA as compared to the BOA. As a result, a singe crossover per chromosome was applied. Each chromosome has 9 genes, which after binary conversion represent the 8 possible states of the 3 design variables on each T Panel. In this way crossover occurs in individual T Panel, a seemingly fair compromise between an uninformed algorithm and one with complete knowledge of the problem.

As the SOGA is noted for its robustness in general as an optimizer, it was always

Grillages	BOA Population Size	BOA Convergence	SOGA Population Size	SOGA Convergence
6	1000	80%	133	90 %
10	2000	30%	266	100 %
16	3000	60%	400	100 %

Table 3.5: Convergence Comparison between BOA and SOGA

able to meet the threshold throughout the simulations. Results converged within the 90% interval for all simulations of 6, 10 and 16 grillages. These problem sizes correspond to binary problem sizes of 54, 90 and 144 bits. The results were recorded and displayed in figure 3.7. With these results, the BOA was also run on the same problem.

3.3.2.2 BOA

This optimization of the BOA represented one of a handful of times the algorithm was applied to solve problems more complex than canonical optimization problems. Thus, the results determined by the optimizer were somewhat mixed. The initial study with the BO used a binary coded version as well for like comparisons. This represents nodes as 2 bin values. Like the SOGA, the BOA needed parameter values to be set, though experience with these parameters was lacking. The BOA was run with a 50% selection rate for selection of the training pool. The number of edges was set a a maximum value of 3 edges, limiting the search space of the algorithm.

One of the critical results of this initial simulation was that the BOA could succeed in solving a complex nearly decomposable design problem, where it had previously been applied to pure decomposable problems. Table 3.5 shows the convergence percentages of the BOA simulation. Clearly the 10 grillage discretization problem struggle to converge within the defined threshold. This was one indication more parametric studies were needed to better understand how to affect convergence through BOA optimization parameters.

3.3.3 Initial Results of the Effort Comparison

The initial comparison yielded interesting results, suggesting further effort into studying the BOA at larger scales. The simple change from 54 bit to 90 bit to 144 bit showed the effort required to solve the problem went from 5 times the effort to 3 times the effort (figure 3.7). The effort to discretize the single section problem into multiple numbers of grillages led to some unintended consequences. The smaller grillages left more of the bins unobserved, such that the learning stages both network structure and conditional probability tables was difficult. As previously discussed, the binary representation has only 2 possible bins to represent a single node. With the need to represent 8 states, 3 nodes in combination are needed for the binary to discretely represent the stiffener catalogue. As a result, some design variable states are not well represented in the statistical sample of the population. This representation led the BOA to poor convergence results as the training set lacked statistical diversity needed to resolve the problem.

Results left multiple new directions for research and exploration. First the parameters of the BOA needed to be explored to better understand the poor convergence observed in the 90 bit simulations. Second, the point at which the SOGA would begin to struggle needed to be identified. The nonlinear trend should be observable, but the required problem size was unknown. Finally, the BOA code structure needed to be reconfigured to better represent the sample.

To better explore the identified challenges, the following results were produced. First the population size and selection percentage parameters of the BOA were explored. Next, the problem size was increased by duplicating the single section multiple times. This leads away from a contrived effort range towards a more practical representation of a multi-station design, where 10-20 station sections are created to design an entire ships scantlings. Additionally, with the use of multiple identical midship sections, as the minimum cost of a single midship section is known meaning individual

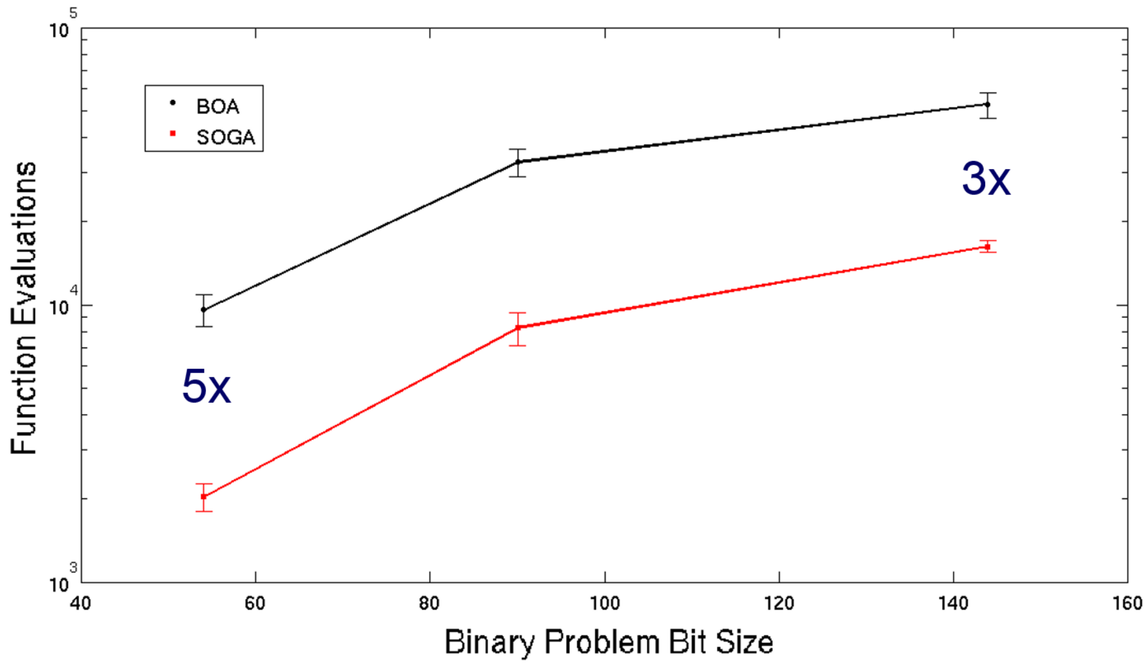


Figure 3.7: Initial comparison of optimizer effort required to reach the 110% objective function threshold. Effort begins with a 5x difference between the BOA and SOGA and the effort closes to 3x in a problem size difference of 54 to 144 bits.

minimum threshold values must not be calculated for each problem size. Finally the BOA is re-coded to represent not only binary bin variables, but discrete bin values, with r possible bins. This change in variable declaration was a major step forward in the representation of problems, as a variable is represented on a single node rather than multiple linked binary nodes and will become critically important in chapters 4 and 5.

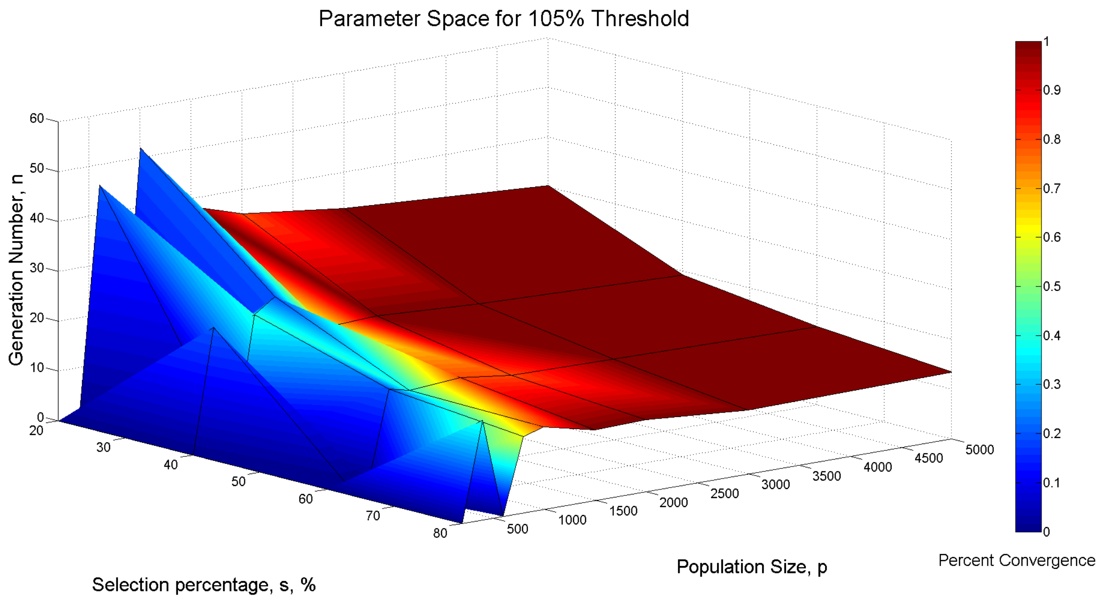
The change from single section to multi-section also adds an extra level of decomposability, furthering the potential of the BOA. Again this matches the pattern canonical problems exhibited, where a design solution with one highly fit section and other poorly fit sections can be combined through the network learning with other design solutions that have equally high fit solutions in other sections. The simplistic evolutionary mechanisms of the SOGA lack the ability to combine highly fit selections between parents, and is likely to struggle even more as problem size increases.

3.3.4 BOA Parameter Exploration

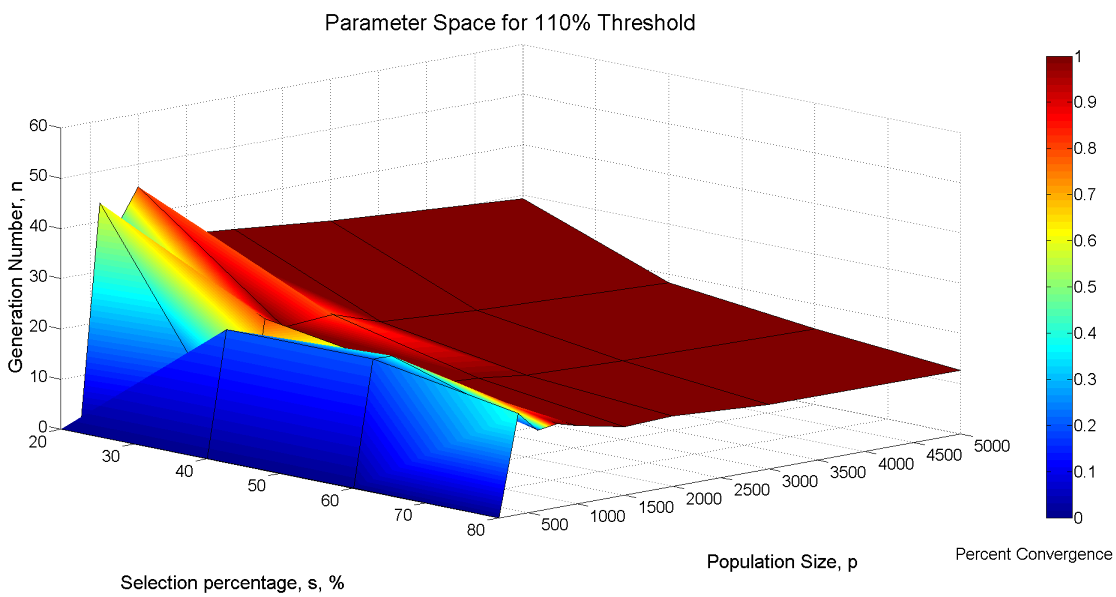
At this point, the knowledge about BOA parameters was lacking as compared to the knowledge base built over time for the SOGA. To better understand how to tune the BOA, a parametric study exploring the effects of population size, training pool selection percentage and optimizer threshold values was conducted. It was hoped that this study might better guide selection of parameters to speed the population size selection process. The parameters being explored have a basic tradeoff within the optimizer process. The BOA needs a quality data sample to better refine the network and produce better candidate solutions. The quality of the training data is most effected by the population size and selection pressure in combination. A larger population size creates more points within the space and leads to a better sample of optimal points to train the network with.

The selection pressure controls the quality of the sample through the range of objective function values it passes to the network learning phase. A lower selection pressure (or higher selection percentage) gives a large range of data to the network learning phases of the BOA. This slows the convergence of the algorithm, but produces better networks as the extra data allows for better refinement. There is also less replacement within the population after each generation, which additionally slows convergence of the population to a single point.

Conversely a higher selection pressure (or lower selection percentage) passes a smaller more highly fit sample to the optimizer. The Bayesian networks will thus more strongly prefer to replicate the solutions, while ignoring some of the space that is less represented by the small training data size. As a result, the optimizer will converge more quickly, but may explore the space less. There is also a much higher replacement of members, allowing single designs to converge much quicker.



(a)



(b)

Figure 3.8: Example convergence surfaces for optimal value threshold values of 105% (3.8a) and 110% (3.8b).

3.3.4.1 Parametric Results

Figures 3.8 shows representative surfaces of the parameters being studied. The base axes have ranges of selection pressures and population sizes, while the vertical axis shows the number of fitness function evaluations required to reach the specified threshold.

The behavior of this graph is quite striking, showing a clear need for a minimum amount of diversity within the sample. This diversity can be achieved either through a larger selection percentage or through a larger population. Without the diversity of the sample, the simulation often converge prematurely, giving a false representation of the ideal design solutions. The lack of diversity is marked by a distinct cliff, colored blue in the aforementioned figure. The designer must navigate this tradeoff in selection of parameters. Erring with too much data in the training subset will lead to inefficiencies in early stage designs, while erring with too little data leads to premature convergence. This mirrors the behavior of evolutionary optimizers closely, and serves as excellent guidance moving forward with BOA simulations. It also suggests that the poor convergence of the 90 bit BOA simulation was due to a poor sample size, and a small increase in the population size of the algorithm would produce much higher convergence rates.

3.3.5 Algorithm Adaptation

After the initial results produced by the BOA comparison, the decision was made to modify the variable representation in the algorithm. Instead of binary variables, strung together, variables would be entirely represented on a single node within the network. This decision has benefits and disadvantages. In terms of information representation, the networks produced by the new version of the algorithm are far more accurate, preventing false edges, connected between intermediate bits of unrelated variables.

On the down side, the spaces associated with the conditional probability tables can become massive. In a binary representation, a node with 4 parents has a total bin search space of 2^5. By the same comparison, with 8 discrete bins, the search space is

8^5. The difference in data representation can have lasting repercussions on the speed of the network learning processes. Additionally it makes more complex networks with large maximum incoming edge limits more difficult and costly to create.

Therefore a decision needed to be made on the representation of variables in the BOA. A discrete representation had the advantage of properly representing single variables on a given node. In theory this change would make the edges of the network have true meaning, as opposed to potentially misleading relationships. With this more concrete network, the BOA might faster resolve the design problem, requiring fewer generations to converge the population. Conversely, as discussed, this representation creates a massive search space, and EDAs rely on the quality of statistical samples to accurately optimize the design problem.

The decision was made to recode the BOA to a discrete representation, specifically for the power to represent the design variables on a single node. With this newly reformatted algorithm, the problem of efficiency was again addressed with a new multi-section approach.

3.3.6 Mutliple Section Effort Comparison

With the need to increase the effort range of the problem to truly understand the potential efficiency advantage of the BOA as a structural optimizer, the simple 6 grillage section optimization was duplicated a number of times to increase the size of the problem. As discussed, this increases the decomposability of the problem as each section becomes an optimization independent of one another, linked through the addition of the cost. Again this benefits the BOA and should further help to improve the initial results.

First this problem was approached with the SOGA to determine how large the problem size must be before the algorithm struggles. Simulations up to 5 sections or 90 variables successfully converged with required 90% success rate. Beyond the

5 section values, the optimizer was unable to reach the 90% convergence rate, with only 22 of 25 simulations reaching the required 6 and 7 section threshhold. In figure 3.9, the number of sections investigated begin to reveal the nonlinear response of the optimizer, previously unobserved at smaller problem sizes. Also of note, the variability of the convergence mean grows in a nonlinear fashion as well.

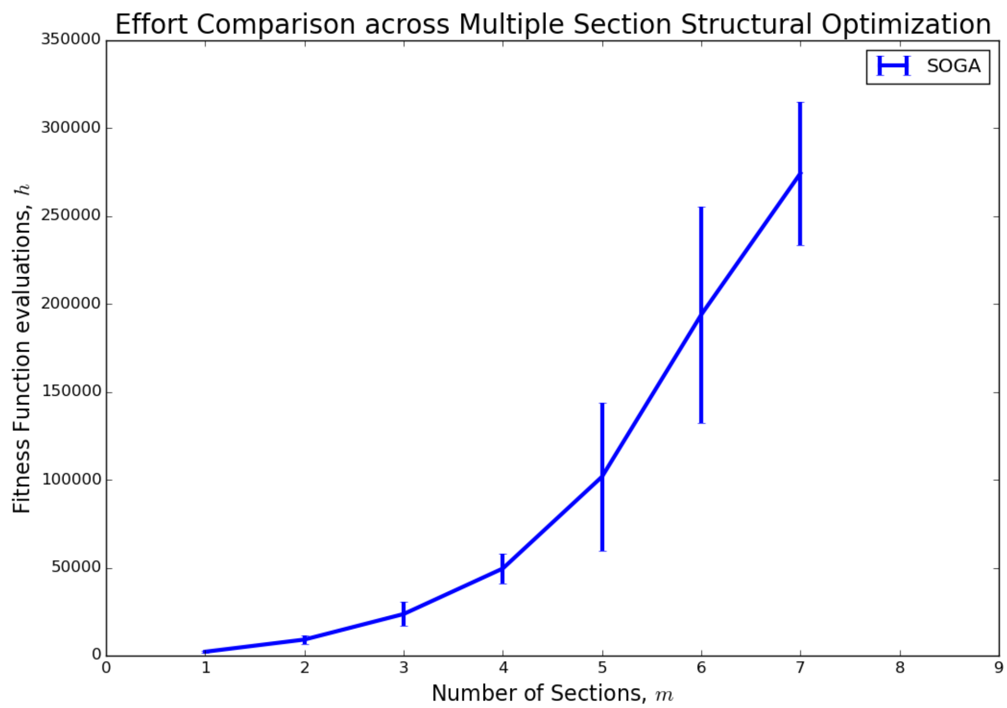


Figure 3.9: Resulting Multiple Section results for only the SOGA optimizer, showing nonlinear response

The BOA showed mixed results. With the new discrete representation, the BOA was able to solve the multi section problem. The network learning process however began to become intractable beyond 2 section problems. Especially in the early stages of the simulation, when the data sample is poor and more random, the optimizer generally tends to hit the maximum incoming edge limit. Even with the lowest limit of 2 incoming edges, this requires a massive amount of search power.

For example in a 2 section simulation there are 36 nodes. In the LGA approach, each of the nodes is check against every other nodes, creating $(n-1)^2$ operations to check each edge addition. Once these edges are added, there are 8^3 bins to check. And if the edge limit is hit for all nodes, this needs to be conducted $2n$ times. Finally this is conducted g times for the number of required generations, which is usually >10. The result is a search space that is approximately $5 * 10^8$ large for a single simulation. The number of searches in the data set increases in order of magnitude as the number of sections grow.

This massive computational requirement makes the network learning process currently intractable. There are many possible strategies that exist to try to reduce the computational time and allow this tool to be rapidly used in early stage design. AA return to the binary based code may more efficient result provided the number of required nodes in the binary representation does not become intractable. The LGA search will become a limiting factor for this approach, as each node must be connected and compared to every other node.

Another more basic approach is to remove the network learning process entirely. Instead a good fit network is used in its place and is applied invariantly, shown in figure 3.10. A good fit network is suggested from work conducted in chapter 5. It adds edges:

- $P_0 \rightarrow S_0$
- $P_0 \rightarrow N_0$
- $P_5 \rightarrow S_5$
- $P_5 \rightarrow N_5$
- $N_4 \rightarrow P_4$

These edges have been shown to be a common set to high quality networks (see chapter 5, and it was thought that these critical edges would allow the algorithm to resolve the design space without a complete network learning step. The result without network learning should be a bounding limit, with a complete BOA learning process performing no worse than the invariant network.

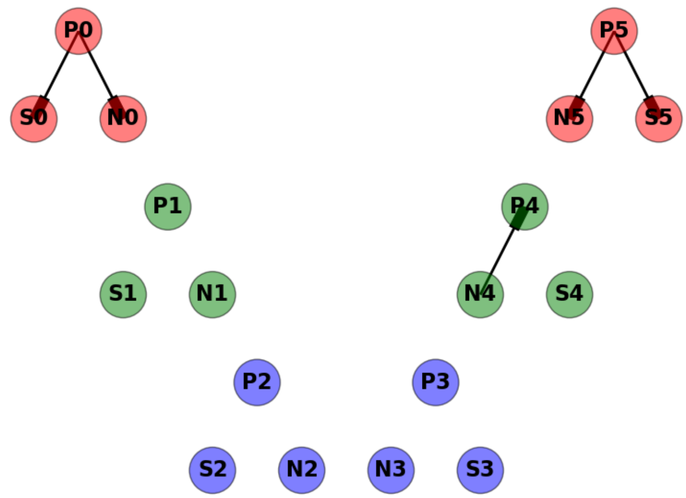


Figure 3.10: Invariant Network used based on high quality training data set

3.3.7 Adjusted Approach Results

With the adjusted approach the simulations were again conducted. The results were mixed, both showing potential for positive improvement while also revealing potential future areas for improvement in the algorithm application. The invariant BOA struggles as many simulations fail to reach the threshold required. When the simulations do reach the threshold, they do so quickly. The invariant network likely prevents some samples from improving the required amount to converge properly. Good initial guesses converge while poor guesses lack the optimizer guidance to reach highly fit areas of the design solutions space. The potential exists that a learned network will out perform this result. In comparison to the true number of function evaluations computed, the mean number of generations shows the potential of the BOA. When the BOA converges, it requires less generations to do so. In a parallel environment, the BOA could improve even further, suggesting the efficiency improvements to be made.

The most striking positive result of the multiple section comparison is the speed at which the BOA converges in number of generations. For the single section case for example, the slowest BOA simulation requires only 14 generations to reach the threshold value. In comparison, the SOGA requires 16 generations for its fastest simulation. The SOGA is, however, able to resolve the problem with a smaller number of candidate members and remains more efficient than the BOA at lower numbers of sections.

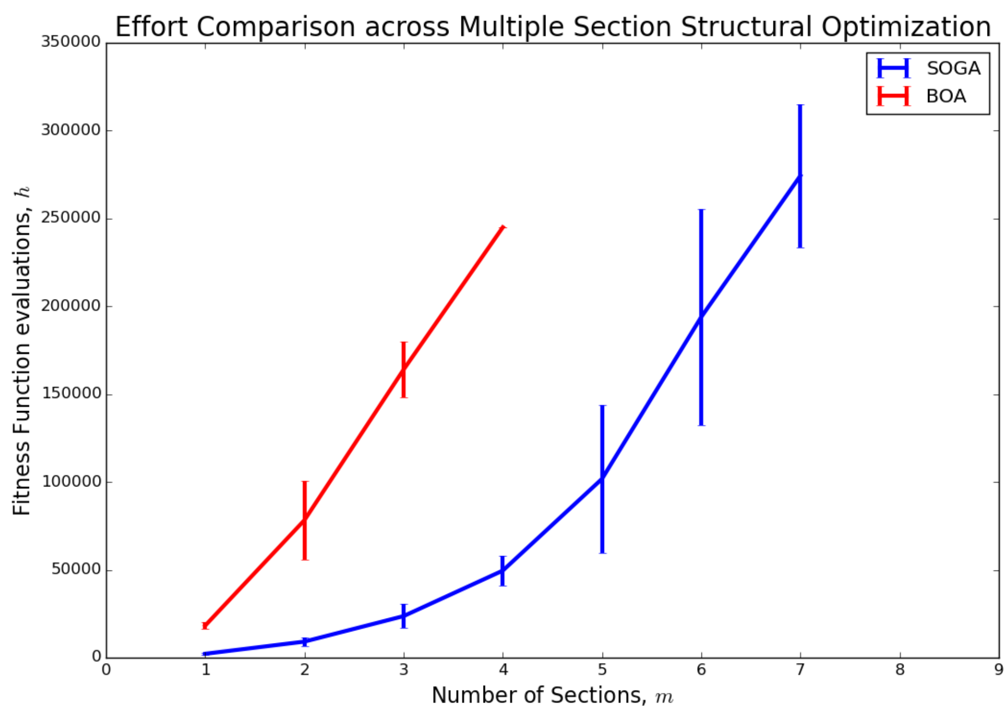


Figure 3.11: Resulting effort requirements for an adjusted invariant network implementation of the BOA

These combined results do suggest there remains potential for exploration of the BOA as an efficient optimizer of marine structural design problems

3.3.8 Room for improvement

The case study presented was used as a flexible foundation for future studies to be conducted. Its formulation allows the ratio of decomposable and non-decomposable constraints to be changed such that the effects on the optimizer may be observed. For example more non-decomposable constraints may be added to test the limit of the BOA's increased efficiency as the problem transitions from nearly decomposable to non-decomposable.

Additionally, the problem is currently formulated as a rule based approach, but the objective functions or constraints may be adapted to add numerical simulation results. The adaptation allows FEA simulations or other numerical modeling approaches to be used provided the results can be condensed into a set of training data. This also allows for the use of heuristic and qualitative terms where an optimizer such as the SOGA might struggle or need adaptation. The added flexibility and capability will become important as the BOA transitions from a tool used purely for its efficiency to one that allows the designer to understand problem dynamics.

In the aforementioned numerical simulations, the results lack simplistic condensed mathematical relationships that can be readily absorbed and applied. This is often a shortcoming of numerical simulation and only through repeated use or trial and error does an engineer build up intuition that can rapidly be used to change respond to and improve designs. The BOA holds promise to return some basic capability to designers who have not spent the years necessary to understand the intimate details of the simulation. Improved designer understanding of the design problem is incredibly important and will become the focus going forward in chapter 5.

3.4 Summary

To demonstrate the BOA's ability to optimize early stage naval structural design problems, a case study was created and simulated solving for minimum production cost. The problem was designed to exhibit nearly-decomposable behavior, common of structural cost estimation and rules based failure mode evaluation approaches. This problem specific topology was hypothesized to provide an advantage to the BOA as compared to other heuristic optimizers. The BOA was then compared to a SOGA using an effort benchmarking strategy focused on objective function evaluations, the most costly portion of the design optimization simulation.

The presented results were the first demonstrated example of BOA application to practical design of ship structures. It also remains one of the few discrete formulations used to optimize engineering problems beyond the scope of canonical benchmarking problems. The BOA results showed the ability of the optimizer to successfully solve the structural design problem, verifying it as a useful optimization technique

Furthermore when comparing the effort required to reach acceptably fit design solutions the BOA outperforms the SOGA as problem size increases towards sizes comparable to practical ship design problems . The trends indicate that the BOA is indeed an ideal optimization approach for use in solving structural design problems. Its response to constraint violation shows the value of the Bayesian networks, resolving the constrains much faster that the BOA on average.

CHAPTER 4

Learning

4.1 Overview

While the gains made by the BOA in efficiently solving nearly decomposable structural problems are groundbreaking, it falls into a pattern that seems to exist for a majority of heuristic optimizers. Optimizer performance is incrementally improved using various techniques and addition relevant to unique problem specific differences, to gain efficiency advantage. In some sense the BOA application to the nearly decomposable structure falls exactly into this pattern. This result however makes no attempt to change how the designer understands or comprehends the problem, seeking to solely chase speed. It leaves a knowledge gap in designer understanding, and particularly in early stage design where many of the potential costs are determined, this is incredibly dangerous. As a result, changes need to be made to the design optimization process not product, in order to have the process capture and learn more information about the problem and relate it to designers in a simple and comprehensible manner.

The exciting part of this paradigm shift is the manner in which the BOA learns about and solves the design problem, through its BN. These networks are legacy representation of the relationships being negotiated and explored at various stages of the design problem. Additionally, edges are the exact type of simple relationships that a designer needs, showing how change or cause and effect might propagate throughout the design. They have rigorous mathematically backbones that can be coupled to the optimization process to help designers better understand large and complex problems.

The BOA can be one potential solution to the shift in design process. The optimization routine must learn about the problem and provide insight back to the designer. To do this the first most salient place to examine is the Bayesian network. Through simple network analysis techniques, the design interaction can be identified and quantified. This chapter of the thesis will focus on initial attempts to learn from the Bayesian network, setting the stage for use of the lessons learned in chapter 5. The results of this analysis will seek to determine importance of variables through in degree and out degree analysis. The networks of the initial optimizations are studied first to see what the algorithm is producing. Then, a structured regimented approach to understand how the designer adds and removes edges to a decomposable problem is presented. Finall, partial derivatives within this network are explored to attempt to connect the mathematical function to the observed results.

4.2 Simple Network Analysis

The language and operators of networks provide a very comprehensive vocabulary and skillset to anaylze simple networks, and are well covered in David and Jon (2010); Newman (2010). Edge position and orientation has the potential to quickly dictate to designers the flow of influence and relation between variables. The most basic uses are in degree and out degree of a given node. In and out degree are calculated by summing the number of edges terminating and originating at/from the node. Upon completion of the initial computational efficiency study, multiple refinements were made to the BOA. Portions of the following are closely excerpted from work published in Devine and Collette (2014). From the last chapter, 3, code was rewritten such that variables were represented on a single node, rather than a collection of binary nodes strung together to create enough discrete bins to best represent the problem design space. This effected the way that the network structure is refined. Conditional dependence indicated stronger interaction between variables, and false relationships representations have been mitigated with the new variable structure within the network. Additionally, a basic Bayesian network analysis code library now accompanied the optimizer. This toolbox, NetworkX, contains a basic complimentary set of network manipulation and analysis methods. This would better allow the networks produced by the optimizer to be analyzed and presented for interpretation. Similar to the efficiency study, 10 numerical simulations were conducted. Only the 6 grillage geometry discretization was used. Within the Bayesian networks, 3 nodes per grillage were created. Nodal designations of P, S and N correspond to Plate, Stiffener and Number of stiffeners per grillage respectively. For example, grillage 1 would be defined by variables P1, S1, and N1.

$$C_T = C_P + C_s + C_w + C_E + C_I \tag{4.1}$$

$$g_1 : p_t \geqq s\sqrt{\frac{P_d k}{1000\sigma_a}} \tag{4.2}$$

$$g_2 : I_{stiff} \geqq 260\frac{P_d s l^3}{K_4 E} \tag{4.3}$$

$$g_3 : I_{Section} \geqq I_{req} \tag{4.4}$$

This design problem was solved to minimize cost as defined by eq. 4.1, subject to constraints defined in eqs. 4.4, 4.2, and 4.3.

To better understand the problem behavior, the initial population was used to determine constraint satisfaction/violation. Table 4.1 shows the percentage of population members with constraint violation of the global neutral axis costraint as well as lo-

Constraint	G0Violation Percentage
Global NA I_{xx}	54.1
Grillage 1 Plate	56.1
Grillage 1 Stiffener	67.9
Grillage 6 Plate	69.9
Grillage 6 Stiffener	70.5
Grillage 3 Plate	4.5
Grillage 3 Stiffener	19.1
Grillage 4 Plate	4.8
Grillage 4 Stiffener	20.0

Table 4.1: Generation 0 Constraint Activity

cal constraints for grillages 1, 6, 3 and 4. Two types of behavior are observed. For the longer grillages, 1 and 6, a very high percentage of the population fails the local constraints, with the stiffener constraint having slightly higher rate of failure. By comparison, grillages 3 and 4, which are the shortest in the design have much lower rates of violation. Ideally, as initial metrics are applied the results would clearly point to the criticality of properly selecting grillage 1 and 6 solutions. The feasible space is clearly quite small and highly fit designs will need to determine where constraint boundaries exist. Furthermore, a random population fails the global constraint on neutral axis moment of inertia in 54% of designs. Since grillages 1 and 6 are large and located relatively far from the neutral axis, they should clearly have great impact on moment of inertia constraint satisfaction as well, with grillage 6 being slightly more important due to its greater moment arm distance.

To begin, out degree was used as a metric to compare the simulations conducted. Out degree is a measure of the number of edges originating from a node. For example a node with 3 children would have an out degree of 3. It was theorized that nodes with higher out degree values would have greater impact on designs and could therefore be identified as design drivers. Based on the analysis of constraint violation, grillage 1 and 6 variables should have high out degree values. It was anticipated these grillages would drive high quality designs. Selection of good values would have a trickle down

Simulation Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Cost (thousand USD)	430.4	457.1	408.1	404.9	532.8	434.3	408.9	506.8	455.3	377.6
Gen. 0 Out Degree P6	17	17	17	17	3	17	17	17	17	17
Gen 0 Out Degree P1	15	1	0	0	14	0	16	16	16	15
Gen 0 Out Degree P2	0	14	16	15	16	0	0	0	0	1
Gen 0 Out Degree P5	1	1	0	1	0	1	0	0	0	0

Table 4.2: Cost simulation results and Generation 0 post learning out degree values

effect based on the structure of the Bayesian network. Table 4.2 shows the out degree results for the 10 simulations as well each simulations converged objective function value achieved. Contrary to the initial efficiency study, simulations which failed to converge close to the optimal value were kept and provide insight into what differentiates elite from good designs.

From the results, the conclusion that high out degree is a decent indicator of importance is supported. In the out degree analysis, the plate variables on grillages 1,2,5,and 6 had non-zero values, while all other nodes in the network remained children. Clearly in solving local constraints, the plate thickness value of each grillage affected the outcome the most, and this is to be expected based on the mathematical formulas in eqs. 4.2 and 4.3. Returning to the initial hypothesis, grillage 6 was critically important. As expected, all but the worst simulation selected P6 as a root node, having 17 edges leading to all other design variables. Its importance in solving the difficult local constraints as well as the global moment of inertia constraint lead the optimizer to value the choice of bins. Interpreting this result, a designer should invest time to properly select a value of P6. By doing so, future designs are predisposed to have a higher level of fitness as compared to a randomly selected design. The failure of simulation 5 to select P6 as a root node predisposed it to a suboptimal converged solution, with a final objective function value 40% greater than the most elite simulation, simulation 10.

Beyond variable P6, network structure was varied. This was to be expected, though an interesting trend did occur in variables P2 and P5. These grillages have intermediate lengths. Resolving the local scantlings of these intermediate grillages had a smaller effect in lowering objective function values as compared to properly resolving grillages 1 and 6. Doing so however, caused good simulations to become elite, with the incremental gain being the defining difference. Simulations reaching elite fitness values attempted to place either node P2 or node P5 as parents to P1 to better resolve the intermediate grillages. Figure 4.1 compares a the average network produced by all 10 simulations to the most elite simulation, simulation 10. The network structure

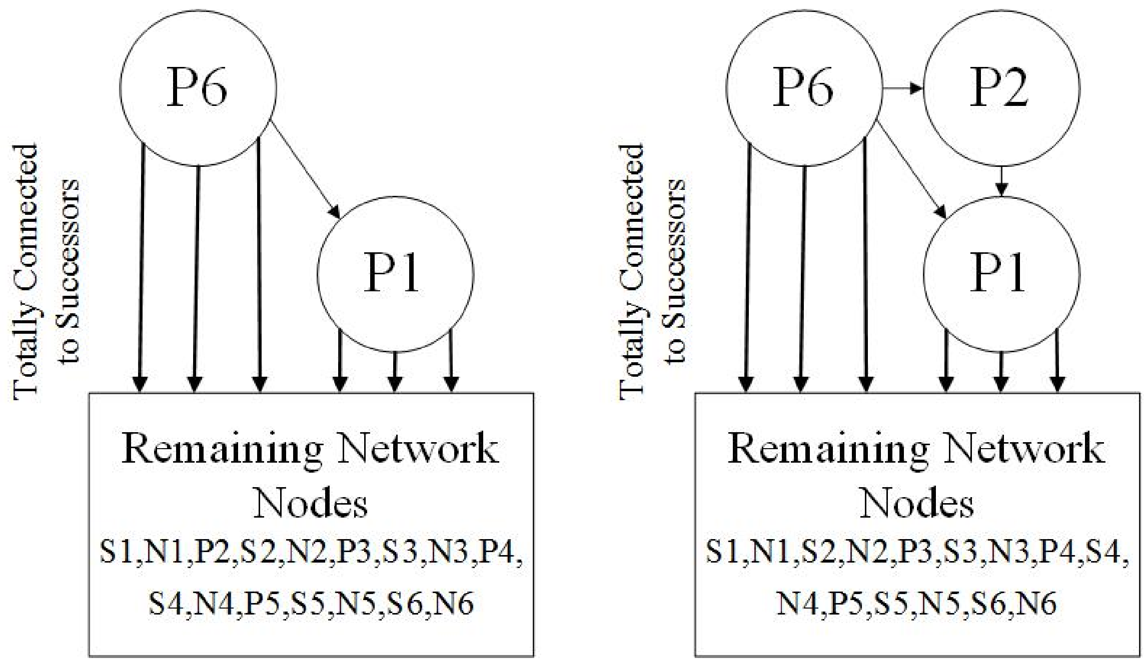


Figure 4.1: Network structure comparison of average(left) and most elite(right) simulations

of the best simulation will better resolve, grillage 2 before using grillage 1 to satisfy the neutral axis constraint. This allows the simulation to incrementally reduce the cost objective function and reach a more fit final solution.

Though initial emphasis was place in studying and interpreting the structure of the Bayesian network, the conditional probability values of the simulations can be just as useful in guiding design decisions. Changes in bin values as the population is

Plate Thickness(mm)	5	10	15	20	25	30	35	40	% Penalized
Gen 0 (Pre-learning)	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	99.2
Gen 0 (Post-learning)	0.022	0.072	0.114	0.088	0.15	0.14	0.21	0.204	99.2
Gen 2	1E-5	1E-5	0.054	0.006	0.072	0.242	0.014	0.358	89.4
Gen 4	1E-5	1E-5	1E-5	1E-5	1E-5	0.33	0.106	0.564	34.8
Gen 5	1E-5	1E-5	1E-5	1E-5	1E-5	0.222	0.014	0.764	11.3
Gen 6	1E-5	1E-5	1E-5	1E-5	1E-5	0.03	0.004	0.966	12.9
Gen 7	1E-5	1E-5	1E-5	1E-5	1E-5	1E-5	1E-5	0.99	14.6

Table 4.3: Conditional Probability Table results on variable P6 for selected generations of Simulation 10. NOTE: values of 1E-5 indicate that no data points in the training set exist.

generationally advanced demonstrate the exploration of the optimizer in the design space. Using the most elite simulation as an example, problem knowledge may also be recovered from CPT analysis. Table 4.3 shows the bin probability values of the design variable P6. It is important to note that these values change generationally. In this analysis example, the network structure remains the same after initial learning is completed. By sampling the network to create new points, the structure is reinforced. Included in the table are 8 columns for the probability values of the 8 discrete plate thickness values. A final ninth column has been added for constraint violation indicating the percentage of designs penalized for violating one of the 13 constraints applied to the structure. Beginning by analyzing the constraint violation, clear increases in performance are noted when the simulation excludes plate thickness values lower than 30 mm beginning in generation 4. Beginning immediately in generation 0 after the initial structure learning process, probability is removed from thinner plate bins and is re-assigned to thicker plate bins. This continues until no data points are observed in the extremely thin plates, during generation 2 and extends to the moderately thick plates in generations 3 and 4. From this one may conclude that the local grillage 6 constraints require a plate thickness value of at least 30 mm for satisfaction under present design conditions.

Furthermore, in Table 4.3 a clear distinction develops at plate thickness values of 30

		Plate Thickness (mm) Grillage 2, P2							
		5	10	15	20	25	30	35	40
Plate Thickness (mm) Grillage 6, P6	5	9.09E-2	1E-5	9.09E-2	1.82E-1	1E-5	7.14E-2	3.63E-1	2.73E-1
	10	1.11E-1	2.78E-2	1.67E-1	8.33E-2	1.39E-1	1.94E-1	5.56E-2	2.22E-1
	15	1.05E-1	3.51E-2	7.02E-2	2.63E-1	7.02E-2	1.23E-1	1.40E-1	1.93E-1
	20	9.09E-2	2.95E-1	1.36E-1	6.82E-2	2.05E-1	2.27E-2	1.14E-1	6.82E-2
	25	1.73E-1	1.07E-1	6.67E-2	8.00E-2	2.00E-1	1.20E-1	1.07E-1	1.47E-1
	30	7.14E-2	1.57E-1	1.29E-1	1.14E-2	8.57E-2	1.29E-1	1.86E-1	1.52E-1
	35	3.81E-2	1.14E-1	1.81E-1	1.33E-1	9.52E-2	1.91E-1	9.52E-2	1.18E-1
	40	1.67E-1	1.08E-1	1.37E-1	9.80E-2	1.47E-1	1.08E-1	1.17E-1	1.18E-1

Table 4.4: Conditional Probability Table results on variable P2 at generation 0 post learning. NOTE: values of 1E-5 indicate that no data points in the training set exist.

and 40 mm. Though existing in a feasible space, the 35 mm thickness value remains unpreferred, likely due to the combination of stiffeners available in in the catalogue and the discrete span values resulting from the geometric length of the grillage the the discrete divisions caused by number of stiffeners in a grillage. This is an incredibly important finding, and if a design containing 35 mm plate in this location is selected, it will likely be unnecessarily expensive as defined by the given objective function. Instead solutions containing either 30 or 40 mm plate are better solutions and should be selected. In the final stages of the optimizer, we see the true refinement power of the BOA. In a span of 3 generations, the population is shifted from mixed values of 30 and 40mm in location P6 to a homogeneous value of 40 mm. This is clearly the most elite solution and the optimizer finds it very efficiently, where other optimizers may struggle to reach convergence.

With this insight, the ability to propagate evidence in the network becomes critically important. As previously mentioned, elite simulations selected P6 as a root variable. Therefore selection of a bin on this variable will have cascading effects on the design. Tables 4.4 and 4.5 contain the resulting conditional probability tables for variable P2. As previously mentioned resolution of this smaller grillage causes highly fit simulations to become elite. In Table 4.4 the post learning distributions are relatively uniform. The initial refinement towards thicker plate thickness values is clear, though

		Plate Thickness (mm) Grillage 2, P2							
		5	10	15	20	25	30	35	40
Plate Thickness (mm) Grillage 6, P6	5	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	10	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	15	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	20	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	25	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
	30	0.0909	1E-5	1E-5	6.06E-3	1E-5	0.0121	0.873	0.0182
	35	1E-5	0.226	0.283	0.264	0.0943	0.0189	0.113	1E-5
	40	0.145	7.09E-3	1E-5	0.0142	0.145	0.0213	0.262	0.104

Table 4.5: Conditional Probability Table results on variable P2 at generation 4. NOTE: values of 1E-5 indicate that no data points in the training set exist.

the refinement pressure is far less intense as compare to the grillage 6 plate thickness values. This is to be expected, as P2 has a much smaller overall length as compared to P6. Therefore the span values will be smaller and the corresponding local grillage constraints will be relaxed. This allows thinner values of P2 as compared to P6. The most relevant information is extracted from Table 4.5.

From the analysis of P6, by generation 4, thickness of either 30 mm or 40 mm are highly preferred by P6. In those corresponding rows of the CPT for P2, there are vastly different results. For P6 values of 30 mm plate, 35 mm plate in grillage P2 is almost exclusively needed, with 87.3% of training data points selecting this plate thickness combination. In contrast for P6 values of 40 mm plate, there is a much greater range of acceptable designs. Recognizing this difference has great implications. Should a designer believe the constraints governing the plate thickness value of P2 will change with an early stage perturbation in the the design model, such that 35 mm plate will become infeasible in grillage 2, solutions containing P6 values of 30 mm should be discarded. Though this is a relatively straightforward example it highlights the potential of the Bayesian network analysis to identify critical casual relationships within the design problem and gather information that may then be used by the designer.

4.3 Partial Derivatives

To better understand the importance of variables within the objective function, the entire formulation including external penalty application was used for partial derivative analysis. This analysis was conducted in the hope that large partial derivatives would in some way be linked to the network degree analysis, linking the importance in the network and the importance in the objective function. As stated in chapter 3 the cost objective function may be stated as:

$$C_T = C_P + C_S + C_w + C_E + C_I \tag{4.5}$$

Where

$$C_P = 2000 * 2.72 * P_{t_i} * L * B$$

$$C_S = 1.05 * 2000 * 2.72 * A_{S_I} * N_i * L$$

$$C_w = \frac{27.0 * 2 * N_I * L}{1.2}$$

$$C_I = \frac{27.0 * 2 * N_I * L}{1.5}$$

$$C_E = \frac{27.0 * 2 * N_I * L}{0.9}$$

Taking a partial derivative of the stiffener from the catalogue becomes problematic, as there are multiple dimensions being used by the objective function. There is no explicit derivative of the catalogue member, though the selection is tighly coupled to the projected area of the T stiffener. Instead only deriatives may be taken of the number of stiffeners N and the plate thickness, P_t. Examining Table 4.6 the partial derivative is independent of location, as each grillage may have 1-8 stiffeners ignoring geometry constraints. Therefore it is unlikely that one particular nodal value of N is preferred over another. The plate thicknesses however vary depending on the grillage breadth locally. Clearly the partial derivatives of grillages G_0 and G_5 have

Grillage No.	$\frac{d(f)}{d(N)}$	$\frac{d(f)}{d(P_t)}$
0	$11424{*}A_{s_i} + 282$	97920
1	$11424{*}A_{s_i} + 282$	16320
2	$11424{*}A_{s_i} + 282$	10880
3	$11424{*}A_{s_i} + 282$	10880
4	$11424{*}A_{s_i} + 282$	59840
5	$11424{*}A_{s_i} + 282$	108800

Table 4.6: Objective function Partial Devivatives for N and P_t in each grillage

the greatest potential to effect the design.

The addition of penalty constraints however muddle this seemingly straightforward result. The penalty function is described as:

$$P = R_k p_i^2 \tag{4.6}$$

Where

$$R_k = 10000$$

and either

$$p_i = 100\frac{p_{t_{min}-p_t}}{p_{t_{min}}}$$

OR

$$p_i = 100\frac{I_{NA_{min}-I_NA}}{I_{NA_{min}}}$$

Again with a simple algebraic substitution, the partial becomes:

$$-\frac{20000}{p_{t_{min}}}\frac{p_{t_{min}-p_t}}{p_{t_{min}}} \tag{4.7}$$

OR

$$-\frac{20000}{I_{NA_{min}}}\frac{I_{NA_{min}-I_NA}}{I_{NA_{min}}} \tag{4.8}$$

In both cases the penalized partial derivatives are based on the functional location of grillage as well as the actual value being assessed. This partial is however much larger in magnitude than the objective function, meaning the constraint will likely drive the optimizer.

From this partial derivative analysis, there are two critical takeaways. First, the unconstrained problem favors the N nodes controlling number of stiffeners, consistent with the findings of Rigterink et al. (2013). The stiffeners are critical to the cost of the structure and thus drive the problem. However when the constraints are added, the plate thickness becomes more important, effectively flipping the direction of influence. This flip will be observable in the network structure. Second, the size of the grillages as defined by the half section geometry controls the magnitude of the partial derivatives. Larger lengths L will cause larger partial derivatives. Therefore, Grillages 0, 4, and 5 are most critical for feasible designs.

4.4 Exploring Network Learning

4.4.1 Network Structure learning on a single Trap5 Problem

From this initial test case, the BOA case produced a vast amount of data, which can be overwhelming. There is a need to understand how the BOA responds to the decomposable portions of the objective function. To explore the networks created by the optimzer, it may be best to step back and examine a series of increasingly more complex and difficult problems. To do this, a methodical examination beginning with Pelikan's Decomposable work and transitioning to a series of stiffened T-Panels is presented. The work seeks to highlight how the optimizer creates networks first at the truly decomposable level and then at the global level as decomposable elements are combined to form a complete design.

The most eye-catching portion of Pelikan's work begins with the trap 5 problem. As previously discussed, this is a binary logic trap, with a size of 5 bits. This problem is interesting in that it has a strongly preferred local optima away from the global optima which will lead a gradient optimizer away from the global solution. The problem may be mathematically summarized as :

$$\begin{aligned} f(u) &= 5 \quad ifu = 5 \\ &= 4 - u \quad else \end{aligned} \tag{4.9}$$

where:

$$u = \sum_{i=1}^{5} x_i \tag{4.10}$$

When used the the BOA, this problem is represented by 5 variables or nodes each with discrete probability bins of 0 or 1 to be assessed by the aforementioned equations 4.9 and 4.10.

As you can see in figure 4.2 the problem is not smooth, causing both ends of the

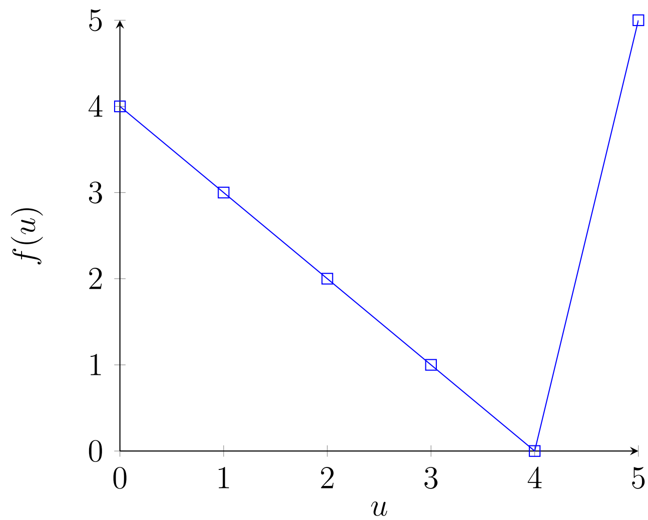


Figure 4.2: Graphical depiction of the Trap 5 problem. u is the sum of the binary bit values and $f(U)$ is the corresponding fitness value. Notice the global maximum value, $f(5) = 5$ is located away from the gradient dominated solution $f(0) = 4$ making the problem more difficult to solve.

problem to be preferred by a gradient optimizer and locating the global solution as far as possible from the local solution. Sampling of this problem is also an interesting challenge. The problem has 2^5 potential outputs, with $f(u)$ value of 4 and 5 occurring for only 2 of the 32 results. Therefore a high quality sample to capture this is needed. Both 4 and 5 point solutions must be in the training data, else the network is mapping a relatively random solution space (as location in the design variable vector is unimportant in this case).

To examine the network formation given a set of data, a series of 10 stoichastic simulations were conducted. Training data size continues to be an area of exploration. The designer seeks to examine a training set that is not random, else the network will lack edges due to the nature of the network scoring K2 metric. For a single 5 bit trap problem 100 random designs were created, of which a portion most elite designs were selected as training data. A sample of 4 heuristic parameters will be shown. These samples seek to recreate stages of the optimization process. Ideally the optimizer goes through 3 major stages: feasibility, refinement, and optimality convergence. The network edges at each stage will be examined.

The first simulation takes the training set and selects 50% of the population to train the network on. The results are reported in Table 4.7. Due to the random nature of the simulation it is difficult to visualize the edge data in compact way. Typically matrices are used to show edge connection. With 10 simulations to represent, this becomes difficult. Instead a tabular output is presented. When an edge exists in the network, the parent is recorded with a + symbol followed by its child. Likewise, the Child records a - symbol followed by its parent. In this way even number of sets must be reported in each line of the table. What we see from the initial sample is that the network edges are essentially random with no preferred form. This is to be expected as the sample is still relatively random and unrefined. There are 3 distinct structures to pay attention to in these tables. Chains within the network will be represented in elements that have both + and - symbols recorded. Downward facing Vs significant for Bayesian inference are represented by two or more + symbols for the same node. Likewise upward facing Vs, important to denote independence in

Sim No.	X1	X2	X3	X4	X5
1		+X3	-X2		
2		+X5	+X5		-X2, -X3
3	+X5			+X5	-X1, -X4
4	+X4, +X5			-X1, +X5	-X1, -X5
5					
6				+X5	-X4
7					
8		+X3	-X2		
9			+X4	-X3	
10	-X3, +X4, +X5	+X5	+X1, +X4	-X1, -X3	-X1, -X2

Table 4.7: Network edge representations of a single trap 5 problem at a population size of 100, with a selection pressure of 50% without addition of any additional highly fit solutions. Note the lack of defined structure. For notation, the + symbol indicates the node is a parent of the designated neighbor, the - symbol indicates it is a child.

networks are identified by two - symbols on the same node. In Table 4.7, no defined structure exists, as edges are haphazardly scattered.

To improve the quality of the sample and remove some of the randomness in the simulation the selection pressure is double. Instead only 25 designs of the 100 member population pool are selected for training. Table 4.8 shows the results of this change. What begins to emerge are a series of downward and upward facing Vs. This is an improvement as the algorithm begins to identify the bifurcation in results. Again though, this sample space is 32 members large and the 25 member training population is likely insufficient to capture enough globally fit solutions to fully resolve the network edges.

To combat this, the known globally and locally fit solutions, uniform 0 bits and uniform 1 bits are added to the training pool prior to selection. Now 25 % of the 110 members are selected, ensuring roughly 40% of the sample is high quality solutions. the results, shown in Table 4.9 highlight the single dominant structure, the downward facing V appearing in every network. This is ideal, as the bifurcation is controlled by the root node in the network. If the root node is chosen to be a value of 0, the entire design should uniformly be 0. Likewise if the root node is chosen to be 1, the entire design should be chosen uniformly as 1.

The final progression is to increase the selectivity of the training pool such that a

Sim No.	X1	X2	X3	X4	X5
1	+X4	+X4		-X1, -X2	
2	-X2, -X3	+X1	+X1		
3	-X2	+X1, +X5			-X2
4				+X5	-X4
5	-X2	+X1, +X3	-X2		
6	+X4	+X3, +X4, +X5	-X2	-X1, -X2	-X2
7				+X5	-X4
8		+X3	-X2		
9		+X4	-X2, +X4	-X2, -X3	
10					

Table 4.8: Network edge representations of a single trap 5 problem at a population size of 100, with a selection pressure of 25% without addition of any additional highly fit solutions. Note the number of edges in the network has increased considerably, as the sample is less random/more heavily pressured when selected. Additionally, downward facing V structures, denoted by 2 or more + signs for the same node, begin to appear. These network structures fit the trap problem as they allow influence from one node to be passed to another.

majority of the solutions are highly elite. To do this, the selection pressure is again doubled such that only 12.5% of the solutions are selected for the training data. The results are very encouraging, as the simulations converge to a single ideal structure.

To graphically visualize this transition, figure 4.3 shows representations of an average network at each of the 4 groups of parameters used. The transition from random to clearly defined and usable network form is stark. This trend likely exists in the refinement of a structural design network, where the quality of the data supplied to the optimizer controls the resulting quality of the network. As previously postulated, the network becomes more valuable for analysis when high quality training sets are used for network learning.

Sim No.	X1	X2	X3	X4	X5
1				+X5	-X4
2	-X2	+X1, +X3	-X2		
3	-X2	+X1, +X3	-X2		
4		+X3, +X5	-X2		-X2
5		+X3, +X5	-X2		-X2
6			+X4, +X5	-X3	-X3
7	+X4			-X1	
8		+X3	-X2		
9	-X3		+X1		
10			+X4	-X3	

Table 4.9: Network edge representations of a single trap 5 problem at a population size of 100, with a selection pressure of 25%. Now five solutions of uniform bits = 1 and five solutions of uniform bits = 0 are added. This begins to crystallize a distinct single network structure, consisting of multiple downward facing Vs originating from a single node.

Sim No.	X1	X2	X3	X4	X5
1	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
2	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
3	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
4	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
5	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
6	-X2	+X1, +X3, +X4	-X2	-X2	
7	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
8	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
9	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2
10	-X2	+X1, +X3, +X4, +X5	-X2	-X2	-X2

Table 4.10: Network edge representations of a single trap 5 problem at a population size of 100, with a selection pressure of 12.5%. Again five solutions of uniform bits = 1 and five solutions of uniform bits = 0 are added. This simulation seeks highly fit training data. The sample size is the same as that of tables 4.8 and 4.9. The highly fit data produces a uniform structure of a single root node with 4 children. This is incredibly useful to designers.

4.4.2 Network Structure Learning on Single T-Panel Problem

This study into the dynamics of network formation became incredibly insightful as focus was shifted to the decomposable portions of the marine structural design

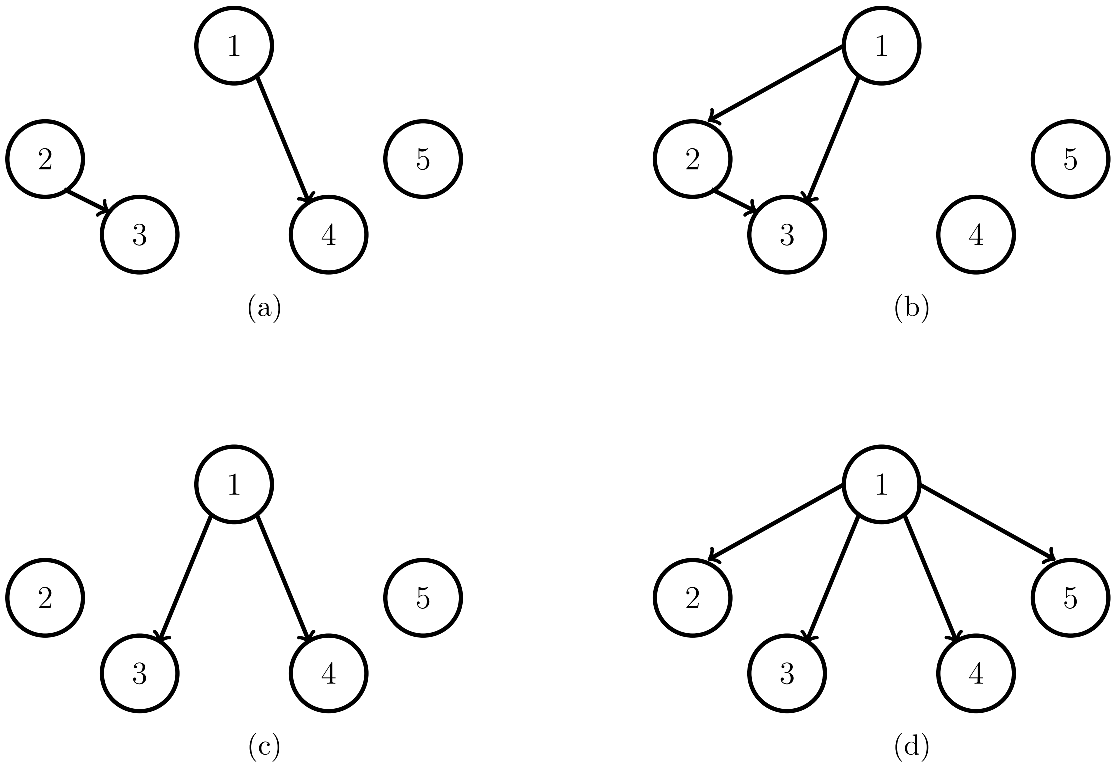


Figure 4.3: Comparison of the network structure as the quality of data is varied through selection and highly fit solution addition. simulations 4.3a, 4.3b, 4.3c and 4.3d correspond to the simulation data in tables 4.7, 4.8, 4.9 and 4.10 respectively.

problem. The fidelity and failure mode evaluations of marine scantlings are almost entire done at a loal or regional level requiring only small portions of the design. A study was conducted to better understand how the BOA might assign network edges to solve the local strength constraints. To do this the basic T-Panel structure representation is again used. In the explorations of the Trap5 problem the global optimal result was known, making it easier to ensure the network received high quality training data. Due to the size of the design space, this becomes far more difficult to repeat with the T-Panel design. Using the lessons learned from the Trap 5 experiments, the network edge learning algorithm was applied to a series of training data sets to determine if/how the optimizer would structure the variables. Simulations of 100 members and 1000 members were conducted, both at 50% selection to mimic the low quality high quality approach of the Trap5 research.

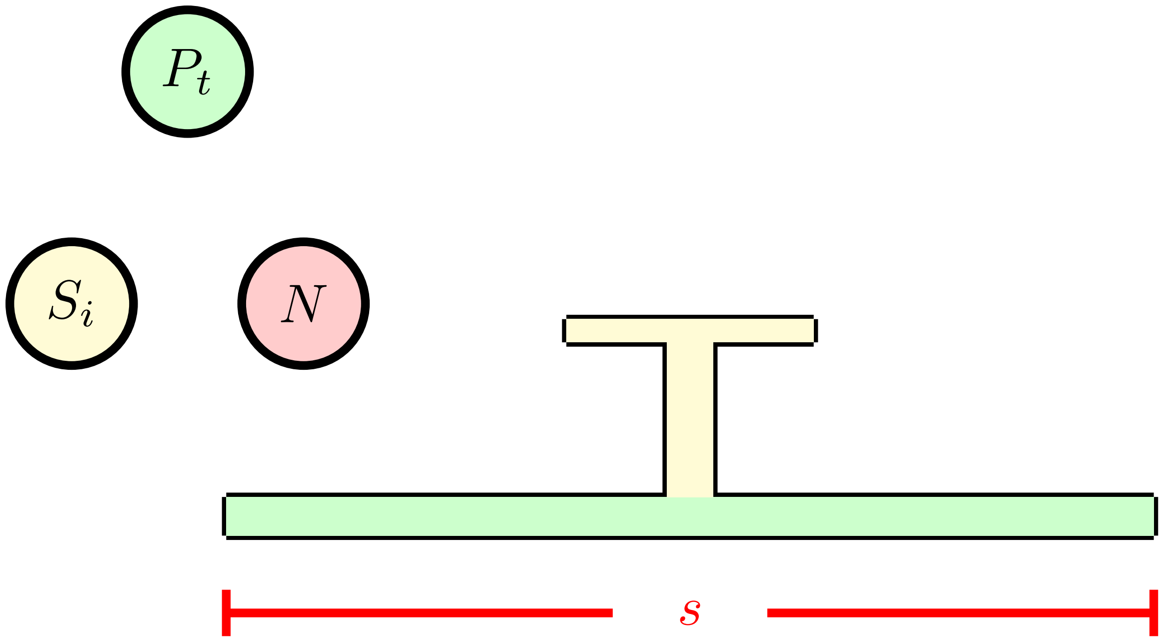


Figure 4.4: Representative T-Panel and its associated design variables. Color indicates the portion of the structure controlled by the variable. The number of variables N, controls the spacing used based on a fixed global grillage dimension.

To reiterate the ideal network is a distinct series of edges, somehow relating the cost function (Rahman and Caldwell (1992)) to the local decomposable constraints. Based on previous research into Rahman and Caldwells costing function it is known that volume of the plate as compared to volume of stiffener is preferred and will drive the cost of the individual panel. The strength constraints assess the minimum plate thickness required and the minimum panel moment of inertia. Both constraints require variables P_t and N, while only one requires the stiffener type S. In this way the network should seek to place either P_t or N as a root node, and it should assign an edge between the two variables.

4.4.3 Network Structure Learning on Multiple Decomposable T-Panel Problems

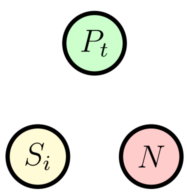
After comparing this training data, the natural progression is to move onto multiple individual stiffened T-panels. Again drawing upon the results of the Trap5

Sim No.	P_t	S	N
1	-N	-N	$+P_t$, +S
2			
3		-N	+S
4	-N	-N	$+P_t$, +S
5	-N	-N	$+P_t$, +S
6		-N	+S
7			
8			
9	-N	-N	$+P_t$, +S
10			

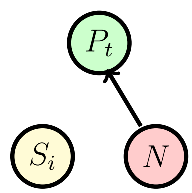
(a)

Sim No.	P_t	S	N
1	-N		$+P_t$
2	-N		$+P_t$
3	-N		$+P_t$
4	-N		$+P_t$
5	-N		$+P_t$
6	-N		$+P_t$
7	-N		$+P_t$
8	-N		$+P_t$
9	-N		$+P_t$
10	-N		$+P_t$

(b)



(c)



(d)

Figure 4.5: Comparison of network form as a result of training data. At left in 4.5a and 4.5c are the results of a simulation of 100 designs with 50% selection rate. Edges are relatively random, though 6 of the 10 simulations correctly identify N as a critical variable. At right in 4.5a and 4.5c are the results of a 1000 design member simulation at 50% selection rate. Note the uniformity of the edge results in this simulation correlated to the higher quality data. The simulation also identifies the two variables present in both constraints being applied.

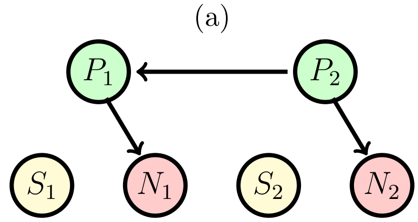
study and a single T-Panel study, mutliple T-Panels were used to create a network refinement study. The objective seeks to minimize the singular production cost of the panels while being constrained in both plate thickness and stiffener moment of inertia as if the T-Panel was an element of bottom shell plating.

What one observes is somewhat intuitive. The network is far less likely to favor a single structure as the data is less correlated. The K2 metric fails to identify strong conditional dependencies in each simulation. The one constant across the simulations is that the number of stifffeners N was preferred to be a root node. The mathematical reason for this is that a fixed shell width L can be divided n ways by the number of stiffeners N. This controls the volume of plate, the single greatest driver of cost.

It was fascinating to observe the 2 panel simulation change of root node from N to P_t. Suddenly win the large design space, feasibility becomes a much more important problem driver as opposed to the true objective function C_{prod}. The optimizer identifies P_t as the principle variable controlling the local decomposable constraints and moves it to the root to better influence the other design variables. This result is incredibly important as it gives insight into the change in network structure as hierarchy becomes involved. The network that best represents local structure may not be the network that best represents global structural solutions. This insight is exactly what a ship designer needs!

Sim No.	P_1	S_1	N_1	P_2	S_2	N_2
1	$+N_1$		$-P_1$	$+N_2$		$-P_2$
2	$+N_1$, $-P_2$		$-P_1$	$+N_2$, $+P_1$		$-P_2$
3	$+N_1$		$-P_1$	$+N_2$		$-P_2$
4	$+N_1$, $-P_2$		$-P_1$	$+N_2$, $+P_1$		$-P_2$
5	$+N_1$, $+P_2$		$-P_1$	$+N_2$, $-P_1$		$-P_2$
6	$+N_1$, $-P_2$		$-P_1$	$+N_2$, $+P_1$, $+S_2$	$-P_2$	$-P_2$
7	$+N_1$, $-P_2$		$-P_1$	$+N_2$, $+P_1$		$-P_2$
8	$+N_1$, $+P_2$		$-P_1$	$+N_2$, $-P_1$		$-P_2$
9	$+N_1$, $+P_2$		$-P_1$	$+N_2$, $-P_1$		$-P_2$
10	$+N_1$, $-P_2$	$-S_1$	$-P_1$	$+N_2$, $+P_1$, $+S_2$	$-P_2$, $+S_1$	$-P_2$

(a)



(b)

Figure 4.6: Resulting network form as a result of training data. To explore the multiple decomposable T-Panel problem, 20000 data points were generated. Then 10 % of that population or 200 members were used to create networks. The result of the 10 simulations are shown in 4.6a and the average network is shown in 4.6b

Theory may seem to indicate that a scantling value should control the failure response of the structure, but in the overall system, other scantling variables may have a greater impact. By selecting an optimal value for P_t the design is better predisposed to feasible portions and can thus be better optimized.

What was needed was a rigorous method of examining the decomposable/non-decomposable interaction within the network formation. As network size and complexity grows it can become difficult to methodically examine and interpret results. To conduct the analysis the variables were separated into sets contingent upon their section number. The sets were then checked for the number of interior edges and the number of exterior edges. It was hoped that this type of analysis would reveal the importance of specific structure vs. the optimality of the section as guided by the optimizer. In this way by removing the non-decomposable constraint and examining the results, the differences or lack thereof would be evident.

One could hypothesize two main scenarios, with results likely to blend the two extreme. Either a single portion of the structure would dominate the cost in each section, and the BOA LGA would connect these local members across section boundary, or the section, like the additive portions of the mathematical Trap5 problem, would become the key piece of information and the network would show a higher number of interior edges.

4.4.4 Network Structure Learning in a single section

By this point in the demonstration it should be clear that the quality of training data the optimizer sees is critically important to the accuracy of the resulting learned network. Chapter 5 goes in depth to develop a network for a single section solution that is of highest quality. The resulting network is reprodeuced here, in Figure 4.7. From the learning investigation done on single and multiple T-Panels and using the partial derivatives as support, some comments may be made explaining why the structure is thusly arranged and how this may correlate well to an optimal design.

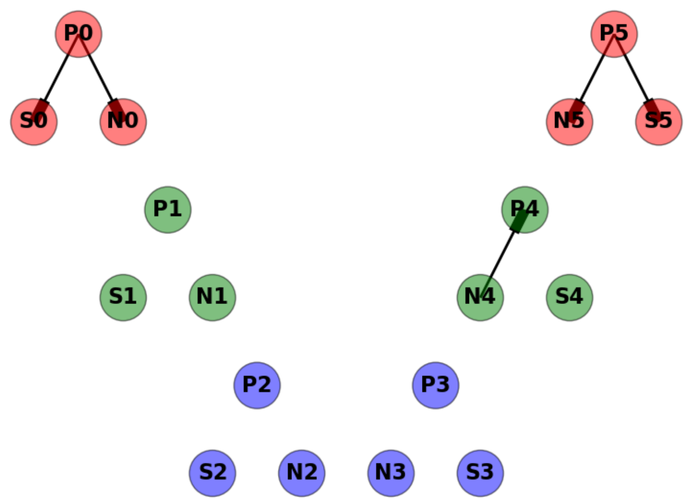


Figure 4.7: Taken from chapter 5, when using a high quality data set, the idealized preresentation of the single section network.

First, from the partial derivative analysis, the largest grillages 0, 4, and 5 are the only grillages to have edges connected. This related to the magnitude of their partial derivatives being far more significant than their neighbors. Next, there are no connections between nodes that are not in the same grillage definition. This was observed in the multiple T-Panel results, as the edge between grillages was oriented in both directions, depending on the randomness of the simulation. The section simulation does not struggle and effectively identifies the important grillages. Finally, in the highly constrained grillages 0 and 5, P_t dominates as the root node where in the smaller less constrained side shell grillage, the number of stiffeners becomes the root node.

4.5 Summary

The results from the BOA BN network analysis are encouraging, showing the value of the design process as a product. The BN creates a legacy object of the learning occuring in the design problem, even capturing temporal changes as evidenced by the conditional probability tables. By linking this temporal shift to locally relevant constraint satisfaction/violation it is clear that the optimizer is driving the design to the most fit feasible portions of the design space. Unlike other algorithms, the networks makes this change easily understood by the designer, and guides the designer quickly to the probability bins which are most successful for the given optimization problem.

The partial derivative analysis, though incomplete across the entire design variable vector, demonstrates the importance of plate thickness over the number of stiffeners. Though the local decomposable network favors the use of N as a problem root node, multiple linked T-Panel problems seem to instead favor plate thickness as a critical root variable. These insights are important to gather, but must be better incorporated into the design process. This sets the stage for Chapter 5. This seeks the culmination of this analysis, using the lessons learned to positively effect the design in the face of change or uncertainty.

CHAPTER 5

Application

5.1 Overview

Having demonstrated the Bayesian network created by the BOA captures the problem structures and dependencies within the network edges and conditional dependencies, it remains to extend this knowledge to application for design. The design knowledge must be incorporated to continue to improve the designers ability to quickly weigh and select high quality tradeoffs, and respond to unexpected changes in the design. The change in design paradigm focused on the use of this knowledge improves the ability of the designer to execute in potentially unknown and unexplored design spaces. To demonstrate this, two potential case study examples are developed. This first focuses on the designers conceptualization of the problem. It introduces a network as the working system model within the designers conscience. After creating potential networks to best represent the designers understanding of the problem, a pseudo-optimization is run to show how the results of the algorithm differ from a traditionally driven BOA learning process. These results highlight failure mode structures as the designer understand them rather than the global objective function.

Second, an idealized design scenario is created in which designers seek guidance on a potentially highly uncertain critical variables. In existing approaches to this situation, designers would often develop concurrent designs or decide which of the variable values was either most elite or most robust. Instead the conditional dependencies of the BOA allow the designer to examine multiple possible values of the solution simultaneously. To do this the variable bins remain uniform throughout the simulation ensuring a minimal amount of biodiversity is carried into each simulation.

5.2 Addition of Designer intent

5.2.1 Designer Specified Networks

The struggle with network scoring is a matter of correlation vs. causation. Many of the most basic and often most useful methods simply seek statistical correlation between variables to impose directed edges. Though these relationships are mathematically supported, it would be more appropriated to explore relationships that are founded on causality within the design fitness function being evaluated. The BOA's BNs allow the designer to better understand the problem they are solving and thus build designer intuition where none may exist.

To examine this, the previously described nearly decomposable problem is considered. At the global level, a series of constraints are linked across the decomposable pieces of the problem. With the test case of section moment of inertia, the designer can calculate the contribution of each scantling piece to the total value. Dimensions that are located far from the centroid, or are large in size will have the greatest impact on the value of global moment of inertia. A designer would prefer if these critical variables remain closer to the root of the network, so that their selection may influence variable selection in a cascading effect.

At the same time the local strength constraints must continue to be solved. To do this, networks must also reflect the local connections we see in constraint definition. consider a simple 4 sided box girder:

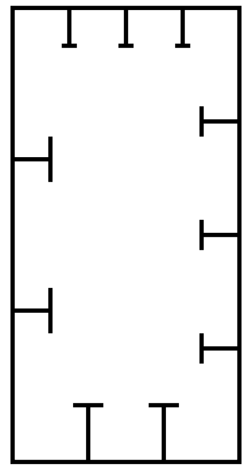


Figure 5.1: Nominal test box girder

GR_1	GR_2	GR_3	GR_4	GR_5	GR_6	Cross Grillage edges
$P_1 \to N_1$	$P_2 \to N_2$	$P_3 \to N_3$	$P_4 \to N_4$	$P_5 \to N_5$	$P_6 \to N_6$	$P_1 \to P_2$
$P_1 \to S_1$	$P_2 \to S_2$	$P_3 \to S_3$	$P_4 \to S_4$	$P_5 \to S_5$	$P_6 \to S_6$	$P_1 \to P_3$
						$P_1 \to P_4$
						$P_1 \to P_5$
						$P_1 \to P_6$
						$P_6 \to P_2$
						$P_6 \to P_3$
						$P_6 \to P_4$
						$P_6 \to P_5$

Table 5.1: Edge list for learned network

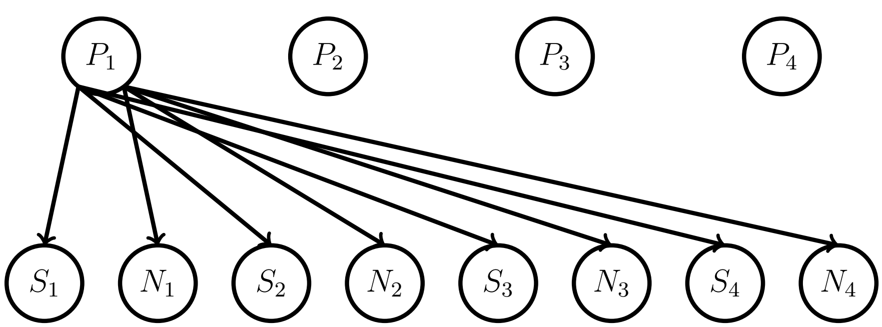


Figure 5.2: Average $K2$ Defined network. Though this network can be used to generate further highly fit design candidates, its srtucture appears to have little regard for the physics and mechanics the structure sees.

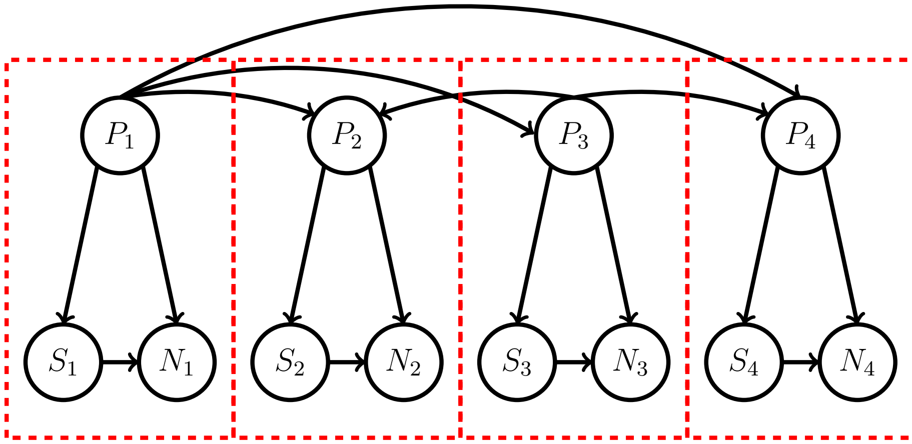
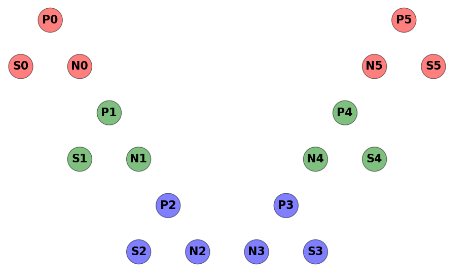


Figure 5.3: User preferred Visualization. In this example, the user defines the 4 cells for the 4 walls of the box girder separated by the dashed bounding lines. Since the Vertical bending moment is more extreme than the horizontal, edges issue from the P_1 and P_3 variables and lead to the P_2 and P_4 variables.

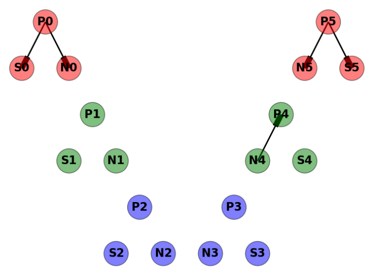
5.2.1.1 Network Score comparison

Creating a data set to compare a LGA learned network with a designer created one is an interesting taks. The data set needs to be sufficiently large and relatively elite, after the discussion of quality on the data in Chapter 4. To accomplish this, 500,000 randomly designs were created and evaluated. After evaluation, the designs were filtered selecting only solutions that had no penalty violations. In this manner the design training data will be sufficiently elite, while still remaining random and unbiased from any selection method outside a feasibility check.

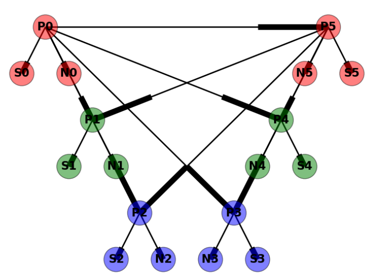
From a random sample roughly 1-2% of designs pass all constraint evaluations, so the selected training sets were approximately 1,000 members large. Table 5.2 shows the log gamma K2 scores of the 5 data sets for both the LGA learned and Designer specified network from figure 5.4. The results show that in all 5 simulations, the LGA constructs a more fit network than the designer specified network. The scale of the K2 metric is difficult to assess as it is population dependent, with smaller sample sizes having larger K2 scores due to the parent factorial term in the denominator of the value. What is clearly apparent from the edges of the learned network is that the



(a)



(b)



(c)

Figure 5.4: Base Network is shown in 5.4a, with the local substructure color coded, the BOA defined network is shown in 5.4b and the designer Designer defined network is shown in 5.4c. Note for the K2 driven learning process the network has less edges between grillages and more within grillage groups.

Sim No.	Data Set Size	LGA Learned	Designer Specified
1	1068	-37090.0	-37694.3
2	1011	-35093.2	-35650.7
3	1071	-37190.1	-37780.7
4	1043	-36201.0	-36787.8
5	1036	-35980.0	-36576.1

Table 5.2: Single Section comparison of K2 scores for LGA and designer created networks

designer has overdeveloped the network. Additionally, the learning process confirms the decomposability of the problem.

Consider how these results could be used for guidance of design study and focus. The designer without examining the topology and constraints in depth already knows something about the problem. The BOA identifies the largest grillages in the wet deck, strength deck, and side shell as the only nodes in the design requiring edges. Occasionally, the fourth largest grillage on the inner side of the seal is identified, but without regular consistency. Clearly these grillages drive the production cost of the design. Highly fit designs have high conditional dependencies within these local grillages.

When using this information to compare cost tradeoffs in the smaller hard chine wall geometry, variabilities effect is shown to be less impactful on the cost of the vessel. Spending slightly more money up front on these grillages to improve structural performance as measured by metrics like blast resistance or fatigue life for example may be well justified as these components have less impact on the total cost. Consider how often major studies have been launched attempting to clarify the true effect of fatigue on optimally designed structures, when it may in fact be more appropriate to simply improve the fatigue resistance of the scantlings. This network result gives value to that assertion. Additionally, the designer is able to tweak their conceptualized design model based on the feedback score of the network. In this scenario, the designer created network appears to be overconnected, with edges between grillages hurting not

Sim No.	K2 Score	Edge Operation
1	-37278.9	$P_0 \to S_0$
	-37207.5	$P_5 \to S_5$
	-37148.4	$P_5 \to N_5$
	-37116.9	$N_4 \to P_4$
	-37101.5	$N_4 \to S_4$
	-37090.0	$P_0 \to N_0$
2	-35239.3	$P_0 \to S_0$
	-35181.7	$P_5 \to S_5$
	-35141.7	$P_5 \to N_5$
	-35103.3	$N_4 \to P_4$
	-35094.7	$P_0 \to N_0$
	-35093.5	$N_5 \to S_5$
3	-37377.5	$P_0 \to S_0$
	-37303.4	$P_5 \to S_5$
	-37267.2	$P_5 \to N_5$
	-37234.7	$N_4 \to P_4$
	-37212.9	$P_0 \to N_0$
	-37200.1	$N_1 \to P_1$
	-37190.1	$N_4 \to S_4$
4	-36390.76	$P_0 \to S_0$
	-36315.11	$P_5 \to S_5$
	-36261.1	$N_4 \to P_4$
	-36219.2	$P_5 \to N_5$
	-36202.4	$P_0 \to N_0$
	-36201.0	$N_5 \to S_5$
5	-36153.2	$P_0 \to S_0$
	-36082.8	$P_5 \to S_5$
	-36032.0	$P_5 \to N_5$
	-35993.1	$N_4 \to P_4$
	-35984.2	$N_4 \to S_4$
	-35981.6	$N_1 \to P_1$
	-35980.0	$P_0 \to N_0$

Table 5.3: Edge addition steps of the LGA learned K2 network. Highlighted in yellow are the 2 identical edge additions that occur in the same order for all simulations. Clearly the plate thickness and number of stiffeners on the 2 largest grillages drive the problem.

helping the K2 score of the vessel. There is almost no need to connect grillages based on the resulting learned networks. By modifying the designer specified network to reflect this understanding the K2 score can again be calculated to see if the designers understanding of the network has been improved. This is valuable when used to build the mental model and understanding of the designers. The cause and effect style of feedback allows for the implementation and testing, leading to more well reasoned design decision making.

5.2.2 Removing Network Structure Learning

In Chapter 3, a large portion of the problem exploration was focused on the speed and accuracy of the network learning step in the algorithm loop. Though the designer specified network has a lower total network correlation score, it clearly gives a somewhat good fit of the design problem. For this reason, the designer specified network was used to propagate optimization runs, effectively removing the effect of network learning from the simulation and allowing only the conditional probability tables to guide the simulation. This resulting algorithm is fast, as simulations that would require minutes to hours computer wall time for both a a complete BOA require only a matter of seconds to complete. Figure 5.5 shows 10 random seed simulation results for population sizes of 200, 500, 700 and 1000 members. As is shown, even without network learning, the simulations with 1000 members converge within the threshold as set by the effort study. The effect of quality of network training data is once again highlighted as the mean fitness of the various population sizes vary greatly.

Figures 5.6 and 5.7 show the behavior of the smallest and largest population sizes 200 members and 1000 members as the algorithm attempts to remove constraint violations. The larger data set sample is much more narrow banded in its removal of designs violating both local decomposable constraints and the problem wide non-decomposable constraint. Particularly in Figure 5.7, the large size simulation stays

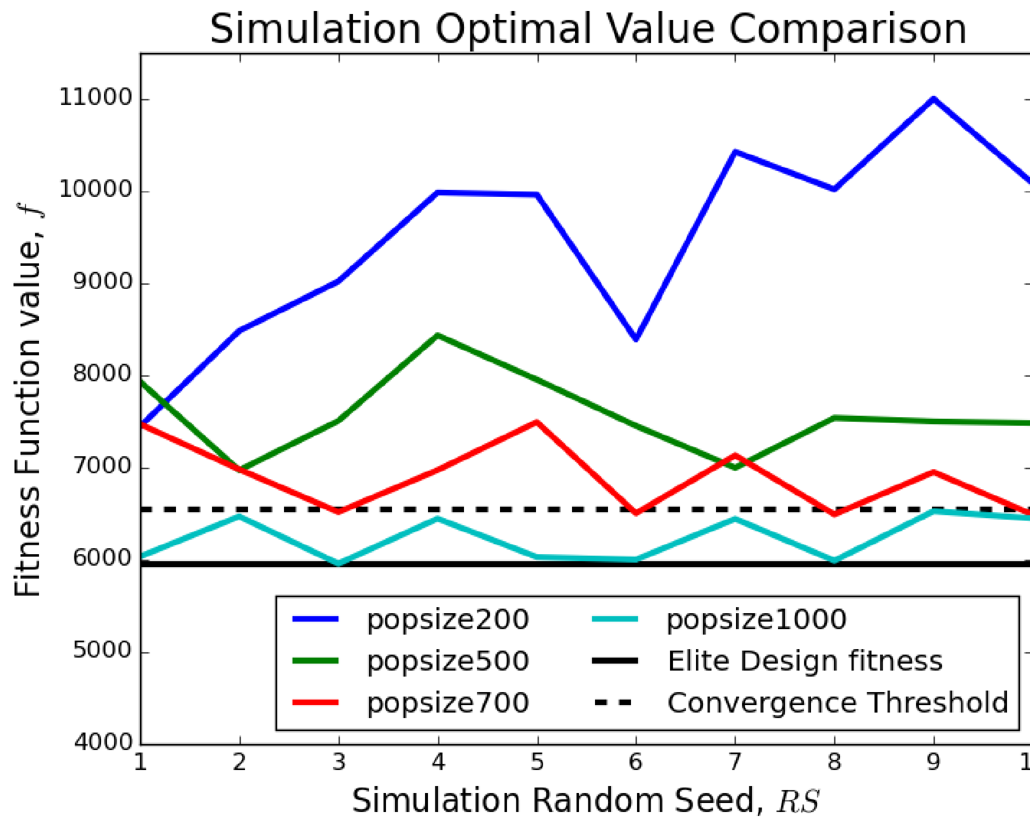


Figure 5.5: Comparison of the simulation fitness values for varying sizes population simulations of designer specified network. The elite design value is shown in the solid black line, with the threshold represented by the offset dotted line.

close to the active constraint edge, with a low percentage of designs violating the constraint in any given generation. The smaller simulation however lacks the data to best resolve the constraint boundary, instead simply seeking local violation removal and predestining the design to less fit regions of the design space.

Again this proves that though more fit networks exists, and have been identified with the LGA comparison, a good fit, can be used for optimization purposes. This strengthens the results in Chapter 3 which proposed a bounding limit to the efficiency of a BOA optimization as compared to the SOGA.

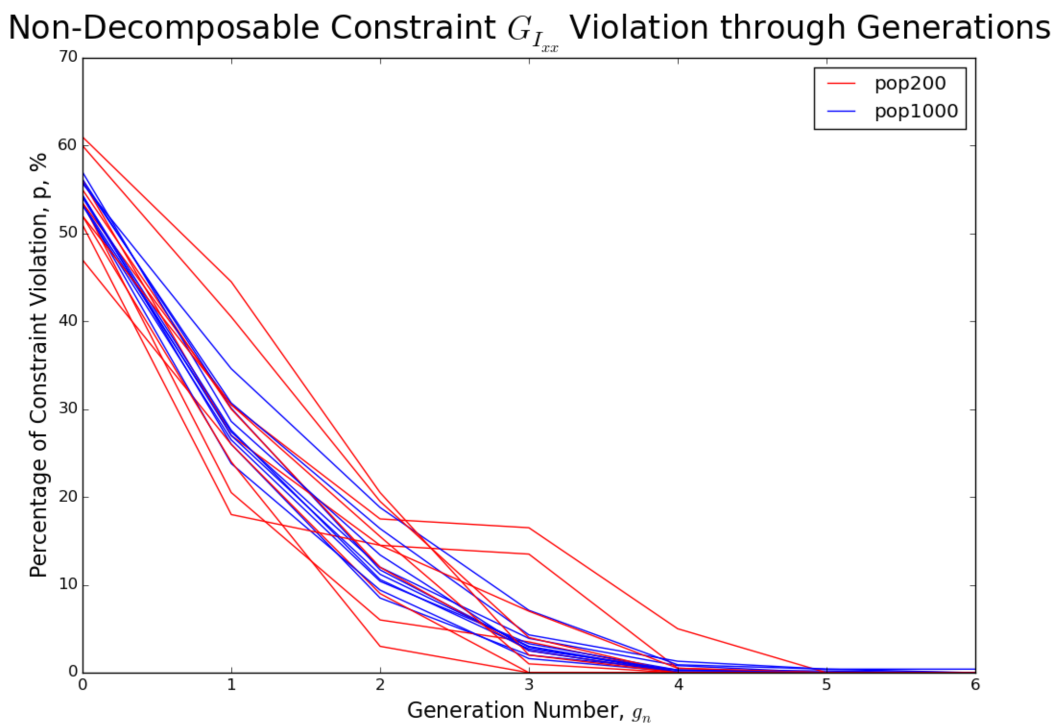
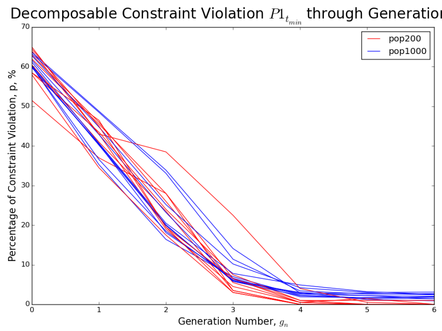


Figure 5.6: non decomposable constraint

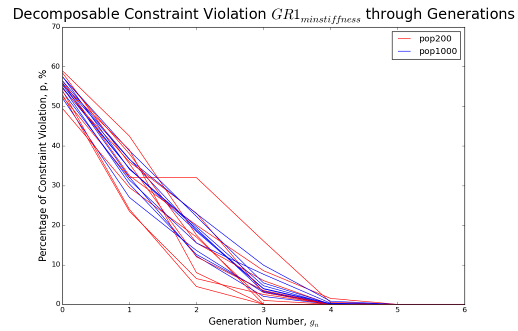
5.2.3 Stages of Convergence

As previously proposed, the algorithm appears to proceed through 3 critical stages of convergence during each simulation. First, the algorithm seeks to push the design candidate population entirely into the feasible region. This can be particularly difficult on a problem similar to the proposed case study which was highly constrained. A number of design solutions from a randomly sampled population violated at least one of the decomposable constraints while others violated the non-decomposable constraint. By comparing the constraint violation of the population on critical constraint functions, one can be certain the algorithm is exerting pressure on the population to become feasible.

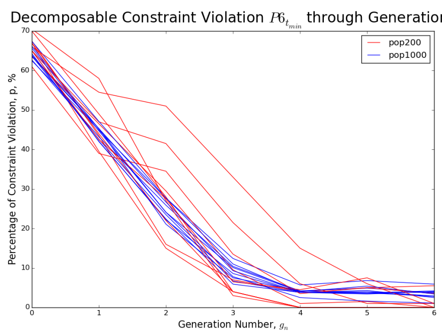
Once the population has entered the feasible region of the design, it begins to negotiate the tradeoffs within the design space. This span of generations will become critically important as the designer seeks to use the BOA for design guidance and



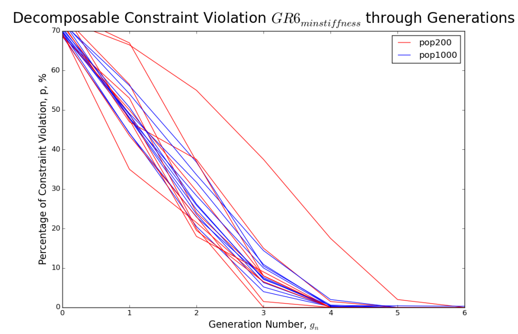
(a)



(b)



(c)



(d)

Figure 5.7: Presentation of the local decomposable constrain violations, Grillage 0 plate thickness at upper left 5.7a Grillage 0 T-Panel moment of inertia at upper right 5.7b Grillage 5 plate thickness at lower left 5.7c and Grillage 5 T-Panel moment of inertia at lower right 5.7d

insight, as discussed at length in Chapter 4. Any changes in the network structure or even within the conditional probability tables signal important tradeoffs within the design variable space.

Finally, the BOA pushes the design solution towards a single point, inflating the conditional probability tables in certain design variable bins, until the population reaches the expected convergence criteria. In many cases this is selected to be a lack of variation on the population, whether it be statistically in the sample, or lack of change over multiple generations.

Sim No.	Data Set Size	LGA Learned	Designer Specified
1	890	-62147.3	-61854.5
2	865	-60540.1	-60199.0
3	863	-60410.7	-60107.9
4	909	-63541.2	-63198.8
5	876	-61236.9	-60945.5

Table 5.4: Two Section comparison of K2 scores for LGA and designer created networks

5.2.4 Multi-Section Network Score Comparison

Having completed the analysis of K2 scoring on a single network, this case study was expanded to a 2 section problem. The larger scope should identify inter-sectional relationships that are falsely identified by the learned network. To begin, the designer specified network must again be created. In this case, the single section network representation as shown in Figure 5.4c will be replicated in the second section . There will be no addition of edges between section 1 variables (nodes with suffix 0-5) and section 2 variables (nodes with suffix 6-11). If a designer were to evaluate this design problem, there is no reason given the considered structural rules that section 1 and section 2 should interact, and the designer in this hypothetical case recognizes this.

Again data sampling becomes an issue, as an entire feasible population was needed to produce quality networks. With the combined effect of a 2 section problem, the total sample population was expanded to 2,000,000 member from which roughly 900 members were entirely feasible. Table 5.4 shows the training set size and the respective K2 scores for the networks. Note the scores are expected to increase due to the additional nodes in the new problem formulation. Again it is demonstrated that the LGA exceeds the network scores as created by the designer.

In a similar manner, the edge addition steps of the two section network learning process are shown in Table 5.5. This tables shows the edges added by the LGA in each iteration of the learning process, along with the improved K2 score for each edge. The final row of each simulation shows the last "best edge addition", with a score that is less than the existing maximum score, terminating the algorithm. Again, outside one or two edges, a common set of edges exist to best fit the training data of the 5 different random sample simulations. This result seems to support the added level of decomposability in the multi-section problem, and signals the decomposable nature of the problem. the LGA also identifies the similar structure of the problem in each section, as edges with suffixes 0 and 6 are the duplicate grillages in sections 1 and 2.

Sim No.	K2 Score	Edge Operation
1	-62147.3	$P_6 \rightarrow S_6$
	-62087.2	$P_0 \rightarrow S_0$
	-62030.4	$P_{11} \rightarrow S_{11}$
	-61979.5	$P_5 \rightarrow S_5$
	-61944.8	$N_{10} \rightarrow P_{10}$
	-61914.8	$P_5 \rightarrow N_5$
	-61889.9	$N_4 \rightarrow P_4$
	-61865.1	$P_{11} \rightarrow N_{11}$
	-61857.6	$N_1 \rightarrow P_1$
	-61854.5	$N_4 \rightarrow S_4$
2	-60540.1	$P_6 \rightarrow S_6$
	-60477.3	$P_5 \rightarrow S_5$
	-60417.7	$P_{11} \rightarrow S_{11}$
	-60366.9	$P_0 \rightarrow S_0$
	-60325.6	$P_5 \rightarrow N_5$
	-60287.6	$N_4 \rightarrow P_4$
	-60250.3	$N_{10} \rightarrow P_{10}$
	-60224.5	$P_{11} \rightarrow N_{11}$
	-60215.6	$P_0 \rightarrow N_0$
	-60209.5	$P_6 \rightarrow N_6$
	-60204.9	$N_7 \rightarrow P_7$
	-60201.9	$N_1 \rightarrow P_1$
	-60199.0	$N_{10} \rightarrow S_{10}$
3	-60410.7	$P_0 \rightarrow S_0$
	-60343.0	$P_6 \rightarrow S_6$
	-60293.0	$P_{11} \rightarrow S_{11}$
	-60244.7	$P_5 \rightarrow S_5$
	-60205.8	$N_{10} \rightarrow P_{10}$
	-60173.6	$P_{11} \rightarrow N_{11}$
	-60146.1	$N_4 \rightarrow P_4$
	-60125.7	$P_5 \rightarrow N_5$
	-60115.7	$P_6 \rightarrow N_6$
	-60111.8	$N_4 \rightarrow S_4$
	-60109.4	$P_0 \rightarrow N_0$
	-60107.9	$N_1 \rightarrow P_1$
4	-63541.2	$P_6 \rightarrow S_6$
	-63464.0	$P_0 \rightarrow S_0$
	-63391.4	$P_{11} \rightarrow S_{11}$
	-63335.9	$P_5 \rightarrow S_5$
	-63291.5	$P_{11} \rightarrow N_{11}$
	-63262.2	$P_5 \rightarrow N_5$
	-63234.4	$N_4 \rightarrow P_4$
	-63209.5	$N_{10} \rightarrow P_{10}$
	-63200.2	$P_6 \rightarrow N_6$
	-63198.8	$P_0 \rightarrow N_0$
5	-61236.9	$P_0 \rightarrow S_0$
	-61175.5	$P_6 \rightarrow S_6$
	-61124.4	$P_{11} \rightarrow S_{11}$
	-61076.6	$P_5 \rightarrow S_5$
	-61038.4	$P_{11} \rightarrow N_{11}$
	-61004.3	$P_5 \rightarrow N_5$
	-60983.4	$N_{10} \rightarrow P_1 0$
	-60965.1	$N_4 \rightarrow P_4$
	-60954.9	$N_{11} \rightarrow S_{11}$
	-60947.1	$P_6 \rightarrow N_6$
	-60945.5	$N_5 \rightarrow S_5$

Table 5.5: Edge addition steps of the LGA learned K2 network. Highlighted in yellow are the 2 identical edge additions that occur in the same order for all simulations. Clearly the plate thickness and number of stiffeners on the 2 largest grillages drive the problem.

5.3 Robustness and Change

Current design focuses have developed many sophisticated strategies for handling design uncertainty and change. Some strategies seek to mitigate the process, while others bound the effects and still others resolve the uncertainty directly. The BOA shows potential to add a new approach in response to unexpected changes in the design. To do this a simulation scenario is created.

Suppose designers remain unsure as to the pressure slamming loads that are likely to occur in the side shell of the vessel. Hydrodynamicists can give approximate mean values but have yet to quantify the statistical distributions of the loading parameters. As a result it is difficult to determine the required span needed to resist the loading. A tradeoff exists in the number of stiffeners used and the cost as already explored in the efficiency studies. The optimizer would prefer to use thicker plate and fewer, smaller stiffeners, but the designer lacks the intuition in a new problem space. How many stiffeners are most optimal for one value of plate thickness as opposed to another. Additionally, in the early stage of the design requirements and operational envelopes may potentially change as the design evolves.

Worst of all, the designer has no intuition as to how to respond to these changes. First principles approaches to structural response give a general notion as to the most likely solution but complicated non-explicit structural response functions interact to create a design space the designer does not understand. To counter this we again turn to the Bayesian Network and the product it produces. Conditional dependencies are well suited for this scenario as inserted evidence into the graph can be used to update the probability of related design variables. In this case, the node corresponding to the number of stiffeners becomes important in the examination of results.

In a standard approach to this problem the uncertainty of the loading could be addressed in any number of ways. Instead with the BOA, the new approach will leave the number of stiffeners on grillage 5, N_5, as a random value, allowing designs with each of the discrete number of stiffeners to be propagated to the solution. Then when a decision must be made the conditional dependencies of the variable will be used to dictate the final design solution. The BOA will prefer to remove inefficient values of N_5 as the evolutionary mechanisms are employed. To counter this, the optimization algorithm is adjusted to guarantee sample diversity on the N_5 node.

5.3.1 Case study Formulation

Again the simple grillage representation will be used to create a nominal structure for simulation. The critical difference in the way the BOA will be used is the change to the selection operator. In a standard optimization the most fit data is selected to be used by the optimizer. For this simulation population diversity must be artificially promoted to keep the entire variable tradeoff range available to the optimizer. Therefore the intial generation occurs as follows:

- Generate Initial population according to uniformly random distribution
- Score population based on designs
- For each discrete bin of the variable being studied:
  - Sort population first according to bin
  - From each bin attempt to select 5 % of the total population
  - If less than 5% of the bin exists in the population, select total bin sample
  - Fill remaining entries of candidate training pool with remaining most fit members
- Train network on design diverse population
- Simulate replacement members from newly trained network

In this way, the training subset is guaranteed to be diverse. With a selection percentage of 50%, most of the data point sampled will be selected for their diversity and not for their elite fitness. In effect 8 different optimizations are occurring within a single run of the optimizer, but the BOA does so in a way that is far more efficient. With the proposed strategy, the number of candidate designs in each bin of N_5 will never fall below the total in the initial simulation. However, the bins that have more than 5% of the total generation population carry the most elite designs.

5.3.2 Case Study Results

As in other case studies a series of simulations were conducted to test the stochastic nature of the process. At each generation of the BOA, the network was captured and stored. Additionally, for this study, since the number of stiffeners was of interest, its conditional probability table was studied..

In Figures 5.8, 5.9, 5.10 and 5.11 the generation 0, 2, 5, and 8 conditional probability tables are shown. These conditional probability tables guide the designer, informing optimizer decision making when selecting and altering variables.

5.3.2.1 Interpreting Conditional Probability distributions on the key variable

From these combinations of network structures and conditional probability tables, a clear story emerges on the path of the optimization. In figure 5.8, the population is as it should be random. The algorithm shows the genetic diversity of the variable of interest, N_4 is conserved, as the distribution is close to uniform. As the population remains random, the nodes are still uncorrelated and no edges appear in the network. At this stage, the constraint satisfaction of the existing population is low and as has been discussed, the quality of results of the network is suspect.

Moving forward 2 generations, the results in Figure 5.9 begin to show how the optimizer moves the population towards the more fit region. The first most important edges have been added, correlating the largest dimensions f plates to local variables, as denoted by color. What is interesting, is that by changing the sampling procedures on N_4, the N_4 - S_4 relationship now has an edge describing its conditional dependence, where a most fit simulation might not. This edge causes the topology of the N_4 CPT to change, as is shown in subfigure 5.9c. Now the CPT shows a two dimensional space, where the key diagonal has a concentration of highly fit design solutions. This agrees with tradeoff behavior observed in the relationship between span and plate thickness. Above the diagonal, the design solutions violate the constraints and the space is sparse. Well below the diagonal, the designs have surplus material and the cost is too expensive. At the same time, the root variable P_4 shows the same probablistic skewing towards the higher values of plate thickness, to best resolve the problem.

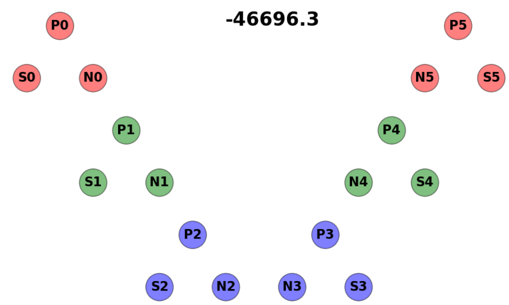
This space becomes more refined as the generation number increases. The diagonal becomes a well defined plateau in the CPT space, with one side failing to meet strength constraints and the other failing to be fit enough. The tradeoffs in the design space become readily apparent in different slices of N_4 values, as the conditioned P_4 CPTs begin to vary significantly. Specifically, generation 5 shows excellent insights into the design space dynamics. As shown previously, this stage of the optimization has reached a sustained level of constraint satisfaction while still remaining genetically diverse. As a result, the network is incredibly complex, as shown in figure 5.10a. As has been discussed in a prior chapter, it identifies a large portion of the local structure in the large grillages of the design problem. In this case however, the variable of interest, N_4 is well connected to a majority of the rest of the graph, through edges to its child, S_4. Through inference a designer decision on N_4 will propagate to the rest of the network. Variable P_4 begins to be highly refined as the distribution is now significantly weighted towards larger plate thicknesses, but this fails to capture the entire story. The remaining small thickness values exist tied to solutions with large numbers of stiffeners. Again this captures the classic tradeoff between number of stiffeners and plate thickness, and is the exact relationship a designer might wish to model to make informed design decisions.

The final set of graphs shown model generation 8 (Figure 5.11). At this point the simulation begins to enter more elite territory, and the resulting network (Figure 5.11a) begins to become more complex and has less first principles association. The variable N_4 still maintains its ability to influence a large portion of the design, but it occurs through a long chain of inference. Though the results in this generation are more elite, the tradeoff exploration is less exciting as the population begins to lose genetic diversity. The result is poorer CPTs. The previously described plateau in figure 5.11c now has incredibly steep sided walls. High quality solutions remain in this narrow probability band, and without the genetic diversity preserving operator on N_4, this band would collapse even tighter. The CPT for P_4 shows the lack of existing feasible solutions at plate thickness values of 5 and 10 mm.

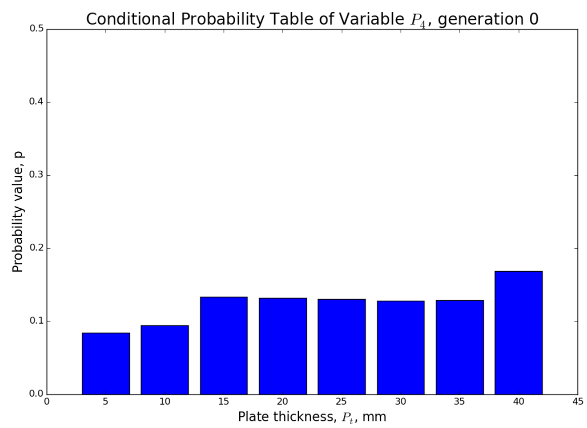
5.3.3 Design solutions

This information though quite interesting is not yet accessible enough for designers to better interpret the design space. Instead, evidence inference must be used in the network to best show tradeoffs. By treating a design variable value as evidence, its effect can be propagated to its neighbors in the network. For this example, the genetic selection modification leads to a network structure in generation 5 that is highly connected, and from what has been presented in previous chapters, also a good fit structure. Inserting evidence into node N_4 will effect 11 other nodes, meaning 66% of the design space will be altered by propagated influence. In Figure 5.12 the cause and effect propagation can be seen in the color coded advancement of information. The designer can condition a value into the N_4 node, and see cause and effect take place. This is the most useful result of the uniformly distributed change to the distribution of the node. A designer can quickly weigh the differences between a design with a side shell stiffened by 2 stiffeners or 3 and see the potential implications of such a decision.

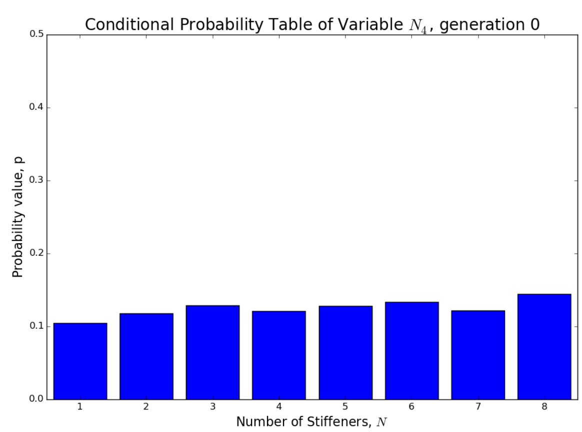
This experimentation would be impossible in other heuristic optimization, other than fixing the value of the node in question. Even if multiple values with multiple random samples are conducted to get a basic idea of the fitness of the parameter, the second and third order propagated effects would be impossible to decipher. Now the designer can produce both an optimal result and at the same time gather design space tradeoff information.



(a)

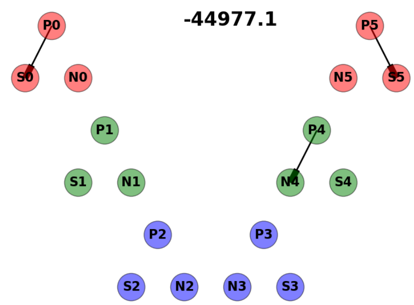


(b)

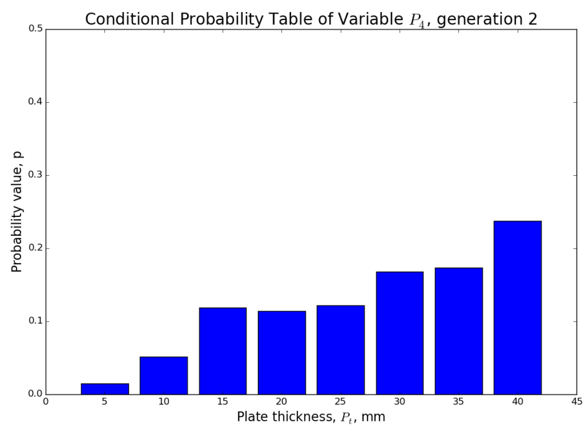


(c)

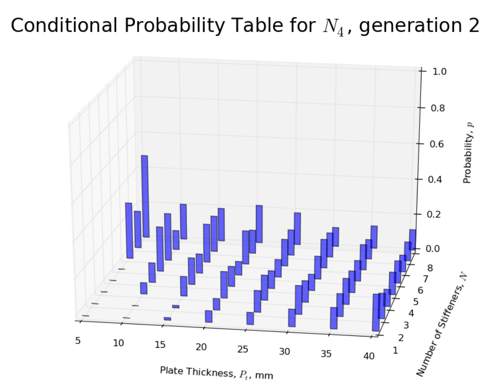
Figure 5.8: Generation 0 Network in 5.8a Plate CPT in 5.8b Number of Stiffeners CPT in 5.8c



(a)

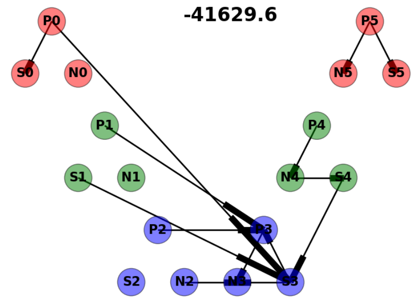


(b)

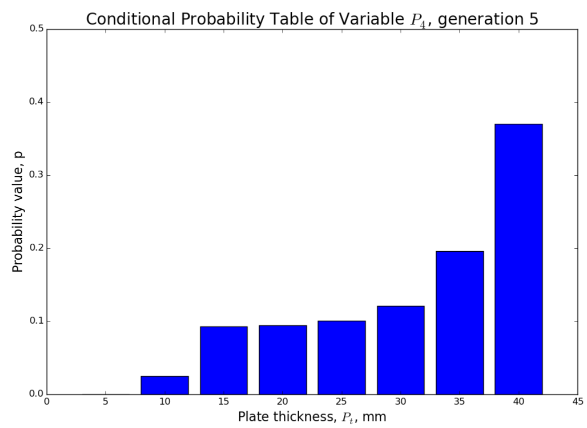


(c)

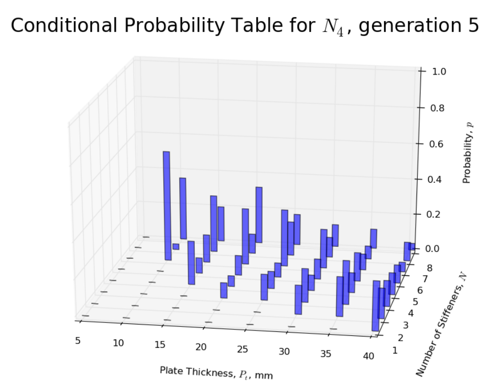
Figure 5.9: Generation 4 Network in 5.9a Plate CPT in 5.9b Number of Stiffeners CPT in 5.9c



(a)

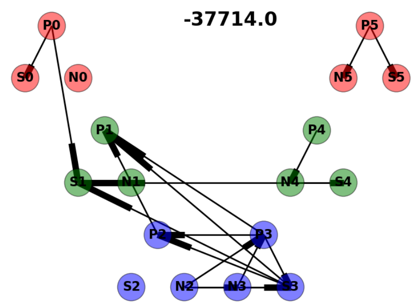


(b)

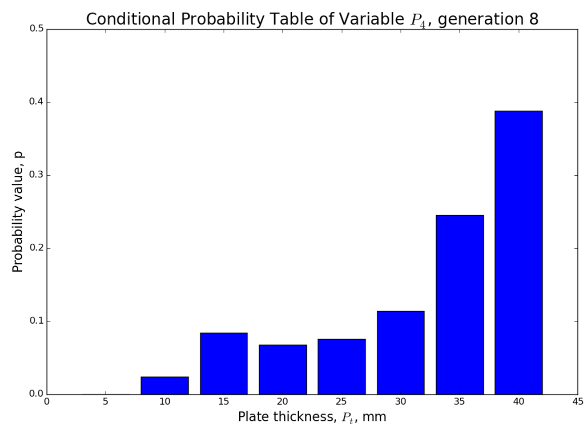


(c)

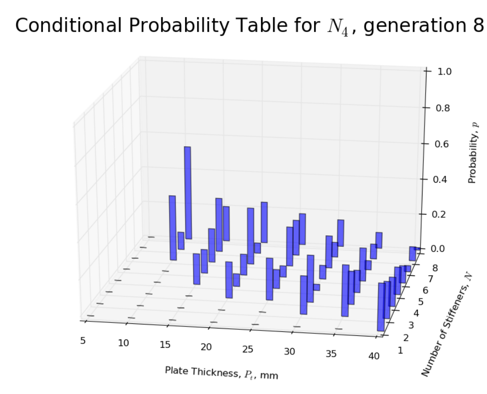
Figure 5.10: Generation 5 Network in 5.10a Plate CPT in 5.10b Number of Stiffeners CPT in 5.10c



(a)



(b)



(c)

Figure 5.11: Generation 8 Network in 5.11a Plate CPT in 5.11b Number of Stiffeners CPT in 5.11c

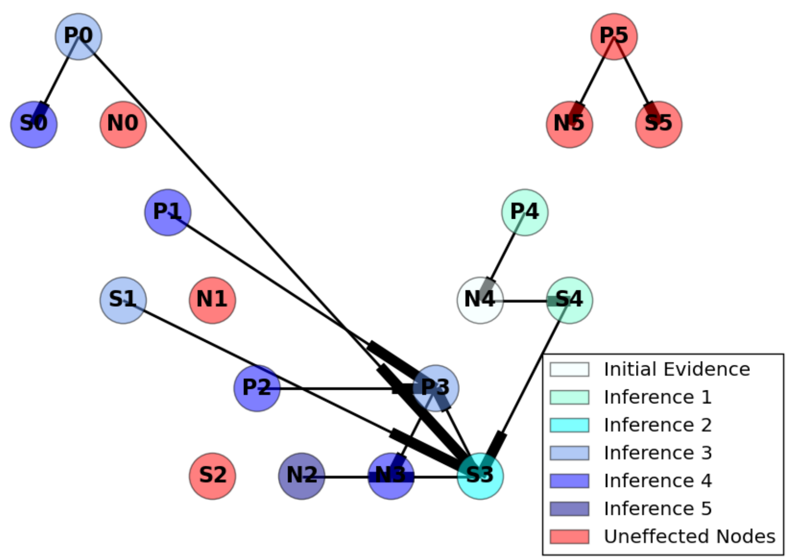


Figure 5.12: Inference progression for selected Generation 5 network. Initial evidence is inserted into node N_4 and through subsequent conditioning, the information passed throughout the entire design variable vector. Color shows the step in which information reaches the node locally, with red nodes being unconnected to the node of interest.

5.4 Summary

To demonstrate the potential use of the BOA and its BNs, two specific scenarios were presented using the engineering example case study. Each scenario produces optimal design solutions as any other optimizer would. However using the Network analysis approaches outlined, the designer gains a more advanced understanding of the design solution space. As outlined in the literature review, this understanding leads to better decision making in early stage design where the designer can largely effect the quality of the design solution.

In the first scenario, the designer uses mental modeling and appropriate network representations and compares these conceptualizations to the artificial intelligence in the BOA. Through this comparison and subsequent changes to the mental model, the designer can more completely understand the design variable interaction. Specifically the designer learned that the level of connectedness between the decomposable T-Panels was vastly over estimated. Additionally, through edge orientation, the design better understands that the type of stiffener, S_i is relatively unimportant in determining a most fit design. Instead the variables P_t and N_i must first be resolved and the stiffener type is then conditionally dependent upon this choice.

In the second case study, the node N_4 controlling the number of stiffeners in the side shell grillage was modified. During selection, the training population is first selected to preserve genetic diversity on the node. This allows the designer to better assess the potential tradeoffs within the space for that specific node. Based on the resulting network structure, results demonstrated that tradeoffs do exist, and the designer can extract this information instantaneously from the conditional probability tables of the network. Also by selecting a network that is genetically diverse, but has reached the feasible space, inserted evidence on the node can be propogated to many other nodes within the network, begininning to reveal higher order interactions within the design.

CHAPTER 6

Conclusion

6.1 Overview

Through a series of developments, the initial potential of the BOA and its associated BNs has been demonstrated, allowing designers to positively affect the quality and understanding of early stage designs. Taking the initial presentation of a binary BOA, the algorithm has been expanded and applied well beyond its original scope. A critical contribution was also made, identifying end exploring the interpretation as a major area for improvement in the design process. Through analysis of the resulting Bayesian network, focus is shifted from a singular design solution set to the design process itself. This leads to better design understanding and decision making, which are critically important in the early stages of design. The result is efficient optimization of early stage marine structures by designers who better understand the design space they are exploring.

6.2 Key Contributions

6.2.1 Efficiency Exploration

The BOA has been presented as a potential solution to the problem of optimizer inefficiency in evaluation of large complex structural designs. This application is one of few existing applications of the optimizer to an engineering problem. The optimization simulations extend the initial formulation of the algorithm to multiple n-dimensional bin sets on each variable, and conclusively demonstrate that the BOA can be used to solve structural design optimizations, reaching high quality solutions also identified by comparable heuristic optimizers.

The problems associated with large complex structural design simulations were clearly identified through comparison with the SOGA, first demonstrating the non-linear behavior and reduced convergence rate of the benchmarking algorithm as the problem grew from single section sizes to multiple section sizes more indicative of complete ship scantling designs. The BOA was hypothesized to be an ideal algorithm for application due to the nearly decomposable format of the design problem. The BOA in its current form does present potential challenges computationally in locating or estimating an ideally representative network. As a result, a secondary set of simulations using hypothetical "good" networks as determined through other explorations into network quality were applied to create a potential bounding limit. The goal is that the efficiency of the BOA may only be improved as higher quality networks and methods are used.

6.2.2 Network Learning Exploration

After establishing a baseline of the performance of the Bayesian optimization algorithm, the network learning processes of the algorithm were explored. Work to better understand and use the legacy Bayesian networks created in each generation of the algorithm shows great potential to shift a design paradigm, from focus on the output/design solutions, to focus on process and design knowledge to be gained. The BOA is unique in that the optimization algorithm uses active learning techniques to learn about the design problems and design variable vector and this knowledge can be stored. This novel approach utilizes the computationally expensive data in a different and novel way than previous optimization approaches which often discard results as they become less fit. The network output gives a simplistic concrete graphical representation of design variable relationships, and these relationships are supported by the mathematical governing equations of the problem statement.

To better understand the network learning process, Bayesian learning algorithms were applied to select data sets on both canonical and the hypothetical design cases. The documented results build a case for potential idea network representations, and the effect data quality and decomposability may have on the resulting network structures. Then, using the known objective functions and constraint functions and their partial derivatives, important variables within the design optimization were identified. Simultaneously, the BOA's networks independently identified these variables, placing them at the root of local structures. This again strengthens the case that the machine learning processes of the algorithm can be used to learn about and identify critical relationships in a previously unexplored or unknown design space.

6.2.3 Designer Application

With the insights gained after studying the network learning process, new approaches were implemented in the use of the BOA as a tool. Specifically, the goal of these case study examples sought to return designer intuition to a complicated design space with ill-defined design interactions. In separate case studies, the designer was able to:

- Verify the notional mental model of the design and its interactions, comparing artificial intelligence to the synthesized models. This comparison allows the designers to make better decisions about critical variables relationships and responses of the design.

- Explore the tradeoffs on specific design variables within the design variable vector and propagate the effects of selection throughout the design space. The designer can nearly instantaneously use the network to affect large portions of the design variable vector based on the decision of a single important node.

Both examples enhance or confirm the designers knowledge and understanding of the design problem, while simultaneously producing optimal results to use as the designer normally would with any other optimization approach. In the second example, this goes a step farther, allowing the designer to specifically pick a design variable and influence the entire variable vector based on this inserted "evidence". The lasting impacts of this cannot be understated as naval designs continue to face increasing uncertainty and variability in early stages.

6.3 Future Work

A wide range of potential work areas exist at the completion of this thesis. As in all engineering optimization explorations, the case and scope of the optimization problems can be expanded in any number of directions. To date, the original publications of the BOA focused on its potential use on decomposable problems. This work has extended exploration to what have been described as Nearly Decomposable problems. Work exists to test the limits of a problems decomposability and the performance of the optimizer. Beyond these case study expansions, two critical questions remain unanswered. First, can a properly constructed network lead to more efficient optimization simulations at large problem sizes? And secondly, what network learning processes can be improved to make the BOA results more accurate.

6.3.1 Advancement of the Optimizer Efficiency Exploration

At the time of completion, the question that has driven large portions of this thesis still remains, can the BOA actually out perform the SOGA, a relative gold standard in heuristic optimization, as problem sizes increase. Work done has outline all the potential pitfalls and snares that exist while trying to answer this question, and results do indeed point to the potential that a properly trained network can more efficiently solve this problem.

6.3.2 Improvements to the Network Learning process

As with other heuristic approaches various operators may be used to improve the performance of the algorithm in the various stages of the optimization. The GA has gone through decades worth of research into various tweaks and their incremental improvement. From the results of this thesis it may be concluded that the goal of the network learning process is to first capture a training population set that is both statistically diverse and yet representative of the objective function space behavior close to optimal solutions. As such many potential selection approaches beyond the simple 2 pass tournament currently applied may be used to cull this training pool from the general simulation population.

Additionally, as described in the previous literature review, identifying optimal network structures is a very difficult task. The current algorithm employs the most basic and robust solution to determining network structure, and many existing techniques may be applied to find better fit networks.

6.3.3 Applying network learning to other algorithms

The BOA also holds excellent crossover potential to combine with other algorithms. Any generationally advanced population based heuristic algorithm effectively creates a training data set. Bayesian networks can then be created from these training sets to capture design knowledge that might be lost. In this way a blend of the best qualities of each approach is achieved. Efficient optimization algorithms remain efficient, while documentation of the deign problem occurs in the network.

Then more network analysis techniques can be applied to help designer intuition and understanding.

6.4 Summary

The BOA has been presented as a novel and useful tool for early stage design. Its used of Bayesian networks make it ideally suited to solve nearly decomposable design problems. This matches well with rule based approaches to early stage marine structural design approaches. The evaluation of failure modes allows the conditional dependencies of the network to better resolve the design problem. This then presents the BOA as one potential solution to the increasing demands of computationally expensive structural design fitness.

The BOA also presents clear advantages with its learning processes, shifting the focus from a single design solution set, back to the design process. Through analysis of the resulting networks and conditional probability tables, a designer can capture design space knowledge and apply it. The BOA network identify the design relationships in a simple and clearly understandable graphical format which adds to designer knowledge. This tool was then used to positively effect design solutions in two case examples, first showing how designer intuition may be challenged and improved and then demonstrating how the network and conditional probability tables can be used for design variable exploration.

The BOA is indeed a useful tool for early stage design. With continued expansion its impacts will be realized allowing designers to ultimately produce better designs in an efficient manner.

BIBLIOGRAPHY

(2010). Marines: From Procurement Tragedy to Triumph.

Andrews, D. J. (2012). Art and science in the design of physically large and complex systems. *Proc. R. Soc. A*, 468(2139):891–912.

Chickering, D. M. (1996). Learning Bayesian Networks is NP-Complete. In Fisher, D. and Lenz, H.-J., editors, *Learning from Data*, number 112 in Lecture Notes in Statistics, pages 121–130. Springer New York.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347.

Crucchiola, J. (2015). The New $3B USS Zumwalt Is a Stealthy Oddity That May Already Be a Relic.

Daly, R., Shen, Q., and Aitken, S. (2011). Learning Bayesian networks: approaches and issues. *The Knowledge Engineering Review*, 26(02):99–157.

David, E. and Jon, K. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA.

Deb, K. (2004). *Optimization for Engineering Design: Algorithms and Examples*. PHI Learning Pvt. Ltd.

Deb, K., Bandaru, S., Greiner, D., Gaspar-Cunha, A., and Tutum, C. C. (2014). An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15:42–56.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Devine, T. and Collette, M. (2013). Application of the Bayesian optimization algorithm to naval structural design. In *Proceedings of the 12th Triennial Conference on Practicals Design of Ships and Other Floating Structures*, pages 694–701, Changwon, Korea.

Devine, T. and Collette, M. (2014). Use of Network Metrics with the Bayesian Optimization Algorithm in Marine Structural Design. In *Proceedings of the 13th*

International Conference on Computer Applications and Information Technology, pages 365–377, Reworth, UK.

Diez, M. and Peri, D. (2010). Robust optimization for ship conceptual design. *Ocean Engineering*, 37(1112):966–977.

Dym, C. L. (1994). *Engineering Design: A Synthesis of Views*. Cambridge University Press.

Ehlers, S. (2012). A Particle Swarm Algorithm-Based Optimization for High-Strength Steel Structures. *Journal of Ship Production and Design*, 28(1):1–9.

Fieldsend, J. (2013). Multi-modal optimisation using a localised surrogates assisted evolutionary algorithm. In *2013 13th UK Workshop on Computational Intelligence (UKCI)*, pages 88–95.

Ghose, D. J., Nappi, N. S., and Wiernicki, C. J. (1994). Residual Strength of Damaged Marine Structures. Technical report.

Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann.

Hart, C. G. and Vlahopoulos, N. (2010). An integrated multidisciplinary particle swarm optimization approach to conceptual ship design. *Structural and Multidisciplinary Optimization*, 41(3):481–494.

Heckerman, D. (1998). A Tutorial on Learning with Bayesian Networks. In Jordan, M. I., editor, *Learning in Graphical Models*, number 89 in NATO ASI Series, pages 301–354. Springer Netherlands.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Herrington, P. and Latorre, R. (1998). Development of an aluminum hull panel for high-speed craft. *Marine Structures*, 11(1-2):47–71.

Hmelo, C. E., Holton, D. L., and Kolodner, J. L. (2000). Designing to Learn About Complex Systems. *Journal of the Learning Sciences*, 9(3):247–298.

Hughes, O. F., Mistree, F., and Zanic, V. (1980). PRACTICAL METHOD FOR THE RATIONAL DESIGN OF SHIP STRUCTURES. *Journal of Ship Research*, 24(2):101–113.

Hrnlein, H. R. E. M. (1987). Take-Off in Optimum Structural Design. In Soares, C. A. M., editor, *Computer Aided Optimal Design: Structural and Mechanical Systems*, number 27 in NATO ASI Series, pages 901–919. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-83051-8_26.

Jang, C. D., Seo, S. I., and Kim, S. K. (1996). A study on the optimum structural design of surface effect ships. *Marine Structures*, 9(5):519–544.

Kamat, M. P. (1993). *Progress In Astronautics and Aeronautics: Structural Optimization: Status and Promise*. AIAA.

Kim, J. and Bernitsas, M. M. (1988). Redesign of marine structures by perturbation. *Marine Structures*, 1(2):139–183.

Klanac, A. and Jelovica, J. (2009). Vectorization and constraint grouping to enhance optimization of marine structures. *Marine Structures*, 22(2):225–245.

Larraaga, P. and Lozano, J. A. (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Springer Science & Business Media.

Lee, K. Y. and El-Sharkawi, M. A. (2008). *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. John Wiley & Sons.

Lewis, K. (2012). Making Sense of Elegant Complexity in Design. *Journal of Mechanical Design*, 134(12):120801–120801.

Lima, C. F., Lobo, F. G., Pelikan, M., and Goldberg, D. E. (2011). Model accuracy in the Bayesian optimization algorithm. *Soft Computing*, 15(7):1351–1371.

Moore, A. and Wong, W.-k. (2003). Optimal reinsertion: A new search operator for accelerated and more accurate Bayesian network structure learning. In *In Proceedings of the 20th International Conference on Machine Learning (ICML 03*, pages 552–559. AAAI Press.

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

Nielsen, T. D. and JENSEN, F. V. (2009). *Bayesian Networks and Decision Graphs*. Springer Science & Business Media.

of Shipping, A. B. (2014). Part 3: Hull Construction and Equipment. In *Rules For Building and Classing High Speed Naval Craft*, pages 29–105. American Bureau of Shipping, 2014 edition.

O'Rourke, R. (2009). Navy DDG-51 and DDG-1000 Destroyer Programs: Background and Issues for Congress. Technical report.

Paik, J. K., van der Veen, S., Duran, A., and Collette, M. (2005). Ultimate compressive strength design methods of aluminum welded stiffened panel structures for aerospace, marine and land-based applications: A benchmark study. *Thin-Walled Structures*, 43(10):1550–1566.

Pelikan, M., Sastry, K., and Goldberg, D. E. (2002). Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258.

Peri, D. and Campana, E. F. (2003). Multidisciplinary Design Optimization of a Naval Surface Combatant. *Journal of Ship Research*, 47(1):1–12.

Peri D, C. E. F. (2003). Multidisciplinary Design Optimization of a Naval Surface Combatant. *Journal of Ship Research*, 47(1):1–12.

Pinto, A., Peri, D., and Campana, E. F. (2007). Multiobjective Optimization of a Containership Using Deterministic Particle Swarm Optimization. *Journal of Ship Research*, 51(3):217–228.

Rahman, M. K. and Caldwell, J. B. (1992). Rule-based optimization of midship structures. *Marine Structures*, 5(6):467–490.

Reilly, U. S. N. p. b. J. n. C. P. (2003). 030313-N-0115R-077 The Mediterranean Sea (Mar. 13, 2003) – The guided missile destroyer USS Arleigh Burke (DDG 51) steams through the Mediterranean Sea. Arleigh Burke is currently deployed in the Mediterranean Sea conducting missions in support of Operation Enduring Freedom. U.S. Navy photo by Journalist 2nd Class Patrick Reilly. (RELEASED).

Rigo, P. (2001). Least-Cost Structural Optimization Oriented Preliminary Design. *Journal of Ship Production*, 17(4):202–215.

Rigterink, D., Collette, M., and Singer, D. J. (2013). A method for comparing panel complexity to traditional material and production cost estimating techniques. *Ocean Engineering*, 70:61–71.

Rittel, H. W. J. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169.

Sekulski, Z. (2010). Multi-objective topology and size optimization of high-speed vehicle-passenger catamaran structure by genetic algorithm. *Marine Structures*, 23(4):405–433.

Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F. (2001). Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization. *AIAA Journal*, 39(12):2233–2241.

Simpson, T. W., Rosen, D., Allen, J. K., and Mistree, F. (1998). Metrics for Assessing Design Freedom and Information Certainty in the Early Stages of Design. *Journal of Mechanical Design*, 120(4):628–635.

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

Singh, M. and Valtorta, M. (1995). Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12(2):111–131.

Snyman, J. (2005). *Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Science & Business Media.

Tsompanakis, Y., Lagaros, N. D., and Papadrakakis, M. (2008). *Structural Design Optimization Considering Uncertainties: Structures & Infrastructures Book , Vol. 1, Series, Series Editor: Dan M. Frangopol*. Taylor & Francis.

Wirsching, P. H. and Chen, Y. N. (1988). Considerations of probability-based fatigue design for marine structures. *Marine Structures*, 1(1):23–45.

Zhu, G., Guo, P., Luo, X., and Feng, J. (2012). The multi-objective optimization of the horizontal-axis marine current turbine based on NSGA-II algorithm. In *26th IAHR Symposium on Hydraulic Machinery and Systems, 19-23 Aug. 2012*, volume 15 of *IOP Conf. Ser., Earth Environ. Sci. (UK)*, page 042039 (8 pp.). IOP Publishing Ltd.