DISSERTATION

OPTIMIZING DESIGNER COGNITION RELATIVE TO GENERATIVE DESIGN METHODS

Submitted by

Michael Botyarov

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2023

Doctoral Committee:

Advisor: Erika Miller

Thomas Bradley
Jeffrey Forrest
Marcia Moraes
Steve Simske
Donald Radford

ABSTRACT

OPTIMIZING DESIGNER COGNITION RELATIVE TO GENERATIVE DESIGN METHODS

Generative design is a powerful tool for design creation, particularly for complex engineering problems where a plethora potential design solutions exist. Generative design systems explore the entire solution envelope and present the designer with multiple design alternatives that satisfy specified requirements. Although generative design systems present design solutions to an engineering problem, these systems lack consideration for the human element of the design system. Human cognition, particularly cognitive workload, can be hindered when presented with unparsed generative design system output, thereby reducing the efficiency of the systems engineering life cycle. Therefore, the objective of this dissertation was to develop a structured approach to produce an optimized parsing of spatially different generative design solutions, derived from generative design systems, such that human cognitive performance during the design process is improved.

Generative design usability foundation work was conducted to further elaborate on gaps found in the literature in the context of the human component of generative design systems. A generative design application was then created for the purpose of evaluating the research objective. A novel generative design solution space parsing method that leverages the Gower distance matrix and partitioning around medoids (PAM) clustering method was developed and implemented in the generative design application to structurally parse the generative design solution space for the study. The application and associated parsing method were then used by 49 study participants to evaluate performance, workload, and experience during a generative design selection process, given manipulation of both the quantity of designs in the generative design solution space and filtering of parsed subsets of design alternatives. Study data suggests that cognitive workload is lowest when 10 to 100 generative design alternatives are presented for evaluation in the subset of the overall design solution space. However, subjective data indicates a caution when limiting the subset of designs presented, since design selection confidence and satisfaction may be decreased the more limited the design alternative selection becomes. Given these subjective considerations, it is recommended that a generative design solution space consists of 50 to 100 design alternatives, with the proposed clustering parsing method that considers all design alternative variables.

keywords: generative design, cognition, systems engineering, clustering

ACKNOWLEDGEMENTS

I would like to thank Dr. Erika Miller for her instrumental support in advising me throughout the doctoral student journey. My desire to conduct research in the human systems engineering domain would not have been fruitful without her exceptional guidance. In addition, I would like to extend a thank you to the entire doctoral committee, Dr. Thomas Bradley, Dr. Marcia Moraes, Dr. Steve Simske, Dr. Donald Radford, and Dr. Jeffrey Forrest. I was first introduced to Dr. Forrest as an undergraduate student at MSU Denver. Since then, Dr. Forrest has not only shared feedback throughout this doctoral process, but also provided invaluable mentorship regarding the world of academia. Most importantly, however, Dr. Forrest has become a true friend whose friendship I cherish.

To my friends and family, I appreciate your continuous love and support. A special thank you to a dear friend, Christi Kazakov, for sharing your technical knowledge, to my wife, Emily Eck, for supporting me through all the highs and lows, to my parents, Aleksey and Olga Botyarov, for always believing in me, and to my furry friend, Pixie, for the meowing encouragement.

DEDICATION

To my grandmother, Nadezhda Botyarov, who instilled in me a passion for learning and striving to be the best version of myself. Your words of encouragement will resonate with me forever, “Будь Умнецей!” I love you and only wish you could see how far we’ve come.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Modern systems engineering processes leverage computer-aided design (CAD) technology to facilitate and enhance the design process. The objective of generative design systems is to utilize designer input to produce novel designs through the use of computing resources and capabilities. Contrary to many conventional design processes, generative design is user-driven, meaning that the designer is part of the system of systems and engages with the design interface to achieve the end goal of selecting a design alternative. Given that the designer is a crucial stakeholder, human cognition plays a crucial role in the context of generative design systems. Since generative design systems produce a plethora of design alternatives given the infancy of design maturity during the conceptual design phase of the systems engineering life cycle, designers have the potential to experience cognitive overload without a structured method of parsing generative design system output to facilitate the selection of a single design alternative. Cognitive overload can manifest in the form of decision fatigue, including paradox of choice, resulting in impaired judgment. These unfavorable consequences yield a cumbersome systems engineering process and could result in insufficient design selection. Therefore, it is imperative that generative design systems leverage parsing methods that methodologically reduce the quantity of design options that are presented to the designer, while retaining novel designs from distinct design solution space regions. Although parsing a solution space can yield smaller subsets of design alternatives, it is also imperative to consider how the subsets are presented to the designer.

This research objective of this dissertation is to determine a structured approach to produce an optimized parsing of spatially different generative design solutions, derived from generative design systems, such that human cognitive performance during the design process is improved. To provide the reader with a structured and comprehensive understanding of the research and associated contributions, the dissertation document has been divided into seven chapters. Chapter 1, which is the introduction, provides the reader with an overview of the dissertation document along with a concise overview of the research. Chapter 2 is the literature review that begins with a discussion of the systems engineering life cycle and an overview of generative design, including existing generative design methods and generative design systems. The literature review then continues with a discussion of human cognition and its importance in systems engineering. At the conclusion of Chapter 2, human cognition in the context of generative design systems is reviewed along with the identification of gaps in existing research, specifically gaps this research aims to contribute to regarding human cognition during generative design processes. Prior to addressing the methodology and subsequent research results, Chapter 3 presents generative design usability foundation work in the form of a case study that was conducted. The objective of this foundation work and associated case study was to elaborate on a gap found in literature in the context of evaluating human cognition during the generative design process, specifically the usability of generative design system output. Since the generative design process is iterative and the designer spends significant cognitive workload evaluating generative design system output, this case study focused on this aspect of generative design systems and gathered input from human subjects regarding subjective impressions of generative design system output. Specifically, analysis of case study data indicated that although generative design systems have useful applications within systems engineering applications, there exists a need for a structured method for parsing generative design system output. This foundation work further validated the need to address generative design systems from the human systems engineering perspective. Chapter 4 presents the methodology and study development for this research, including an overview of a generative design application that was created for the purpose of evaluating the dissertation research objective. Chapter 5 presents a clustering methodology discussion, which addresses a novel generative design solution space parsing method that was developed as a part of this research. The method developed leverages the Gower distance matrix and partitioning around medoids (PAM) clustering method to structurally parse a generative design solution space, resulting in a refined decomposition of a generative design solution space to a level of fidelity of the designer's preference. Chapter 6 applies this parsing method and presents the generative design study results and analysis. The performed generative design study incorporated the generative design application and associated methodology to evaluate designer performance, workload, and experience during a generative design selection process, given manipulation of both the quantity and filtering of parsed subsets of design alternatives. Lastly, Chapter 7 summarizes the contributions of this research along with providing suggestions for future work in this domain.

Results of this research have been published in three journal articles highlighted in the list below.

- Botyarov, M., & Miller, E. (under review). Generative design solution space parsing: An evaluation of user experience, workload, and performance. *Journal of Systems Science and Systems Engineering*.
- Botyarov, M., & Miller, E.E. (2022). Partitioning around medoids as a systematic approach to generative design solution space reduction. *Results in Engineering*, 100544. doi.org/10.1016/j.rineng.2022.100544

- Botyarov, M., & Miller, E. (2021). Evaluating usability of generative design process for human-centered design. *International Journal of Development Research, 11*(3), pp 45148-45152.

CHAPTER 2: LITERATURE REVIEW

Systems engineering is a holistic discipline with a broad scope of applications spanning many domains. Systems engineering can be applied to any system development, ranging from a household appliance to an aircraft (Ryen, 2008). Given this dynamic nature, there is not a singular universal definition of systems engineering or a widely accepted approach to conducting systems engineering (K. M. G. Adams & Keating, 2011). However, systems engineering can be described as a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific technological and management methods (Walden et al., 2015). In the context of systems engineering, a system is an assemblage or combination of functionally related elements or parts forming a unitary whole (Blanchard & Fabrycky, 2014). In other words, a system is a collection of components, that when integrated together, perform a function that is greater than simply the sum of the components. Prior to discussing generative design, it is imperative that the systems engineering life cycle is reviewed to provide a foundation for understanding generative design systems and their relationships with the systems engineering life cycle.

2.1 Systems Engineering Life Cycle

The systems engineering life cycle encompasses elements the product life cycle, in addition to several concurrent phases that enable an efficient systems development process (Forsberg & Mooz, 1992). The product life cycle originates with a need and consists of several phases including conceptual design, preliminary design, detail design and development, production, operations, and disposal. The systems engineering life cycle goes beyond the product life cycle viewed in isolation given the overall concurrent engineering approach. Concurrent engineering aims at specifying optimal system configurations by leveraging an interdisciplinary team of engineers to consider system configurations from the perspective of the overall systems engineering life cycle (Schumann et al., 2010). This approach to systems engineering is prevalent in multiple domains, including aerospace, where computer-aided tools are commonplace to facilitate simultaneous activities of the systems engineering life cycle (Knoll et al., 2018). The concurrent engineering approach to the systems engineering life cycle results in four concurrent life cycles progressing in parallel, illustrated by Figure 1 (Blanchard & Fabrycky, 2014). Given the scope of generative design systems, discussion of the systems engineering life cycle is limited to the conceptual design, the preliminary design, and the detail design and development phases.

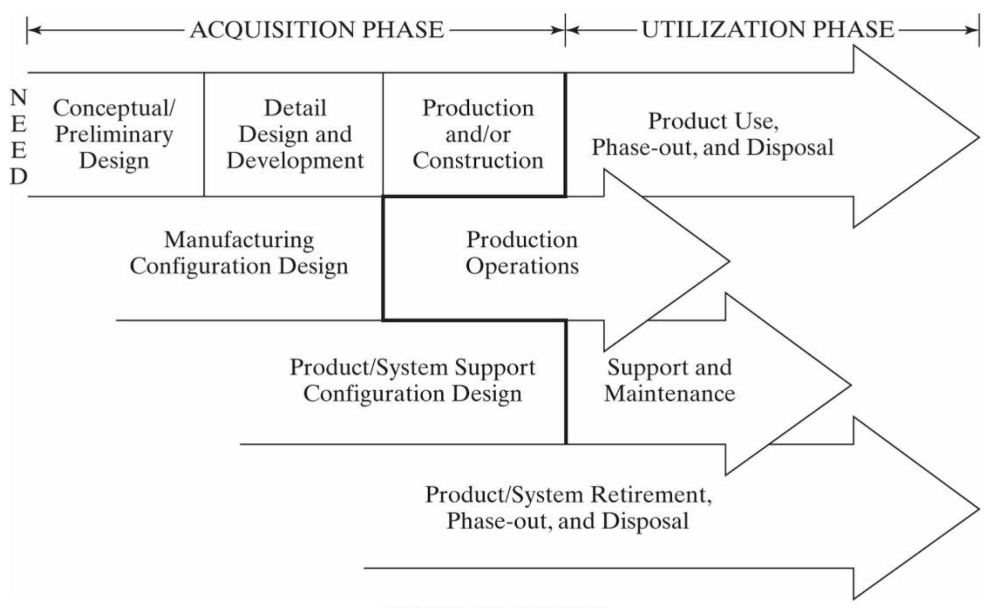


Figure 1: Systems Engineering Life Cycle

2.1.1 Conceptual Design

Conceptual design is the first phase of the systems engineering life cycle and is often viewed as the most important phase because the main outputs are system requirements along with a functional baseline of the system (Blanchard & Fabrycky, 2014). Conceptual design begins with an identification of a problem and an associated definition of need that provides a valid starting point for system design that can then be mapped to engineering characteristics (Eres et al., 2014). It is during conceptual design that the need is translated into system level requirements, specifically operational requirements and support requirements. Operational requirements and an associated operations concept define the system in terms of its projected mission, performance, operational deployment, life cycle, utilization, effectiveness factors, and the anticipated environment (Blanchard & Fabrycky, 2014). This leads to the creation of technical performance measures (TPMs) and associated design criteria. Support requirements along with an associated maintenance and support concept evolve from the definition of operational requirements. Support requirements encompass the maintenance and support infrastructure and define specific design-to requirements for various elements of support (Zhang & Chu, 2010). Approaching system design from this perspective facilitates the design process such that supplementary functions of the systems engineering life cycle and their impacts on system design are considered early (Andreasen et al., 2015). This ensures that form follows function, meaning system functionality is considered prior to assigning specific components to execute those functions.

2.1.2 Preliminary Design

Preliminary design is the second phase of the systems engineering life cycle which further refines the preferred system concept developed during the conceptual design phase and allocates defined system level requirements for subsystems (Blanchard & Fabrycky, 2014). The primary goal of the preliminary design phase is to explore the feasible design space via the consideration of defined system requirements and any identified constraints (Lapins, 1997).

During preliminary design, the functional baseline becomes an allocated baseline as system requirements are decomposed to the subsystem level. It is at this period in the systems engineering life cycle that computer-aided design (CAD) tools are leveraged to document design characteristics and conduct preliminary modeling of the interactions between proposed subsystems. CAD tools are also used to assist in the decision-making process given that the preliminary design phase involves the parsing of multiple design alternatives to conduct trade-offs and similar activities (Sebastian & Ledoux, 2009). The decisions made during the evaluation of design alternatives correspond to cost implications given that 50% to 75% of the projected life-cycle cost for a given system can be committed based on engineering design and management decisions made during the conceptual design phase and preliminary design phase (Blanchard & Fabrycky, 2014).

2.1.3 Detail Design and Development

Detail design is the third phase of the systems engineering life cycle during which system and subsystem level requirements are further decomposed and allocated to the lowest level of the system hierarchy (Blanchard & Fabrycky, 2014). During this phase, system components are identified along with units, assemblies, lower-level components, software modules, people, facilities, and elements of maintenance and support (Blanchard & Fabrycky, 2014). The output of the detail design and development phase is the final configuration of the system that can then be produced for operational use (Pidaparti, 2023). The final system configuration is verified and validated via the application of computer-aided engineering (CAE) and computer-aided design (CAD) tools that can assist with validation of system requirements can be accomplished through the use of simulation methods (Blanchard & Fabrycky, 2014). System modeling tools, such as a prototype that uses all the proposed final configuration elements, undergoes testing to replicate the intended operational scenario(s) to validate changes by modeling global system consequences of local changes (Sheard & Mostashari, 2009). Change requests are common during the detail design and development phase, therefore it is imperative that a formal change control process is followed to track all change requests. If changes are not vetted and documented, then system complexity will increase, causing potentially undesirable emergent behavior both at the system level and when a system is integrated into a larger system-of-systems (SoS) (Osmundson et al., 2008). SoS are particularly prevalent across multiple domains as they are systems that are composed of independent constituent systems acting jointly towards a common goal through the synergism between them (Nielsen et al., 2015).

2.1.4 Systems Engineering Life Cycle Outlook

As modern systems evolve and become exponentially complex, it is increasingly important to leverage independent systems to assist designers throughout the activities associated with the systems engineering process. Specifically, the United States Department of Defense (DoD) has begun to recognize the need to orchestrate the development and management of ensembles of systems to address use capability needs (J. S. Dahmann & Baldwin, 2008). As DoD systems and SoS become increasingly recognized with explicit management, systems engineering, and funding support, it is imperative that systems are created and optimized to assist in applying systems engineering processes. This will ultimately result in the standardization of tools and processes that can be applied across domains for systems and associated SoS (J. Dahmann & Roedler, 2016).

2.2 Generative Design Overview

Generative design can refer to any design process in which the designer uses a generative design system to solve a design problem with a certain degree of automation (Di Filippo et al., 2021). During the 1980's Computer-Aided Design (CAD) systems were prevalent as common tools assisting human designers with design representation and manipulation. Although these traditional CAD systems brought efficiency to the design process, they did not support the human design decision (Mountstephens & Teo, 2020). The concept of generative design originated in the early 1970's and has since been enhanced by additive manufacturing, artificial intelligence algorithms, and unlimited cloud computing power (McKnight, 2017). Generative design was first utilized in the architectural domain with primary applications in open-problem scenarios characterized by large design spaces (Buonamici et al., 2020). However, with advances in computing, generative design has seen exponential growth in the design community. With an expansion of use for design applications, generative design continues to provide an innovative perspective to the design process by identifying unique geometric solutions that would otherwise not be considered under traditional design approaches (Buonamici et al., 2020). Therefore, generative design is used to encourage human divergent thinking and creativity, specifically to applications where aesthetics and innovativeness are important, as in the product development process (Buonamici et al., 2020). Generative design yields the greatest benefits between the conceptual design phase and detail design and development phase of the systems engineering life cycle. Design alternatives are produced via algorithms, subsequently offering the potential for the exploration of expansive design spaces, the nurturing of human creativity, the combination of objective and subjective requirements, and the integration of the design process (Mountstephens & Teo, 2020).

It is important to differentiate between the concept of topology optimization and generative design in the context of research since there are similarities between the two terms. Topology optimization is method of design exploration, primarily used during late stages of the design process, specifically the detailed design and development phase of systems engineering life cycle (Krish, 2011). Since the majority of design elements are already established by this phase of the systems engineering life cycle, topology optimization focuses on operations within narrow bounds to improve specific performance, such as reducing mass of existing design (Krish, 2011). Therefore, the objective of topology optimization is to find the optimal material distribution of a structure with respect to its design and boundary constraints (Tyflopoulos et al., 2018). Considering the distribution of material, topology optimization methods aim to provide a design parameterization that leads to a physically optimized design with respect to its applied loads, boundary conditions, and other constraints (Sigmund & Petersson, 1998). On the other hand, generative design operates primarily during the conceptual design phase, where the design is still under formulation (Krish, 2011). With increasingly integrated systems composed of multiple subsystems and myriad components, the complexity of engineering problems has subsequently increased. By leveraging generative design in the early phases of the design process, generative design facilitates the design process by providing novel solutions to complex problems that designers may have otherwise been inefficient in solving or even unable to solve (McKnight, 2017). Therefore, topology optimization is leveraged to refine a design alternative from the perspective of material distribution, given existing design constraints, whereas generative design is leveraged to create multiple design alternative given broader constraints than topology optimization, such as material type(s), manufacturing method(s), and other similar parameters.

Traditional CAD-based design processes require experts in the subject area and a meticulous understanding of design principles and the design problem (McKnight, 2017). Therefore, it is evident that generative design will continue to play an increasingly crucial role early in the design process by producing novel solutions to complex problems in an efficient manner, thereby reduced development time and associated costs. The ability to explore a large design space via generative design processes is one of the main objectives besides achieving efficiency (e.g., multiple design instances in limited time) and cost reduction (e.g., reduced time and labor) (Singh & Gu, 2012). Additional benefits of leveraging generative design, when compared to the traditional design process, include (1) improved design alternative trade-offs, (2) increased designer creativity, and (3) customized product development (McKnight, 2017).

2.2.1 Generative Design Process Theory

The primary objective of the conceptual design phase is to outline the function of the system at a high-level via functional analysis (Blanchard & Fabrycky, 2014). This implies that the maturity of the design is fluid and multiple design alternatives are under evaluation by the designer. Generative design is most effective when leveraged during the conceptual design phase of the system life cycle since it provides the designer with multiple design alternatives and facilitates design manipulation. Every generative design process requires (1) a performance metric, (2) configuration variation, and (3) decision-making response (Marsh, 2008). A performance metric is a quantitative or qualitative evaluation of performance that is derived directly from a computer model or calculation (Marsh, 2008). When evaluating performance metrics, it is crucial that an ordinal relationship between the results of multiple analyses exists, meaning there is a benchmark for comparison. In the context of generative design, this could imply evaluating design alternatives against an existing solution or against each other. The configuration variation is an aspect of the design that will be manipulated or changed before each iterative calculation (Marsh, 2008). Common aspects of design that can be manipulated are design constraints within a certain requirements range (e.g., minimum and maximum values). Since generative design problems are complex, there are often multiple constraints that are varied simultaneously. This complexity requires a decision-making response, which is a means of determining which configuration parameter should be varied and by how much in response to each iterative analysis result (Marsh, 2008). Therefore, it is crucial to understand the relationships between design constraints (e.g., independent or dependent) to adequately balance them as designs are generated and evaluated within the framework of stipulated system requirements. When evaluating designs, the performance metric must be referred to judge if the required target has been reached or to compute the magnitude and direction of subsequent variations (Marsh, 2008). As a result, independent of the generative design method used, the generative design process consists of (1) a design schema, (2) a means of creating variations (e.g., rule or algorithm), and (3) a means of selecting desirable outcomes (Krish, 2011). Regardless of continuous development of generative design methods and systems, the designer and their experience will always remain an essential element of the design process (Buonamici et al., 2020). Figure 2 below provides a visual representation of the generative design process.

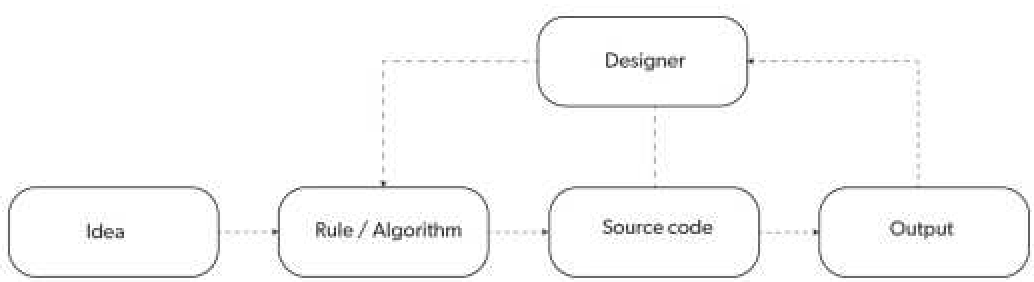


Figure 2: Generative Design Process (Agkathidis, 2015)

2.2.2 Generative Design Methods

Generative design process theory is executed by implementing generative design methods. The five most common generative design methods include (1) cellular automata, (2) genetic algorithms, (3) shape grammars, (4) L-systems, and (5) swarm intelligence and multi-agent societies (Singh & Gu, 2012). Since the design does not reach high levels of fidelity during the conceptual design phase, it is plausible to evaluate the applicability of each generative design method during each iteration of the generative design process. Generative system applications could utilize one or more of these generative design methods depending on the nature of the problem and the expertise of the designer (BuHamdan et al., 2021). This would allow other generative design methods to be implemented when they present significant technical, design, and system development benefits when compared to a singular method being used.

There have been multiple generative design methods developed to execute the fundamentals of generative design methodology. The following sections provide an overview of generative design methods leveraged across generative design systems. In addition to the methods below, ongoing research aims at exploring neural networks and similar artificial intelligence technology to further enhance generative design systems (Singh & Gu, 2012).

2.2.2.1 Cellular Automata

Cellular automata involves a collection of cells on a grid of a specified shape that evolve over time according to a set of rules driven by the state of the neighboring cells (Singh & Gu, 2012). Design constraints are implemented from bottom-up governing the local behaviors of each cell. Therefore, the outcomes of cellular automata are often complex and difficult to predict.

The grid-based nature of cellular automata has applications for architectural design, but the method is unsuitable for product design and has not been used (Mountstephens & Teo, 2020).

2.2.2.2 Genetic Algorithms

Genetic algorithms are inspired by evolutionary processes as they use the analogues of evolutionary operators on a population of states in a search space to find those states that optimize a fitness function (Gero & Kazakov, 2001). Leveraging this approach, genetic algorithms provide a systematic methodology that mimics the natural selection process to automatically solve problems beginning with a high-level statement of requirements (Langdon et al., 2008). Design problems are solved by employing biologically inspired operators such as mutation, crossover, and selection to generate solutions to optimization and search problems (Gero & Kazakov, 2001). Genetic algorithms have been used in optimization applications, including space layout planning and architecture forms (Singh & Gu, 2012). The following sections present a couple of applications of genetic algorithms in the context of generative design.

2.2.2.2.1 GENE ARCH

Given the options of leveraging generative design for architecture applications, GENE_ARCH is an evolution-based generative design method that leverages adaptation to shape energy-efficient and sustainable architectural solutions (Caldas, 2006). This system applies goal-oriented design for multicriteria optimization, by using a genetic algorithm as the search engine and DOE2.1E building simulation software as the evaluation module (Caldas, 2006). This generative scheme seems to interface directly with CAD systems (Krish, 2011).

2.2.2.2.2 Genetic Algorithm Designer (GADES)

GADES is a structured approach which evolves shapes from random constructs (Bentley, 1999). Under this approach, a phenotype is first specified based on the design space and the genotype is specified based on the solution space (Krish, 2011). A phenotype is a generated default design containing any relevant equations and relationships (Krish, 2011). A genotype is a generic parametric CAD model that consists of a list of design parameters with an initial value exploration envelope (Krish, 2011). From there, a multi-objective evolutionary algorithm is chosen, and the fitness function is defined to evolve the solutions (Krish, 2011).

2.2.2.3 Shape Grammars

A shape grammar is a set of shape rules that can be applied to generate a set or language of designs (Stiny, 1980).The set of shape transformation rules influences the design generation process (Gu & Behbahani, 2021). Transformation rules are applied step-by-step to an initial shape to generate designs (Özkar & Kotsopoulos, 2008). Applying rules generates designs, and the rules themselves describe the generated designs as a design grammar. Due to this, shape grammars are used both generatively and as tools for analysis to formalize existing designs along with discovering underlying patterns (Mountstephens & Teo, 2020). Shape grammars allow users to create a wide range of designs by defining and applying various rules depending on the corresponding phase of the systems engineering life cycle (BuHamdan et al., 2021).

2.2.2.4 L-Systems

L-systems, also known as Lindenmayer systems, were founded on the concept of rewriting in which parts of basic shapes are replaced in an iterative manner to form a complex shape (Prusinkiewicz & Lindenmayer, 1990). L-systems are similar to shape grammars in that both are bottom-up generative design methods (BuHamdan et al., 2021). However, the difference is that L-systems operate on strings that represent a design symbolically rather than spatially (Mountstephens & Teo, 2020). The symbolic representation of the design, instead of directly on the design itself is the primarily differentiator of L-systems from other generative design methods (Parish & Müller, 2001). String rewriting is a substitution system in which rules are used to operate on a string consisting of letters of a certain alphabet (Weisstein, 2021). In the context of generative design, string rewriting is a technique for generating successive iterations (Weisstein, 2021).

2.2.2.5 Swarm Intelligence and Multi-Agent Societies

Multi-agent societies leverage a simulation approach to study the behavior of autonomous agents, individually and collectively, in a given environment (BuHamdan et al., 2021). Within the same context is swarm intelligence which is a property of a system that emerges when a large number of simple agents interact with their environment locally to produce coherent patterns of behavior at higher levels (Mountstephens & Teo, 2020). Emergent behavior in the context of systems engineering refers to unexpected and often complex patterns or phenomena that arise from the interactions of individual components or subsystems within a larger system (Osmundson et al., 2008). Therefore, swarm intelligence is the emergence of patterns on the global level of a system due to the behavior of its interacting agents (BuHamdan et al., 2021). The collective behaviors of unsophisticated agents interacting locally with their environment ultimately causes coherent functional global patterns to emerge (Anderson, 2001). As a result, swarm intelligence systems have the ability to solve design problems without the need for centralized control or a global level model (Mountstephens & Teo, 2020). This allows swarm intelligence to be an effective method for optimizing a wide range of functions (Kennedy

& Eberhart, 1995). Therefore, swarm intelligence presents additional robustness and flexibility when compared to traditional design methods (Blum & Li, 2008).

2.2.3 Generative Design Systems

Generative design systems introduce key concepts that are the foundations of the design process, including geometry, spatial relations and transformations, recursion, reiteration, procedures, and encapsulation (Chase, 2005). As a result, generative design systems change the conventional design practice by adding a new component to the previously binary relation between designers and products, thereby introducing a tertiary relationship that includes a designer, a generative design system, and a product (Fischer & Herr, 2001). The addition of this tertiary relationship brings multidisciplinary expertise that is not available to all designers and aids evaluating designs from a broader perspective (Fischer & Herr, 2001). Although there exists a multitude of generative design systems, they all share several characteristics, specifically they (1) generate complexity via the aggregation of simple forms to create more complex forms, (2) generate forms interconnected with their environment, (3) can self-maintain and self-repair, and (4) can generate novel forms, behaviors, or outcomes (Mccormack et al., 2004). Therefore, generative design systems encompass the processes geometry generation, performance analysis, and evaluation to deliver meaningful designs (BuHamdan et al., 2021).

Modern generative design systems are focused on creating new design processes to produce spatially novel yet efficient, effective, and buildable designs through exploitation of current computing and manufacturing capabilities (Shea et al., 2005). Unlike other design systems and processes, generative design is designer driven, meaning that parametric constraints are defined by the user (Krish, 2011). The parameters that a user inputs, derived from operational requirements, are used to define and subsequently search an abstract design space to explore thousands of design variants (McKnight, 2017). Generative design systems allow a designer to create thousands of design options in less time than the traditional design development lifecycle (McKnight, 2017). It is important to note that generative design is an iterative process, meaning the designer can continuously modify the generative scheme based on the resultant outcomes, until a single solution is selected (Krish, 2011). This type of design process preserves the creativity of the designer by providing novel solutions that occur at various regions in the specified design space. The designer then evaluates competing requirements to make the best set of compromises from a wide range of available design options (Marsh, 2008). The design alternatives that generative design systems produce satisfy a set of imposed design constraints and maximizes a goal function passed to the algorithm, such a technical performance measure threshold or other systems engineering objective (Buonamici et al., 2020).

2.2.3.1 Autodesk Fusion 360

Aside from cases in reverse engineering, systems engineering best practices stipulate that form follows function (Verma & Wood, 2001). One popular CAD based generative design tool is Autodesk Fusion 360, produced by Autodesk, that considers design function and constraints first and then synthesizes this information to produce a physical form. Autodesk has invested significant effort into the integration of generative design systems within the traditional CAD environment (Buonamici et al., 2020). The generative design workflow using Autodesk Fusion 360 consists of (1) opening an existing model or creating a new model workspace to serve as the basis for the generative design study, (2) optionally modifying the generative model (e.g., if existing model, can create bodies to represent preserve, obstacle, and starting shape geometries in a design problem), (3) setting up a design problem and specifying requirements, (4) generating outcomes that satisfy requirements, and (5) exploring outcomes using tools to help identify the optimal outcome. Setting up requirements is an essential step of the Autodesk Fusion 360 generative design process. Aside from editing a base model, if one exists, Autodesk Fusion 360 allows the designer to define design problem parameters including, (1) the design space (e.g., preserve geometry, obstacle geometry, starting shape), (2) design conditions (e.g., structural constraints, structural loads, load case attributes), (3) design criteria (e.g., objectives [minimize mass, maximize stiffness, safety factor], manufacturing [additive, milling, 2-axis cutting, die casting]), and (4) materials (e.g., study materials [metals, plastics, etc.], manage physical materials). To generate solutions, Autodesk Fusion 360 leverages cloud computational services. All generated solutions will inherently satisfy the specified functional, manufacturing, and mechanical requirements. It is important to note that the more variability that is included in the study setup, particularly if the problem is complex and not well known, the wider the set of design solutions will be. Preliminary exploration features within the Autodesk Fusion 360 system allow the designer to view design thumbnails, design properties, and graphically interpret relationships between designs.

2.3 Systems Engineering Design Processes and Human Cognition

Up to this point, discussion has centered around systems engineering processes, including an overview of generative design methods and tools as they pertain to the systems engineering life cycle. However, given that humans are involved throughout the systems engineering life cycle, it is imperative to address the human element in the context of systems engineering and subsequently generative design.

2.3.1 Cognitive Science

Cognition is a term that refers to the mental processes that occur in understanding and thought. Cognitive science emphasizes the importance of cross-disciplinary collaboration by integrating views of thought across multiple domains, including philosophy, psychology, engineering, computer science, and neuroscience (Núñez et al., 2019). Cognitive scientists research adaptive information processing at multiple scales, both spatial and temporal (Allen, 2017). Humans have the ability to learn, remember, sense, perceive, and think, which are all elements of cognition (F. Adams & Aizawa, 2001). Within the last decade, investigation into the role of cognition and cognitive processes in system design is becoming more prominent (Rao & Francis, 2022).

2.3.2 Cognitive Engineering

The growth of cognitively complex systems has led to research on how to improve systems to support human work, leading to research in cognitive engineering (Militello et al., 2010). Cognitive engineering is a multidisciplinary domain that explores the analysis, design, and evaluation of complex systems of humans and technology (Gersh et al., 2005). This provides a structured approach to the design of technology, training, and processes intended to manage cognitive complexity in sociotechnical systems (Militello et al., 2010). In cognitive engineering, cognition is typically assumed to be information processing in a human's mind or brain, specifically a system operator (Blomberg, 2011). Specifically, this domain integrates knowledge and experience from cognitive science, human factors, human-computer interaction design, and systems engineering (Gersh et al., 2005).

Leveraging a cognitive engineering approach, the design of complex systems must involve an ecological stance, and system design must simultaneously consider humans, artifacts, goals, and the environment in which the goals are to be achieved (Gersh et al., 2005). Substantive research exists that emphasizes the importance of applying cognitive engineering methods to support the design of complex, human–machine systems. Such methods encompass several activities, including (1) in-depth data collection through interviews of domain experts and observation of practitioners, (2) system, operator, and task modeling, and (3) development of outputs oriented towards function allocation, design of interfaces to control and information systems, task definitions, and training requirements (Bisantz et al., 2003). Therefore, it is crucial to emphasize observation and developing a cognitive task analysis that captures the tasks and goals of a system user. A cognitive task analysis encompasses systems users performing domain tasks using the concepts and tools of their domain (Gersh et al., 2005). Application of cognitive engineering methods can result in cognitive modeling that captures both the contribution of the domain and the computational characteristics of human cognition that constrain how humans respond to our environment (Gersh et al., 2005). Ultimately, successful implementation of cognitive engineering methods results in the identification of system and task demands that pose complexities for system users and subsequently developing design solutions to mitigate identified complexities (Bisantz et al., 2003).

2.3.3 Generative Design & Designer Cognition

Design is a highly complex task that involves creativity, specialist knowledge, experience, and judgment with regard to both the objective and the aesthetic aspects of a problem domain (Mountstephens & Teo, 2020). In traditional design, the role of the designer is to explore a design solution space given that the designer is direct responsible for the creation of every design in the design solution space. In contrast, generative design methods allow the designer to create and modify a set of rules that interact to generate a design solution space autonomously (Mccormack et al., 2004). Therefore, the designer does not directly manipulate the produced design solution space when compared to traditional design. Instead, the designer primarily interacts with rules and methods of generative design systems that are involved in the production of the design solution space where completed designs are the result of the emergent properties of the system (Mccormack et al., 2004). The creativity of the designer in the context of generative design is in balancing relationships between process specification, environment, and generated designs (Mccormack et al., 2004). Given the creativity of the process, there is no formalized method that can be used to guide these relationships, so the role of the human designer remains, as with conventional design, central to the design process (Mccormack et al., 2004).

Based on the definitions of cognition, it can be implied that generative design is a cognitive system, inclusive of the designer-in-the-loop, since behavior can be iteratively modified based on experience to achieve specific anti-entropic ends (Hollnagel & Woods, 2005). To achieve specific anti-entropic ends means that that the system controls itself to reach a specific goal or to maintain a certain equilibrium state (Blomberg, 2011). In the case of generative design systems, the designer leverages a set of generative design methods to evaluate a design solution space with the specific goal of selecting a design. In doing so, a designer executes pragmatic and epistemic actions. A pragmatic action brings the designer physically closer to a goal state, but in contrast, an epistemic action performed by the designer changes their environment in such a way that they get exposed to new information input (Blomberg, 2011). In other words, an epistemic action might take the designer further from the desired goal, of selecting a design, but the new information might later allow the designer to reach the goal.

Epistemic actions facilitate the thought process and are prevalent in generative design where the designer modifies design criteria or parses the design solution space, as an interim action to evaluate new information, that would then lead to the selection of a design.

Generative design systems attempt to enhance the creativity of the designer by exploring search spaces in an innovative and efficient way to produce novel solutions (Bentley & Corne, 2002). The overall design process focuses on the emergence of shape and form in response to objectives (e.g., requirements) (Oxman, 2002). The processes of reasoning that result in the emergence of form are, by definition, creative processes (Oxman, 2002). During the design process, the designer leverages cognition that encompasses (1) conceptual emergence (e.g., structured knowledge of design concepts, design domain knowledge, etc.), (2) transformational emergence (e.g., externalization of retrieved images and the activation of transformational operations as a class of design knowledge), and (3) anticipated emergence (e.g., knowledge of visual prototypes and ability to transform emerging shapes) (Oxman, 2002). Creativity has often been considered to thrive when minimal structured processes exist, however creative design has been demonstrated to be less a perceptual accident than the activation of a form of design intelligence (Oxman, 2002). This implies that in addition to visual prototypes (e.g., visual stimuli [design semantics] that cue the emergence of generative knowledge and guide perceptual identification), domain knowledge supports innate cognitive abilities to deal with visual images in design, to classify them, and to create generic knowledge (e.g., generic classes of design operations) (Oxman, 2002). This further implies that generative design systems cannot function independent of a designer and that the designer is the main actor that assists with design selection and evaluation. It is the designer’s existing visual memory that helps guide shape manipulation, transformation, and invention during the design process (Oxman, 2002). Human designers cannot be replaced in the generative design process for several reasons, including (1) design options are generated by algorithms that designers create, even though there exists generative design software, (e.g., Autodesk Fusion 360), designers still need to define necessary design parameters and constraints, such as spatial constraints and engineering requirements, (2) designers need to make trade-off decisions among numerous generated design options, often with multiple conflicting features, and (3) the aesthetic requirements cannot be easily encoded into algorithms and largely depend on the subjective perspectives of designers (Li et al., 2021).

The generative design process often produces thousands of design alternatives, which inherently places a significant cognitive workload on the designer (Krish, 2011). Due to the limitations of human cognitive ability, the designer is only able to evaluate a limited number of design solutions without cognitive fatigue (Bentley & Corne, 2002). Therefore, to maximize cognitive performance, a limited number of designs must be presented to the designer (Krish, 2011). Since the designer would only be presented with a subset of the total number of generative design options, it is crucial that the design options be widely dispersed within the viable design space, such that each generated instance may be taken to represent a region of design possibilities (Krish, 2011). Figure 3 below describes how a design solution instance could serve as an accurate representation of the similar design population within the relative region.

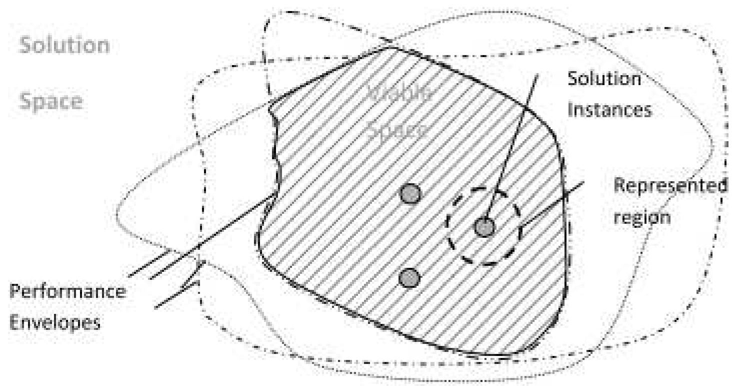


Figure 3: Design Solution Space (Krish, 2011)

As described earlier, at the conclusion of the generative design process the designer selects one design to move forward with to the next phase of the systems engineering life cycle. However, this stage of decision making is often subject to forms of bias (Polman & Vohs, 2016). A common source of designer induced bias is decision fatigue, which describes a phenomenon in which the limited reserve of stamina for making decisions becomes drained, which leads to poor self-control and impaired judgement (Polman & Vohs, 2016). Studies have indicated that decisions that take place over a short amount of time (e.g., minutes or several hours), such as the selection of a design alternative, are also susceptible to decision fatigue (Polman & Vohs, 2016). An outcome of decision fatigue could be that the designer passively selects the default option (Polman & Vohs, 2016). In the context of generative design, this could mean that the designer would select a design alternative that they are most familiar with, which could not be the best option for a given application. Furthermore, this contradicts the principle of generative design to generate novel ideas that explore the entire design space and selecting a default option, which often already exists (e.g., genotype), would defeat the purpose of the generative design process.

In addition to resorting to selecting a default option, decision fatigue also results in an impaired ability to make trade-offs due to pursuing a passive role in the decision-making process, leading to choices that seem impulsive or irrational (Pignatiello et al., 2020). Since generative design is intended for use in the conceptual design phase, it is crucial that the designer has sufficient cognitive ability to evaluate trade-offs that would result in a balanced design which satisfies requirements. Performing trade-offs often hinges on the designer's ability to visualize sub-shapes (e.g., components) within a design that can be modified. Recognizing emergent sub-shapes is a natural capacity of the human designer and is not yet automated by current computational systems (Oxman, 2002). Therefore, it is imperative that the designer does not experience cognitive overload due to the multitude of design alternatives presented. In addition to cognitive overload, the need to determine an optimal quantity of novel generative design options would also help eliminate the complexity associated with the paradox of choice. The traditional definition of the paradox of choice is a phenomenon in which the result of too many choices leaves the human less happy, less satisfied, and occasionally paralyzed (Piasecki & Hanna, 2011). However, an expanded definition of the paradox of choice yields that lack of meaningful choice, rather than an overwhelming amount of choice, that leaves the human less happy, less satisfied, and occasionally paralyzed (Piasecki & Hanna, 2011). To further complicate the decision process, humans themselves are often unable to explicitly define what constitutes a meaningful choice (Piasecki & Hanna, 2011). In the context of generative design, this further implies that output needs to be optimized such that not only a limited subset of design options is presented, but that the design options themselves are visually dissimilar.

2.4 Current Gaps in Research

Modern engineering design systems, including generative design, present challenges to designers given the amount of data that designers must keep track of while performing system analysis and synthesis (Krupa, 2019). Since applications of generative design systems occur during the conceptual design phase of the systems life cycle, the infancy of design further contributes to the generation of a complex design solution space. Generative design systems are intended to facilitate the designer role by generating and exploring design alternatives, but the amount of information generated during a design session can become overwhelming. Therefore, structured navigation methods are needed if designers wish to review the design solutions that generative design systems generate with ease (Chien & Flemming, 2002). A combination of human evaluation and objective performance is required when evaluating design solutions produced by generative design systems (Mountstephens & Teo, 2020). Specifically, important issues to address when considering generative design system evaluation are creativity and design novelty. This is especially crucial because a primary characteristic of generative design systems is the ability to yield novel designs (Mountstephens & Teo, 2020). Given that generative design systems leverage parametric representations, there can simultaneously be too much variation and yet not enough among design alternatives (Johnson, 2016). This suggests that it is imperative to develop a structured approach to parsing a generative design solution space while also considering the human element that is integrated with the generative design process.

During the evaluation of design alternatives there is no single best design alternative. Instead, there are a number of tradeoffs between requirements, requiring humans to evaluate multiple design alternatives options, even if they did not create them manually and instead a generative design system was used (Schulz et al., 2018). In addition to objective requirements, there are also subjective requirements, namely aesthetic requirements of design, that must ultimately be assessed by human judgment, necessitating surveys of user opinion with results differing depending on the chosen audience (Mountstephens & Teo, 2020). To facilitate the evaluation of all facets of design, further research is necessary to test and discover methods to maintain the legibility of a generative design solution space independent of the quantity of design solutions created. Additionally, there is a need to evaluate the design of decision support methods for different generative design tasks with realistic applications (Chien & Flemming, 2002). The development of methods and associated models for the conceptual design phase of the systems engineering life cycle is a field of high practical relevance given the success of subsequent design phases relies on the development of efficient methods during the conceptual design phase (Scheidl & Winkler, 2010). Humans are frequently mentioned as the vital ingredient to the success of the conceptual design phase and concurrent design activities, yet there is little systematic coverage of the cognitive and relational skillset that would result in successful design processes (Knoll et al., 2018). Further research can benefit from assessing the impact generative design systems on the design processes elements of productivity, efficiency, and quality (BuHamdan et al., 2021). Results from such research could then be applied to define best practices to maximize the efficiency of generative design systems in context of generative design system output and human cognition.

Generative design system evaluation in a realistic design context has so far been neglected (Mountstephens & Teo, 2020). Although generative design systems have been developed to create object forms, assessing the quality of these systems is crucial. To assess the quality of generative design systems, at least two questions must be asked, specifically (1) how good is the design in itself, and (2) how effective is the generative design system in the context of a practical application? (Mountstephens & Teo, 2020). Chapter 3 presents generative design usability foundation work that further elaborates on these questions and additional gaps found during the literature review.

CHAPTER 3: GENERATIVE DESIGN USABILITY FOUNDATIONAL WORK

3.1 Introduction

A generative design usability case study was conducted to evaluate human factors considerations for generative design tool outputs. Participants completed a task using a common software program used in generative design and then answered survey questions regarding their experience. This study had approval from the Colorado State University's Institutional Review Board (IRB), protocol 20-10385H.

3.1.1 Case Study Participants

There were 28 participants that completed this study. All of the participants were graduate students from the Systems Engineering Department at Colorado State University, which is a predominately online graduate program. As such, the majority of the participants were full-time systems engineering industry workers in addition to being graduate students. Graduate students in the department were invited to participate in the study, via email, if they were familiar with the concept of generative design and had computer-aided design (CAD) experience.

3.1.2 Case Study Software

Autodesk Fusion 360 was used as the generative design tool for this study. Since all of the participants were students, they were able to download the software for free using the educational license.

3.1.3 Generative Design Space Overview

To ensure validity of the data, all participants were given the same, pre-generated design file to analyze. If participants created their own solution to the problem, then the generative design output would have been different between participants, thereby presenting other factors that would have altered survey responses. As such, participants were instructed to use a demo file that came pre-installed with Autodesk Fusion 360. The demo file used was the ‘Explore_Motorcycle Triple Clamp’ file, which contained 53 unique design solutions for a motorcycle triple clamp, presented in Figure 4 below.

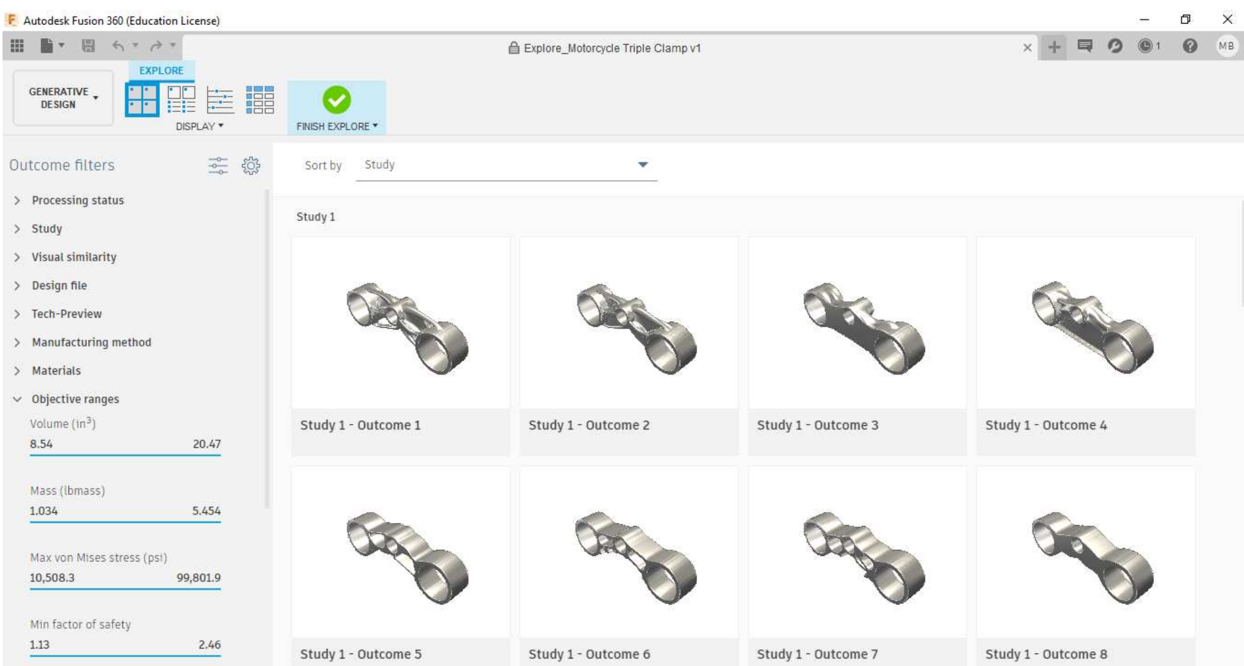


Figure 4: Autodesk Fusion 360 Partial Solution Space for a Motorcycle Triple Clamp

Participants were provided a brief scenario on why they were using this file in the study, as well as the function of a motorcycle triple clamp. They were told they were part of a design team that was designing a new motorcycle, and had input several design constraints into Autodesk Fusion 360, which produced the multiple design alternatives found in the study file. They were then told to follow the task instructions to review the design solutions for use in this new motorcycle.

3.1.4 Case Study Task

Participants completed the experiment on their own computer. The entire task took about 30-45 minutes to complete. Each participant was emailed the same set of instructions and a link to the survey. The instructions were divided into four sections, for a total of 13 steps. The first section instructed them on how to download the educational version of Autodesk Fusion 360. The second section directed them to the Fusion 360 demo file named 'Explore_Motorycle Triple Clamp.' The third section provided instruction on comparing the design outputs, by guiding them to look through all the different designs and utilize various filters. Participants were told within each of the steps what survey questions they would be asked that pertained to each step. The final section of the task instructions directed them to the online survey link.

3.1.5 Case Study Survey

The case study survey was developed and administered using Qualtrics Online Survey platform. The survey included 5-point Likert scale questions regarding participant satisfaction with the quality and quantity of design solutions, their ability to identify differences in design solutions using various filters, and their opinions on limiting the design solution space. Each Likert scale question also had a text input box, for participants to justify their responses. The case study survey also asked participants about their previous experience with generative design.

3.2 Generative Design Usability Case Study Results

Case study results were collected anonymously from participants using Qualtrics Survey Software. Data analysis was conducted in Rstudio (version 1.2.5001). The following section presents the results of this case study and discusses potential impact on generative design output.

3.2.1 Participant Demographics

This case study was conducted with 28 graduate students studying systems engineering at Colorado State University. Since this case study pertained to generative design, it was necessary to determine the experience level every participant had with generative design tools and concepts prior to participating in this case study. Of the 28 students, 26 of them had one year or less of generative design experience, whereas the remainder had between 7 and 12 years of generative design experience. This is an important demographic since those participants with less experience in generative design may not have strong opinions regarding generative design output, however they may have novel ideas based on their unique engineering backgrounds. Likewise, those with significant generative design experience may have strong opinions and be aware of bottlenecks in generative design processes.

3.2.2 Satisfaction with Generative Design Solutions

Analysis of the data indicates that most participants were satisfied with the quality and quantity of generative design solutions. Participants stated that the generative design tool provided a novel solution set with design solutions based on multiple criteria that encompassed a broad range of the overall design envelope. Recall that in this instance, the generative design tool presented 53 unique solutions. It is important to note that actual applications encompass problems with increased complexity, a broader range of parameters (e.g., material options, etc.), and an iterative design cycle, which would produce a larger set of design solutions. This additional complexity could impact perceived levels of satisfaction with the quality and quantity of design solutions. Figure 5 below displays data regarding participants' opinions about the quality and quantity of generative design solutions of the Autodesk Fusion 360 generative design tool presented in this case study.

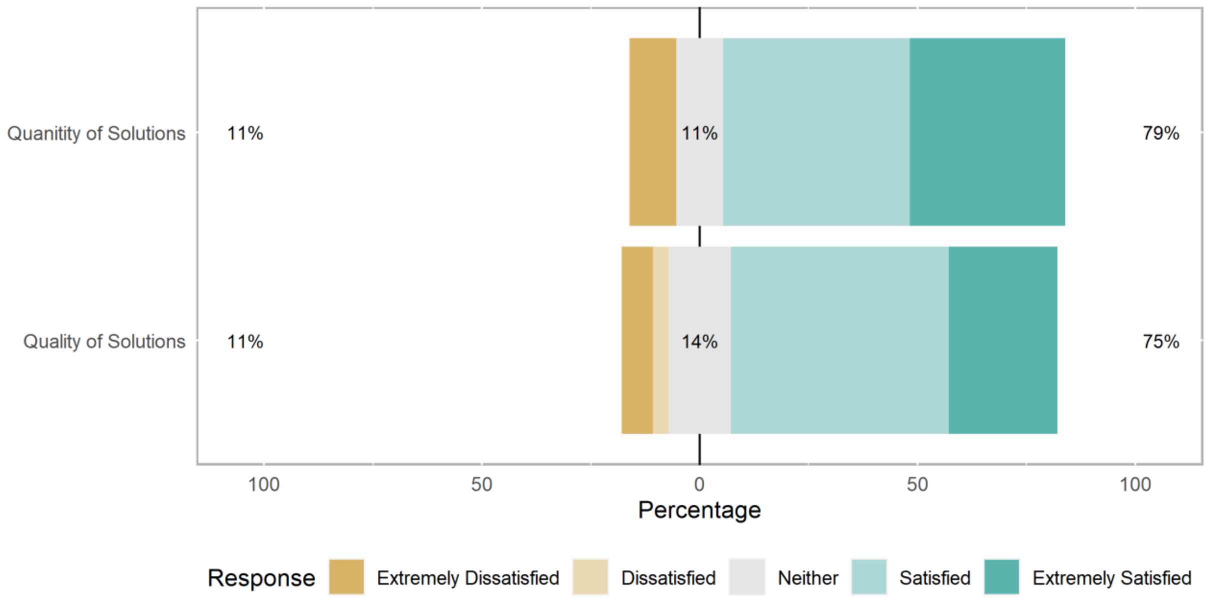


Figure 5: Satisfaction of Solutions

3.2.3 Ability to Identify Differences Between Generative Design Solutions

When the 'study' filter is applied, the generative design tool simply displays all 53 solutions for the study without any grouping or analysis. Therefore, this could be considered as if no filter were applied. The data indicates that when no filter was applied, participants were nearly split when it came to detecting differences between solutions, meaning nearly half believed it was extremely difficult to detect differences between solutions and the remainder believed it was extremely easy to detect differences between solutions. When applied, the visual similarity filter groups similar solutions (e.g., shape) and created 16 subsets of these similar groups from the entire solution set with each group containing between 2-4 solutions. Along with the groups, the filter identified 13 solutions as unique, meaning they were not assigned to a group. With the visual similarity filter applied, the data indicates that participants were able to improve their ability to detect differences between groups of solutions when compared to their ability to detect differences with no filter applied. With the same filter applied, the participants' ability to detect differences between solutions within the same subset (e.g., group) also increased when compared to detecting differences between subsets and detecting differences with no filter applied. This indicates that filtering a design solution based on visual similarity features of a design, such as shape, increases the ability of users to detect differences between the design. Figure 6 below displays data regarding participants' opinions about their ability to differentiate between generative design solutions by using generative design tool filters.

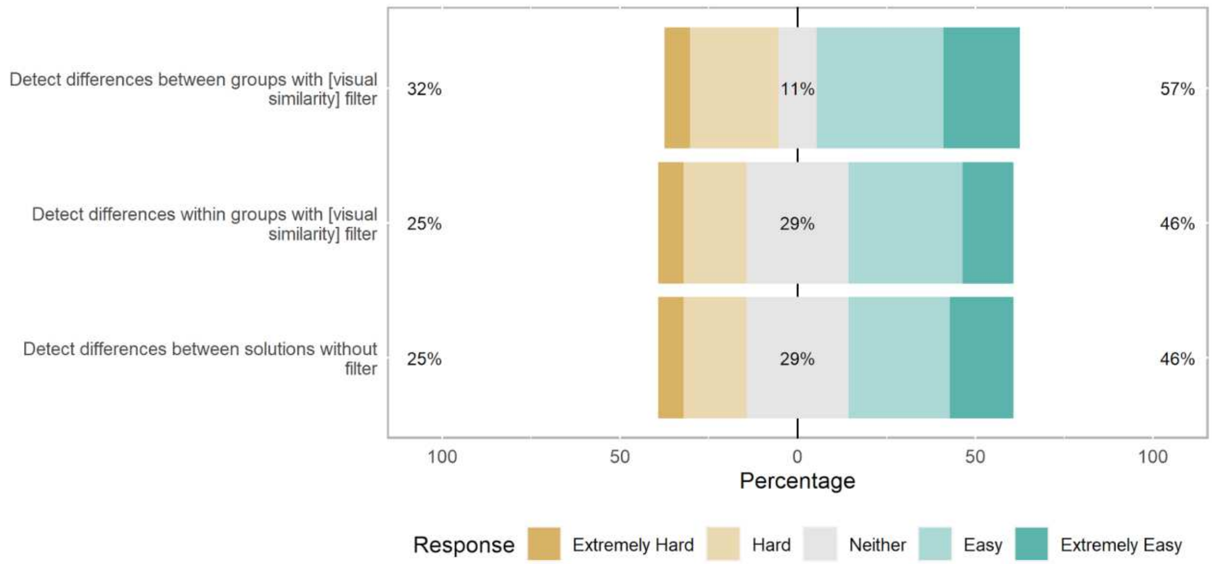


Figure 6: Ability to Identify Differences

3.2.4 Perception of Existing Generative Design System Filters

Most participants indicated that leveraging existing filters would be useful in analyzing a scenario with substantially more generative design solutions. This indicates that initial perceptions of a design, such as shape, play a significant factor in a designer's ability to process design alternatives. Figure 7 below displays data regarding participants' opinions about how useful existing filters (e.g., visual similarity, etc.) would be in identifying unique solutions if substantially more generative design options were generated.

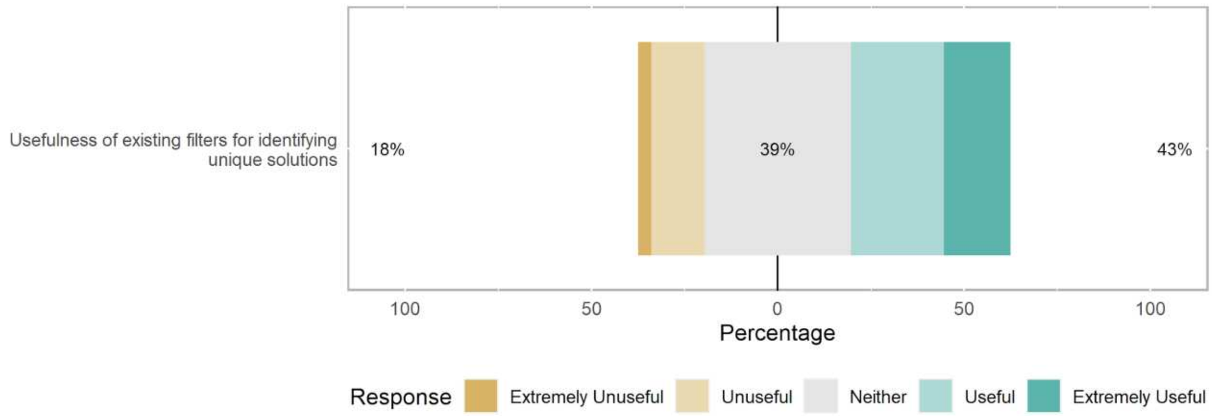


Figure 7: Perception of Existing Filters

3.2.5 Optimal Quantity of Generative Design Solutions

The data indicates that most participants believe that the quantity of generative design solutions a generative design system provides should be limited. This would enable more efficient systems engineering processes since designers would require less cognitive workload to parse a limited subset of an entire design solution space. Furthermore, reviewing less generative design solutions would correlate to reduced costs, particularly if an algorithm was able to assist with the reduction of available design alternatives. Figure 8 below displays data regarding participants' opinions about how useful limiting the quantity of solutions generated by a generative design tool would be, provided that novel solutions remain.

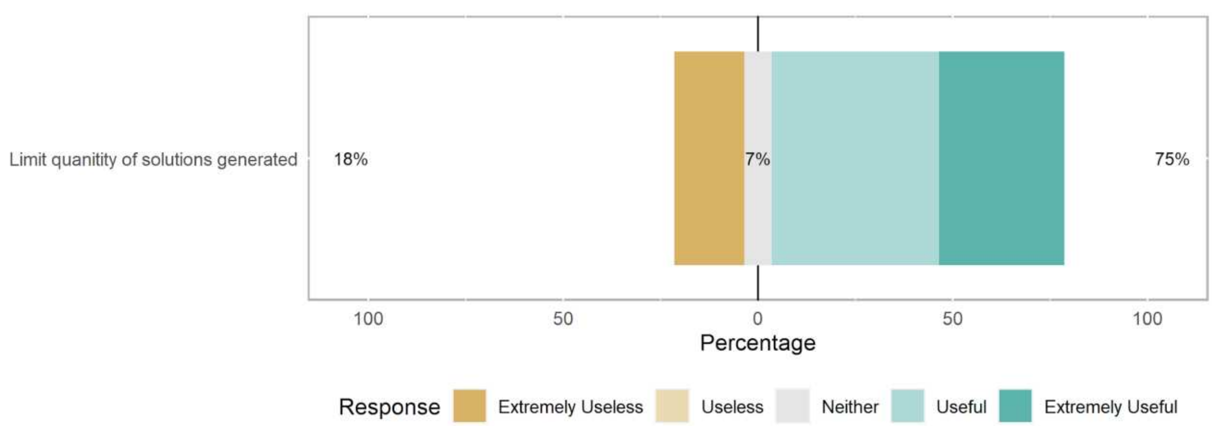


Figure 8: Limiting Quantity of Solutions

The data indicates no strong preference for the quantity of design solutions for the scenario in the case study. The ambiguity of this data could be attributed to the level of experience with generative design tools of the participants. Since most participants were students that had little to no previous experience with generative design tools, they likely did not have a strong opinion. Furthermore, this experiment only presented participants with 53 design options, which is not as many as generative design systems can generate for substantially more complicated applications that are found across various domains, such as aerospace. Therefore, with an already limited quantity of solutions, there may have not been a clear need to further reduce the quantity of the solutions in the experiment. Figure 9 below displays data regarding participants' opinions regarding the optimal quantity of generative design solutions for the scenario in the case study as a percentage of the design solution space provided by the generative design system.

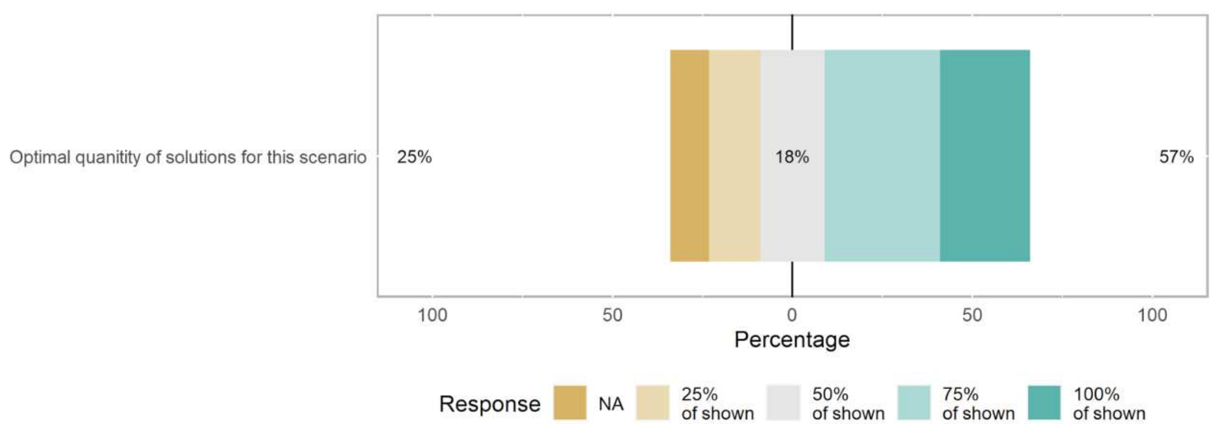


Figure 9: Optimal Quantity of Experiment Generative Design Solutions

The data indicates that most of the participants believe 100 or less generative design solutions is the optimal quantity that should be presented to a designer. Indeed, more than 100 solutions would likely place a high cognitive burden on the designer and negatively impact systems engineering processes. Figure 10 below displays data regarding participants' opinions regarding how many generative design solutions a generative design tool should produce in general, particularly when faced with a significantly more complex problem than the one presented in this case study.

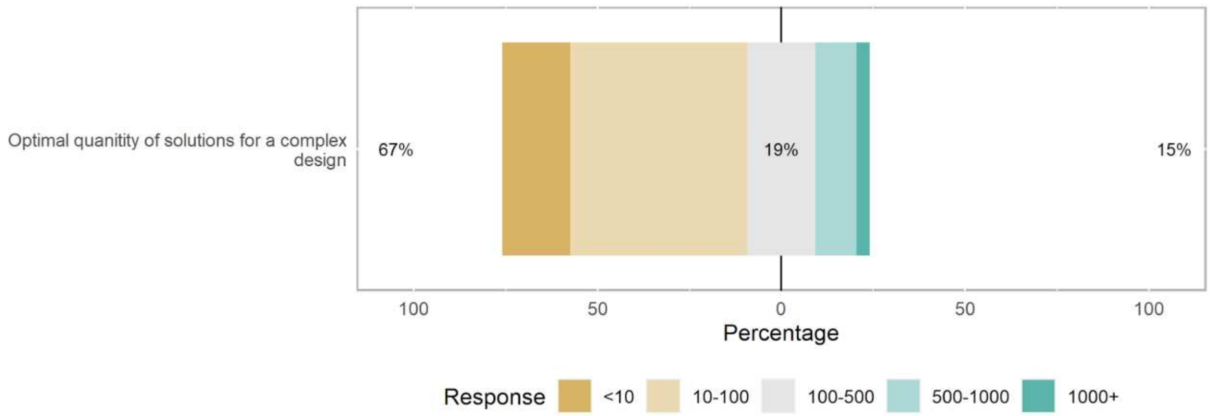


Figure 10: Optimal Quantity of Generative Design Solutions

3.3 Generative Design Usability Case Study Conclusions

Generative design is a powerful tool that could assist designers with producing novel solutions that address complex problems (Krish, 2011). Although generative design systems have myriad benefits they can provide to modern system engineering processes, the systems must be usable from the human perspective to be leveraged successfully. In order for systems to be usable, through the support of human centered design, a combination of the following elements must be incorporated, including (1) drafting and planning for human-centered design processes, (2) understanding the context of use for the system as a basis for identifying requirements and evaluating the system, (3) understanding and specifying user requirements in a clear manner which can be assessed for achievement, (4) developing a system and user interface based on a flexible and iterative approach, and (5) performing an usability evaluation based on expert and user testing throughout system design (Maguire, 2001). Therefore, a successful generative design system would incorporate these elements.

This case study performed a usability evaluation to obtain feedback regarding the elements of generative design tool output by using Autodesk Fusion 360 and a predefined solution space. Participants that completed the experiment agreed that although generative design is a powerful tool, the main area for improvement is developing a method for effectively parsing generative design solutions. The case study indicated that filter types, such as visual similarity, are useful in evaluating a generative design solution set. Filter types could be used to parse through solutions sets with substantially higher volumes of solution instances. Furthermore, case study results indicated that there needs to be a method to limit the overall quantity of generative design solutions presented to the designer. A solution set with an exponentially larger quantity of solutions than this case study (e.g., 1,000 design solutions) would likely impair the designer's ability to select an alternative and limit the benefit of the generative design tool.

CHAPTER 4: METHODOLOGY & STUDY DEVELOPMENT

4.1 Introduction

Evaluation of the literature along with a foundational case study into generative design usability indicated that an optimized parsing of spatially different generative design solutions needed to be determined such that designer cognitive performance was maximized. The following chapter presents the methodology leveraged to (1) quantify if user satisfaction with generative design solution quality and quantity was impacted when more than 50 solutions were generated, (2) quantify if generative design filters decreased designer cognitive workload, (3) determine if certain filter types efficiently processed a large set of generative design solutions, (4) identify a parsing method that reduced the generative design solution space while retaining novel solutions, and (5) determine how human cognitive ability was impacted by various quantity levels of generative design solutions.

4.2 Generative Design Study Solution Space

This study involved human participants with various levels of experience in generative design tools and methodologies. Since the size of a generative design solution envelope varies depending on the complexity of a problem, the experience level of the user, and the inputs the user provides the generative design system, it was crucial to constrain the solution space. This provided a controlled environment in which variables that were hypothesized to influence human cognition, in the context of generative design, were evaluated. Prior to involving human participants, a solution set containing 1,000 solutions for an aircraft engine loading bracket was generated using Autodesk Fusion 360. The solution set generated was then transferred to a static repository, specifically a generative design application, that was developed for the purpose of this study. Further insight into the development and structure of the generative design application is presented in Section 4.5. This constrained solution space served as a control in the study since all participants used this set of data as the starting point.

4.2.1 Generative Design Study Platform

As described above, Autodesk Fusion 360 was leveraged to create the initial set of 1,000 design solutions. However, the design solutions were transferred to a static repository that was independent of Autodesk Fusion 360, which was then leveraged by the generative design application. This allowed the researcher to manipulate the solution space as needed such that the study was controlled between participants. This ensured that participants interacted with the same subset of solutions independent of their unique study inputs (e.g., design selection, time to complete a task, etc.). Furthermore, this allowed the researcher to adjust the solution space as needed to identify an optimized parsing without the constraints of external Autodesk software features.

4.3 Generative Design Study Overview

The participants of this study interacted with a constrained solution space, initially comprising 1,000 generative design solutions. Output was generated from an existing generative design system (e.g., Autodesk Fusion 360), with design solutions produced from minimal requirements, to mimic a new engineering project. This maximized the design space such that the software produced a large quantity of design solutions. All design solutions were then transposed into a script that only contained this specified design space, thereby creating a constrained design space (e.g., finite number of elements) such that all participants interacted with the same interface. Study instructions, that were provided to participants over email, subsequently directed participants through multiple iterations of the study that included various filter types and subsets of the initial solution space, outlined in Table 1 below.

Table 1: Study Structure Overview

Study Iteration	**Quantity of Design Solutions**	**Filter Types**	**Variables Captured**
1	1,000	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings
2	500	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings
3	250	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings
4	100	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings
5	50	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings

6	25	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings
7	10	- Material - Manufacturing Method - Mass - Max von Mises Stress - Volume - Parameter Similarity - No Filter	- Task Time - Detection Response Task (DRT) Response Time and Hit Rate - Secondary Task Accuracy - User Experience Ratings

4.4.1 Generative Design Study Sequence of Events

Upon signing a voluntary agreement indicating consent to participate in the study, each participant was assigned to one of seven possible study groups. Each study group correlated to one of the seven possible filter types, described above. The generated design alternatives were arranged and displayed to the participants based on one of these seven different filter types, which were (1) material, (2) manufacturing method, (3) mass, (4) Max von Mises stress, (5) volume, (6) parameter similarity, and (7) no filter. In all the filter types except the "no filter" type, cluster analysis was performed to create similar groups of solutions, which were used for organizing and uniformly down selecting the solution space for each iteration. The design space was reduced from iteration to iteration leveraging a cluster analysis approach, specifically using the Gower distance matrix and partitioning around medoids (PAM) clustering method, which allowed for consideration of both categorical and continuous variables (Botyarov & Miller, 2022). For example, the solution space was reduced from 1,000 in iteration one to 500 in iteration two, by performing a cluster analysis on the 1,000 design alternatives and then randomly selecting an equal number of alternatives from each cluster until there were 500 alternatives. This was repeated for each iteration and was preprogrammed into the generative design application such that all participants within a filter experienced the same design alternatives across all iterations. The cluster analysis was performed independently for each filter, where the variable used to form the clusters was the filter type parameter, except for the parameter similarity filter option, which considered all variables, and the no filter option, which did not consider any variables. The novel clustering approach developed for this study is further described and verified in Chapter 5. The variable(s) used to perform the cluster analysis were based on the filter type. *Material* filter was defined by clusters formed using only the material variable (e.g., cobalt chrome, Inconel 625, etc.). *Manufacturing method* filter was defined using clusters that were formed considering only the manufacturing method variable (e.g., additive, 5-axis milling, etc.). *Mass* filter was defined by clusters formed using the mass variable only. *Stress* filter was defined as clusters formed considering only the stress variable. *Volume* filter was defined using clusters based only on the volume variable. *Parameter similarity* filter was defined as presenting selected design alternatives sorted by clusters that were formed considering all design variables. *No filter* was defined as a random presentation of the selected design alternatives, but all participants were presented with the same randomization.

Once a participant was assigned to a study group (e.g., filter type), they continued using the same filter type throughout the entirety of the study, meaning all seven iterations were completed using the same filter type. In addition to study group assignment, each participant was given an identical set of instructions that guided them through each iteration. Every iteration began with a constrained design space comprised of a defined quantity of design solutions (e.g., 1,000, 500, 250, 100, 50, 25, and 10). The provided instructions directed participants through a task, called the design selection task, that concluded with the participant selecting one design alternative. In addition to the design selection task, the study included a secondary task, called the visual dissimilarity task, to further evaluate participant cognition for each iteration. Each task concluded with each participant being asked a series of survey questions that recorded their experience. Once the final iteration survey was completed, the subsequent iteration began. During each iteration the same set of metrics was be recorded, which are further described in subsequent sections. Figure 11 below visually depicts the sequence of events for the generative design study.

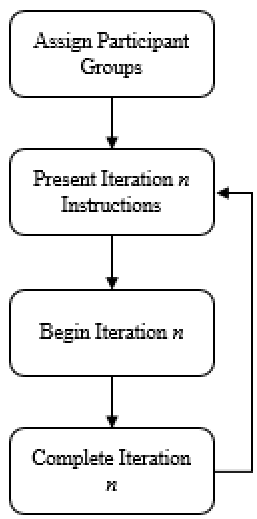


Figure 11: Study Sequence of Events

To obtain a larger data set for analysis, each participant was randomly assigned two filter types. The first time a participant completed the study, the generative design application would arrange the design alternatives in descending order based on the corresponding filter type for all seven iterations. Then, the second time a participant completed the study, the application would re-arrange design alternatives in descending order across all seven iterations based on the second assigned filter type. The different paths through the generative design application, via these filters, was enabled by the participant entering a token number value, that corresponded to a filter type, on the login page of the generative design application. The participants had no knowledge of the filter type to token number relationship since the researchers provided the token number to the participants as part of the study instructions. Table 2 shows the number of participants that completed each pairwise filter type combination.

Table 2: Pairwise Filter Combinations Completed by Participants

Filter Type	No Filt. *(N=12)*	Pa. Si. *(N=13)*	Ma. Me. *(N=17)*	Material *(N=14)*	Stress *(N=16)*	Mass *(N=13)*	Volume *(N=13)*
No Filter	--	--	--	--	--	--	--
Parameter Similarity	0	--	--	--	--	--	--
Manufacturing Method	2	3	--	--	--	--	--
Material	2	3	4	--	--	--	--
Stress	3	3	2	2	--	--	--
Mass	2	2	4	1	3	--	--
Volume	3	2	2	2	3	1	--

4.4.2 Dependent Variables

There are two types of usability measures, (1) those that measure the result of using the whole system (e.g., usability in use) and (2) measures of the quality of the user interface (e.g., interface usability) (Bevan, 2008). Usability in use is measured by effectiveness (e.g., task goal completion), and efficiency (e.g., resources used) (Bevan, 2008). The variables used to evaluate human cognition in this study included (1) task time (e.g., usability in use), (2) detection response task (DRT) response time (e.g., usability in use), (3) DRT hit rate (e.g., usability in use), (4) task accuracy (e.g., usability in use), and (5) user experience ratings (e.g., interface usability).

4.4.2.1 Task Time

Task time (in seconds) to complete the design selection task was measured for each participant for each iteration. A timer automatically began once the participant navigated to the design selection task page for a given iteration and all design solutions were loaded on the page.

This would ensure that external variables, such as lag in page loading time given the variability of participant technology did not impact the data collected. The timer was embedded within the code such that the participants did not see the timer. The timer automatically stopped, and duration was recorded once the participant completed the design selection task. Time (in seconds) was also used during the visual dissimilarity task for each iteration and was recorded in the same manner as for the design selection task. Task duration was used as an objective measure for efficiency and was therefore related to questionnaire scales assessing perceived efficiency. In this study, time to completion was a crucial measurement of cognition where faster completion times correlated to reduced cognitive workload when compared to slower completion times.

4.4.2.2 Detection Response Tasks (DRTs)

Detection Response Tasks (DRTs), which are often used in driving studies, provide a method for measuring the attentional effects of cognitive load in the given environment (Miller et al., 2018). During a detection response task, participants are presented with a sensory stimulus (e.g., light or vibration) during a prescribed interval with some randomness (e.g., every 10-15 seconds) and are requested to respond to it (e.g., pressing a button). Response times and hit rates are interpreted as indicators of the attentional effect of cognitive load (Stojmenova & Sodnik, 2018). Given that the primary task in this study was for the participant to identify a single design alternative, a supplemental detection response task, during the design selection task, measured the attentional effect of cognitive load. The detection response task in this study consisted of a 48x48 pixel static red box, with the text “Click Me” written on it, that appeared in the same area of the computer screen (lower right). Since a predictable interval was to be avoided, the box appeared randomly during a prescribed interval. To register a successful detection response, the participant had to press on the static red box within 10 seconds of it appearing. ***DRT response***

time (in seconds) was recorded from the onset of the stimuli until the participant pressed the box, where longer response times corresponded to higher cognitive workload. ***DRT hit rate (%)*** was computed as the percentage of stimuli the participant responded to within 10 seconds of onset within each iteration, where lower hit rates corresponded to higher cognitive workload.

4.4.2.3 Task Accuracy

Task Accuracy (binary yes/no) is a measure used to evaluate to what degree a participant has completed a given task correctly. For this metric to be effective, there needs to be a correct answer to a task. The primary task, specifically the design selection task, of each iteration is the selection of a design alternative. This task was a highly subjective task since each participant could leverage existing knowledge, creativity, and given filter type to select an alternative that met specified requirements. However, each iteration has at least one design alternative that met stated requirements, so task accuracy for the design selection task was evaluated where a correct response was the selection of a design alternative that complied with stated requirements. The visual dissimilarity task was presented to record another objective measure of cognition. In the context of generative design, being able to visually discern a difference between design alternatives is crucial because the intent is to visually evaluate different designs within the design space. Participants were shown four design alternatives from a given iteration and asked to select the most visually different. The four design alternatives were selected using the same cluster analysis method, where three of the designs were randomly selected from the same cluster and one of the designs randomly selected from a separate cluster. Unlike the design selection task, participants were only shown the image of each design alternative, meaning no additional information regarding the attributes of the design alternative was presented. Participant design selections for the visual dissimilarity task were then evaluated for accuracy, where the correct response was the selection of the design alternative that was from a different cluster than the others.

4.4.2.4 User Experience Ratings

Participant reported ***user experience ratings*** were measured by utilizing subjective metrics to evaluate experience with both generative design study tasks. Immediately after completing the design selection task for each iteration, participants were asked two questions on a 7-point Likert scale ranging from strongly disagree to strongly agree, specifically (1) “I am satisfied with the models generated by the generative design system” (i.e., satisfaction); and (2) “I am confident that the design alternative I selected meets the specified requirements” (i.e., confidence). Then, NASA Task Load Index (TLX) questions were used to collect perceived workload associated with the design selection task (Hart & Staveland, 1988). This same set of questions were asked following each iteration. Similarly, immediately after completing each iteration of the visual dissimilarity task, participants were asked the same NASA TLX questions.

The NASA TLX process has six indicators, which are (1) mental demand, (2) physical demand, (3) temporal demand, (4) performance, (5) frustration, and (6) effort (Prabaswari et al., 2019). For reference, the overall range of scores observed in the literature in 2015 from over 1,000 global NASA TLX scores from over 200 publications was 6.21 to 88.5 (Grier, 2015). NASA TLX scores are commonly interpreted as low cognitive workload (scores 0-9), medium cognitive workload (scores 10-29), somewhat high cognitive workload (scores 30-49), high cognitive workload (scores 50-79), and very high cognitive workload (scores 80-100) (Prabaswari et al., 2019). For a system to be usable and accessible participants should be able to use it to achieve their goals in an acceptable amount of time and be satisfied with the results

(Bevan, 2008). Therefore, user experience ratings helped gauge participant cognition during each iteration and were evaluated in conjunction with previously discussed objective measurements to identify an optimized parsing of spatially different generative design solutions. Figure 12 below describes the sequence of events within an iteration and where the described metrics were captured.

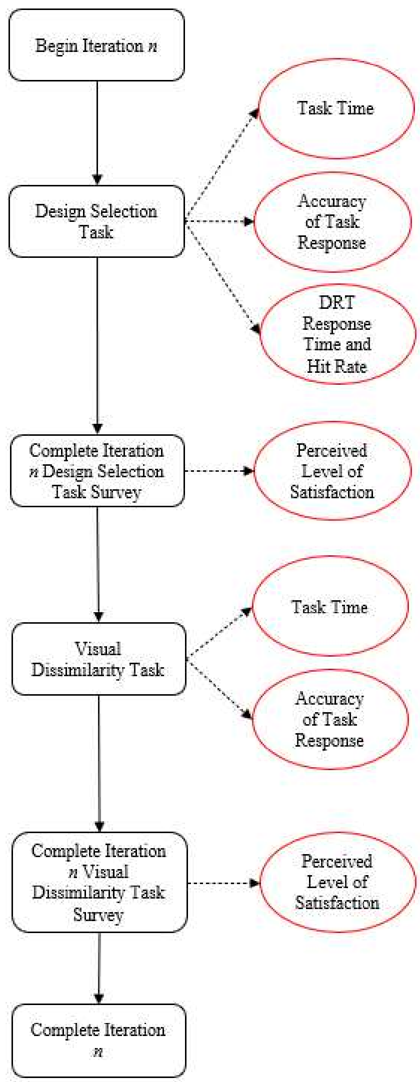


Figure 12: Iteration Sequence of Events and Variable Capture

4.4.3 Independent Variables

In this study, the independent variables were the ones that the researcher could influence, such as (1) filter type and (2) design solution space iteration.

4.4.3.1 Filter Type (between-subject): 7 Levels

The seven filters used in the study included (1) material, (2) manufacturing method, (3) mass, (4) Max von Mises stress, (5) volume, (6) parameter similarity, and (7) no filter. These filters were selected due to their use in existing generative design tools, such as Autodesk Fusion 360. Each design alternative had attributes related to these filter types such that the presented design space could be evaluated by participants. During the design selection task, participants were only able to leverage their assigned filter type throughout the entire task. This provided a method for comparing the efficiency of various filters between iterations and how they impacted cognition.

4.4.3.2 Design Solution Space Iteration (within-subject): 7 Levels

There were seven iterations of design solution space downselect. The first iteration of the study began with the entire design solution space, which consisted of 1,000 unique generative design solutions. Subsequent iterations consisted of a subset of generative design solutions (e.g., 500, 250, 100, 50, 25, and 10). Therefore, there needed to be a method for design solution downselect such that participants were presented the same subset of design solutions every iteration per filter type. Analysis was performed on the initial set of design solutions to identify design regions. This was accomplished via a clustering approach developed for this research, specifically the Gower distance matrix and partitioning around medoids (PAM) clustering method, that is further described in Chapter 5. A subset of solutions in each subsequent cluster were preserved and displayed in the subset. The selected process was repeated for every iteration in the study. Task type (e.g., design selection task or visual dissimilarity task) was also a controlled variable, but these tasks were modeled separately and not included as an independent variable within the study models.

4.4.4. Data Analysis Approach

The data analysis approach consisted of both generalized linear mixed models and binary logistic regression models. Generalized linear models (GLMs) enable a unified likelihood regression approach for the analysis of a wide range of continuous and discrete outcomes (Dean & Nielsen, 2007). GLMs are defined by a random component which specifies the probability distribution of the response variable, a systematic component, which specifies a linear function of the explanatory variables used as the predictor, and the link function which relates the systematic component and the mean value of the random component (Dean & Nielsen, 2007). Therefore, GLMs are comprised of several types of dependent variables where the linear predictor includes only fixed effects, with applications in the analysis of binomial and count data (Hedeker, 2005). Generalized linear mixed models (GLMMs) are a variation of GLMs to include one or more random effects (Bolker et al., 2009). As a result, GLMMs have the potential to deal with data involving multiple sources of error, such as repeated measures within participants and are especially useful for analysis of correlated nonnormal data (Tuerlinckx et al., 2006). Given the advantages of GLMMs, these models were used for analyzing study task completion times for the design selection task, DRT, and visual dissimilarity task, in the context of filter type and design solution space iteration. In addition, GLMMs were used for analyzing participant survey responses (e.g., perceived workload) for the design selection task and visual dissimilarity task along with participant reported confidence and satisfaction with the design selection task.

The binary logistic regression model is related to GLM statistical models (Ely et al., 1996). A pivotal characteristic that differentiates binary logistic regression from other GLMs is the type of dependent variable incorporated into the model, specifically a dependent variable in a binary logistic regression has two levels (e.g., yes or no) (Harris, 2021). In addition to a binary dependent variable, a binary logistic regression has at least one independent variable that is used to evaluate the values of the dependent variable (Harris, 2021). Therefore, dependent variables are dichotomous and independent variables can be either categorical or continuous (Ranganathan et al., 2017). Binary logistic regression models rely on several assumptions, including independent observations, no perfect multicollinearity, and linearity (Harris, 2021). In the context of this study, DRT hit rate and the accuracy of task responses for the design selection task and visual dissimilarity tasks are dependent binary variables since all have two categories, yes or no. Similar to GLLMs, the independent variables for the binary logistic regression are filter type and design solution space iteration.

4.4.4 Participant Selection

Participants were recruited through aerospace industry company networks and the graduate degree program in the Department of Systems Engineering at Colorado State University. There were a total of 64 participants recruited, but only 49 participants fully completed the study, meaning 49 participants completed all seven iterations for two different filter types. Therefore, the data analysis was only conducted on the 49 complete responses.

Of the total 49 participants, there were 15 industry experts and 34 students. Most of the students were enrolled in the online CSU Systems Engineering graduate program, which predominately serves working professionals. Thirty of these students reported being students with a full-time job, while four self-reported as full-time students. There were 35 (71.4%) total participants who reported their profession was engineering, for which 17 worked in the aerospace sector. Participant ages ranged from 21 to 69 years old, with a mean age of 35.88 (SD = 10.8). Further participant demographics are provided in Table 3, which were collected after participants completed the study.

Table 3: Participant Demographics

Variable	**Count**	**Percent**
Gender		
Male	34	69.4%
Female	15	30.6%
Education		
High School Diploma	4	8.2%
Bachelor's Degree	24	49.0%
Master's Degree	20	40.8%
Doctorate Degree	1	2.0%
Generative Design Familiarity		
Not familiar at all	20	40.8%
Slightly familiar	18	36.7%
Moderately familiar	9	18.4%
Very familiar	2	4.1%

4.5 Generative Design Application Overview

Prior to discussing how the data associated with this study was analyzed, it is imperative to review how the generative design application mentioned previously was constructed and deployed. A generative design software application was developed for use in this study to replicate the down select process of the generative design solution space. As such, the researchers did not evaluate the user interaction of inputting requirements to generate the initial entire solution space since this task was performed prior to generative design study deployment. The application was developed using the Python programming language in Visual Studio. The database component of the application, for recording study data was pgAdmin4, which is an open-source management tool for PostgreSQL. Once the application was tested and ready for participant input, it was hosted on Heroku, which is a cloud hosting platform for applications. The design space used in the application followed the procedure for developing computational models defined by (Cagan et al., 1997) which involved, (1) definition of search space for the design alternatives via qualitative knowledge on a well-defined search space (i.e., Autodesk Fusion 360); (2) problem formulation involving the design of the generative design application; (3) solution of the problem through an appropriate technique (i.e., clustering method); and (4) verification and critique to establish whether the solution indeed satisfied the design objectives, as confirmed in (Botyarov & Miller, 2022).

The generative design solution space was developed using Autodesk Fusion 360 that initially produced 1,000 design alternatives for an aircraft engine loading bracket. An image of each design, as well as properties related to the design alternative's material, manufacturing method, mass, Max von Mises stress, and volume were captured and exported from Autodesk Fusion 360 into a standalone environment. Each aircraft loading bracket design alternative was assigned a unique identification number for use within the generative design application. The flow of the application is presented in Figure 13 below.

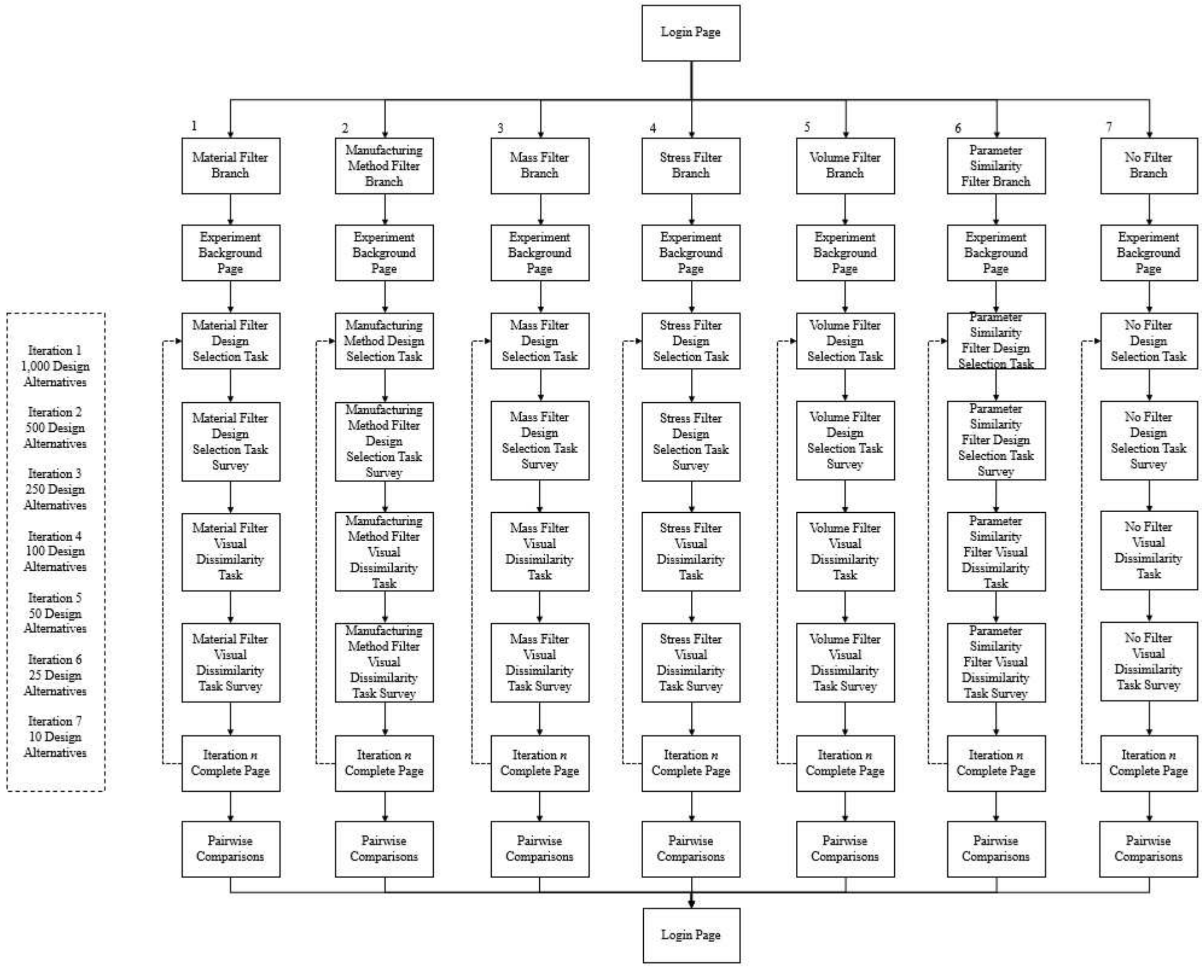


Figure 13: Generative Design Application Flow

4.5.1 Generative Design Application Login Page

To access the application, participants used a PC or Mac on their preferred browser and navigate to generative-design-study.herokuapp.com, which led to the login page indicated by Figure 14 below.

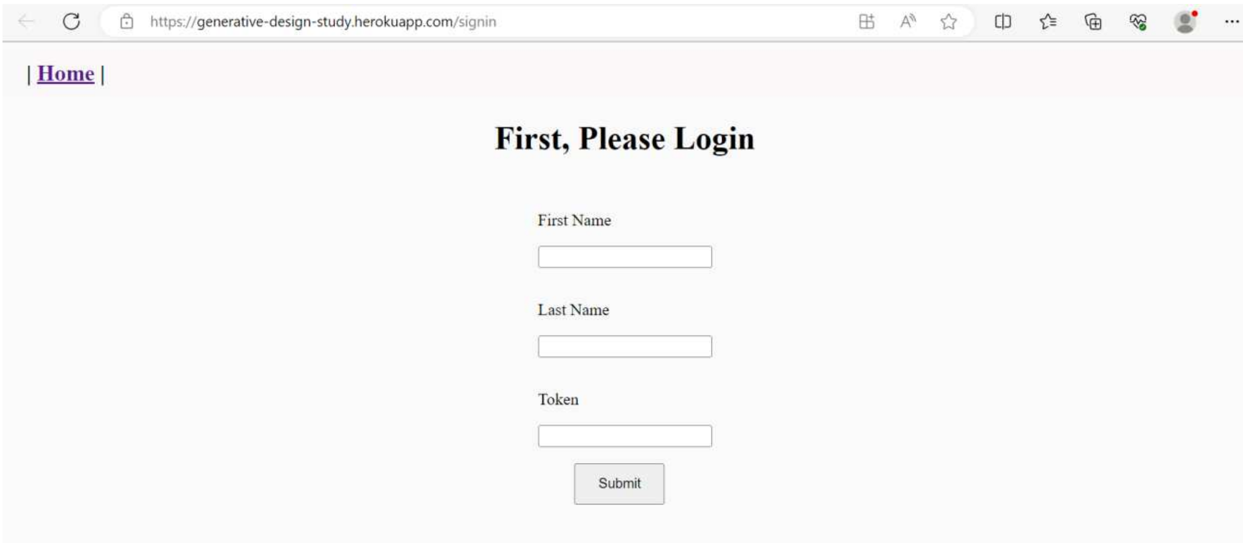


Figure 14: Generative Design Application Login Page

To complete the login page, participants input their first name along with their last name. The final item to input was a token, which was an integer value provided to participants by the researcher as part of an instructions email communication prior to the study. Each participant was assigned two tokens, that were completed one at a time. The token value corresponded to a branch of the experiment, correlating to one of the filter types discussed. For example, if a participant would input the value "1" in the token field on the login page, they would be directed to the material filter branch. The design solution space was parsed based on the filter type, and then design alternatives were presented in a sorted fashion by the corresponding filter type. Design alternatives were presented low to high for numerical variables (e.g., continuous variables including mass, stress, volume, and parameter similarity) and alphabetically from A-Z for categorical variables (e.g., nominal variables including material and manufacturing method). For the no filter branch, the order the design alternatives were randomized to sort the design space, however the randomized order was then identical for that filter branch for all participants (e.g., the randomization occurred once). Figure 15 below further illustrates the described concept along with presenting the token integer for each associated filter type branch (e.g., material filter branch correlates to token one, etc.).

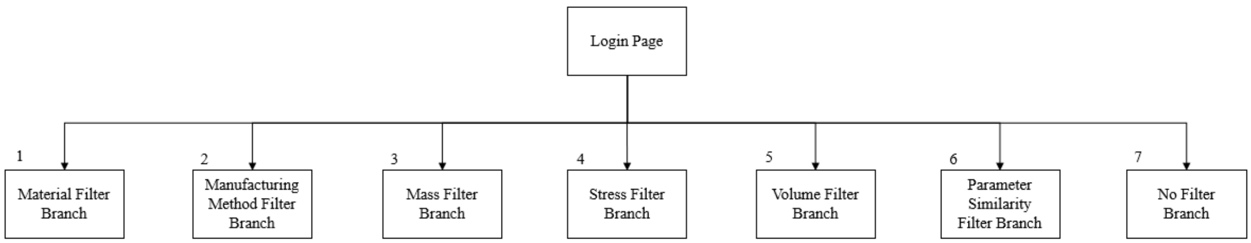


Figure 15: Generative Design Application Filter Type Branches

Figure 16 below provides an example of the completed login page, upon which participants would click the ‘Submit’ button to be taken to the next page for the specified filter branch.

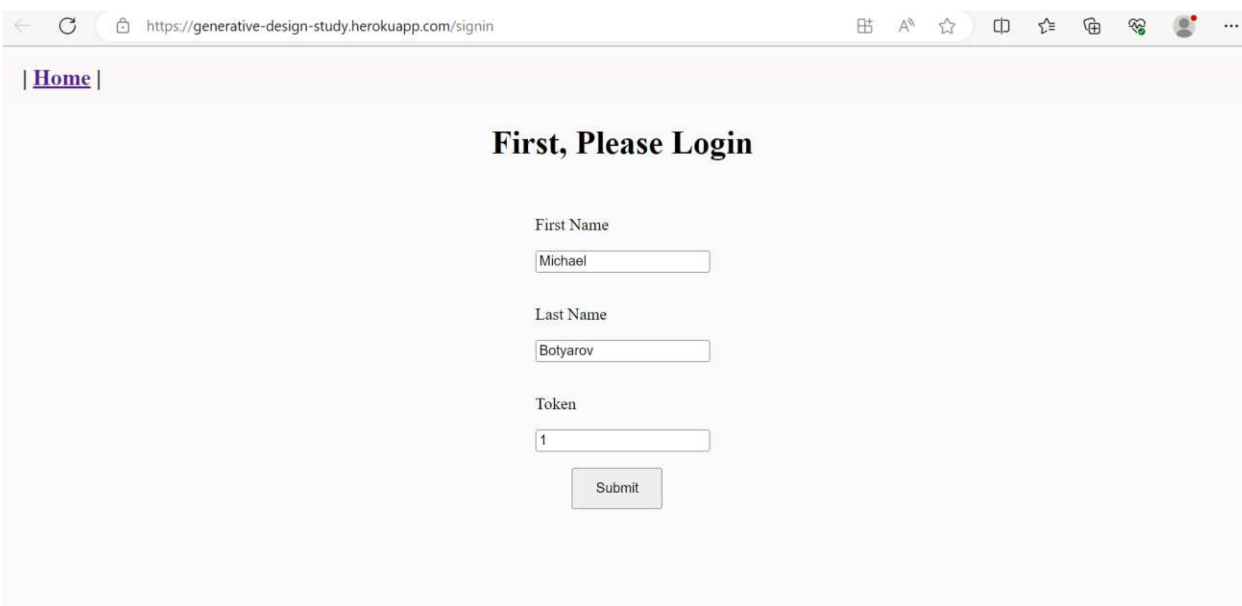


Figure 16: Completed Generative Design Application Login Page

4.5.2 Generative Design Study Background Page

Once the login page was completed, participants were routed to the study background page. The purpose of the study background page was to provide participants with context regarding the experiment, specifically a short synopsis of generative design and present the design problem. The design problem surrounded an aircraft engine loading bracket and mentioned that the participants would act as designers attempting to select an aircraft loading bracket design produced via a generative design system to move forward with in the systems engineering life cycle given a set of customer requirements. Once the participants completed reviewing this page, the 'Begin' button would be clicked to initiate the study, which is shown in Figure 17 below.

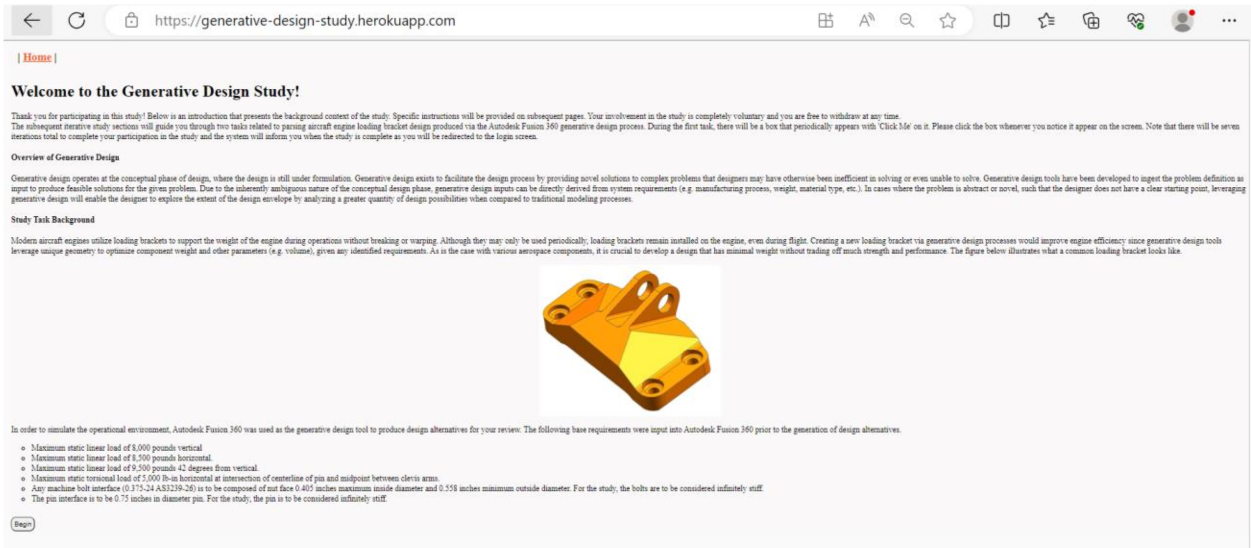


Figure 17: Generative Design Study Background Page

4.5.3 Design Selection Task Page(s)

Once the experiment background page was reviewed, participants were routed to the design selection task page. The quantity of design alternatives presented on the design selection task page depended on the iteration as presented in Figure 13 (Section 4.5). Each design alternative had associated metadata presented under it (e.g., material, manufacturing method, etc.). Design selection task requirements (e.g., customer design requirements) were presented in a static pane on the left side of the design selection task and varied depending on iteration. All design selection task requirements were the same for each level of the iteration (e.g., iteration one…, iterations seven) between filter type branches. The design selection task requirements were in a static pane such that they were available to be viewed by participants at any time while on the page, regardless of if participants scrolled to explore all design alternatives presented. Each design selection page had a timer that started once all design alternatives were loaded, thereby removing any data impact associated with the variability of network connections.

In addition to the timer, the design selection task pages had a 48x48 pixel static red box that appeared in the same area of the computer screen, specifically on the pane under the design selection task requirements, simulating a detection response task (DRT). Since a predictable interval was to be avoided, the box appeared randomly during a 10-15 second interval. To register a successful detection response, the participant had to press on the static red box within 10 seconds of it appearing. The data captured related to the DRT included time until first red box appeared, time until first red box clicked, time until *n* red box appeared, time until *n* red box clicked, and time until design selection task was completed. All mentioned data were a timestamp from the initial design selection task page loading, so calculations were conducted to derive the time it took a participant to respond to each DRT instance. Figure 18 below presents an example of a design selection task page with the DRT visible.

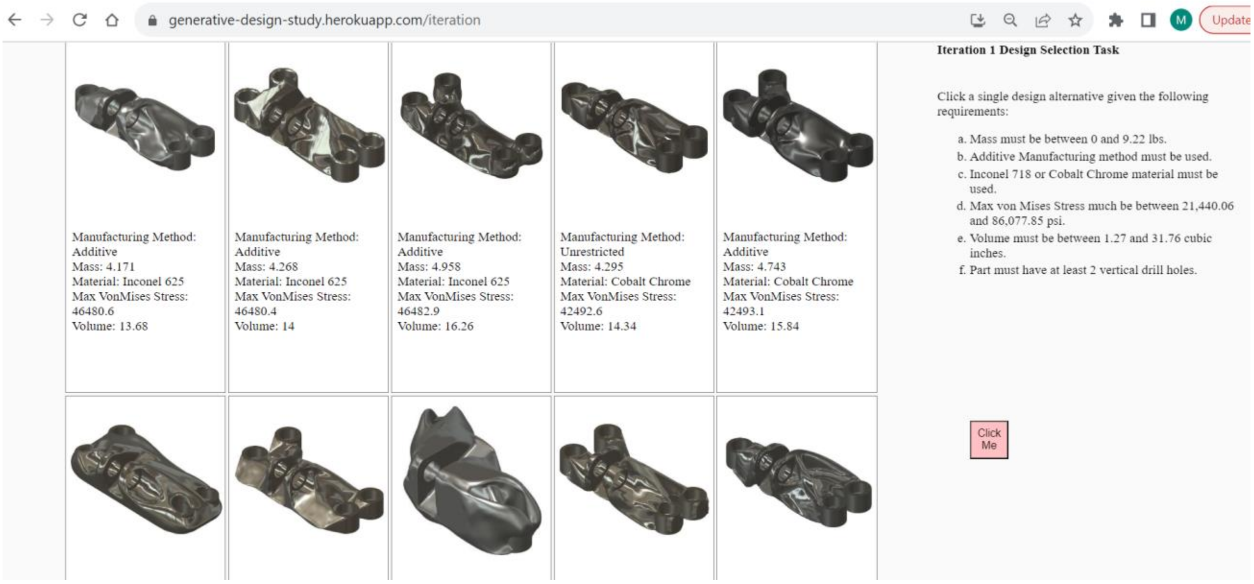


Figure 18: Design Selection Task Page

As participants arrived at a final design alternative selection, they clicked the design alternative that corresponded to their desired selection. This action prompted a confirmation dialogue box to appear, from which participants would confirm their selection. Once the selection was confirmed via this method, the timer would stop and the corresponding data was recorded, indicating the time it took a participant to complete the design selection task for the corresponding iteration. The associated application database subsequently captured the design alternative that was selected, along with additional information for traceability purposes (e.g., participant identifier, iteration number, and filter type branch). DRT response times were also recorded in the application database. Figure 19 below displays the confirmation dialogue box that participants completed to confirm their design selection.

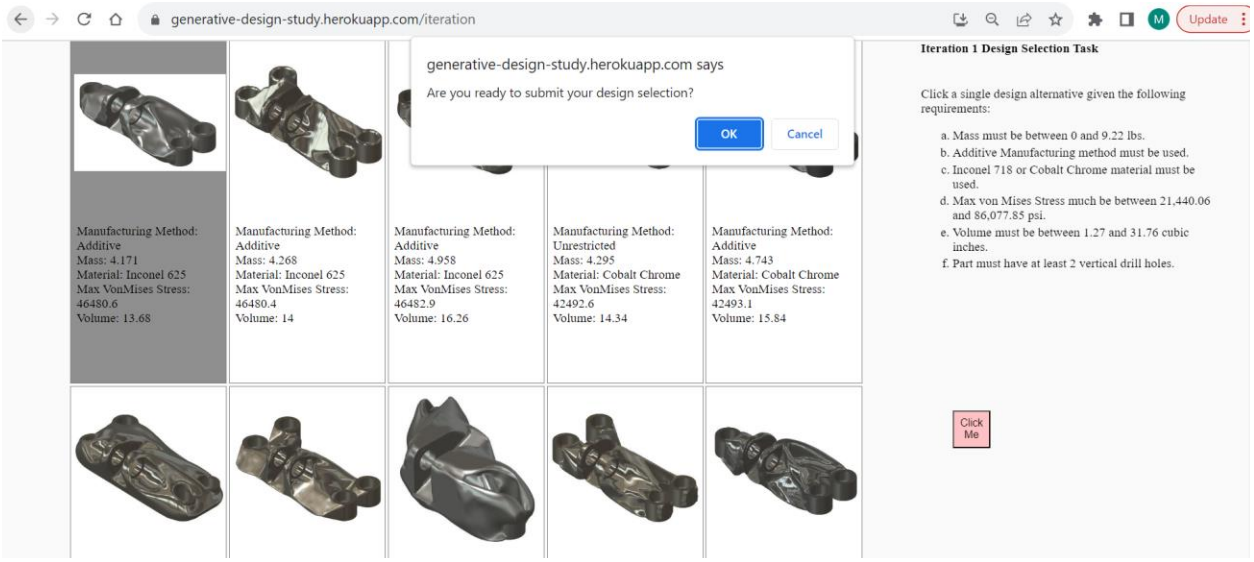


Figure 19: Design Selection Task Page Confirmation Dialogue

4.5.4 Design Selection Task Survey Page(s)

Upon the completion of the design selection task, for each iteration, participants were routed to the design selection task survey page. A survey was presented subsequently upon completion of design selection task such that the task experience could be recorded immediately. Participants were asked several questions on a 5-point Likert scale from strongly disagree to strongly agree. To answer these questions, participants selected a radio button that corresponded with their response (e.g., strongly agree). Figure 20 below presents the 5-point Likert scale survey questions that were presented for each design selection task survey.

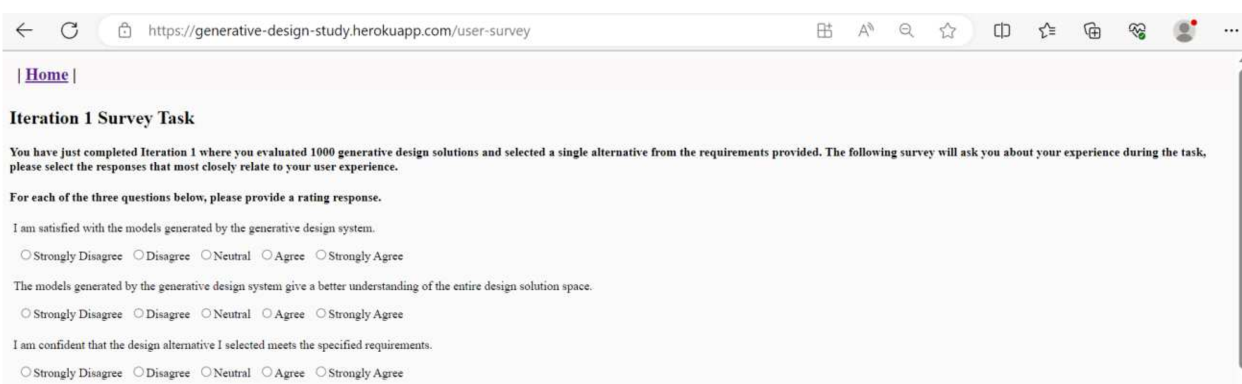


Figure 20: Design Selection Task Survey Page

After the Likert scale questions, participants were presented with a set of NASA Task Load Index (TLX) questions that were used to collect perceived workload associated with the design selection tasks. To answer these questions, participants were presented with a number line from zero to 100 in increments of five where zero was very low or poor and 100 was very high or excellent. Participant survey question responses were then recorded in the application database. Figure 21 below presents the NASA TLX questions that were presented for each design selection task survey.

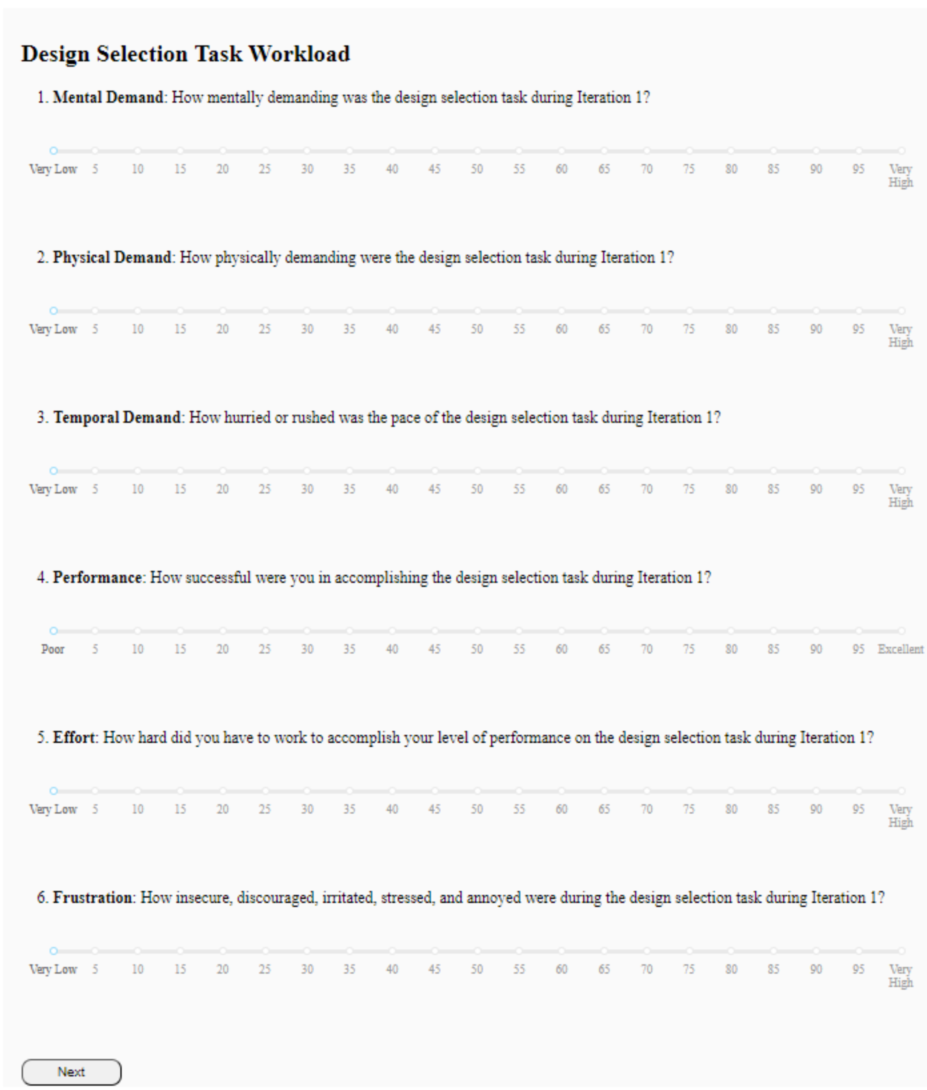


Figure 21: Design Selection Task NASA TLX Questions

4.5.5 Visual Dissimilarity Task Page(s)

Once the design selection task survey was completed, participants were routed to the visual dissimilarity task page. The quantity of design alternatives from which the visual dissimilarity alternatives were selected from depended on the iteration as presented in Figure 13 (Section 4.5). The design alternatives presented for the visual dissimilarity task between filter type branches had variation, depending on how clustering was conducted specific to this task for each iteration. To select the design alternatives that were presented during the visual dissimilarity task, the design solution space for each iteration (e.g., 1,000 solutions to 10 solutions) was clustered using the Gower distance matrix and partitioning around medoids (PAM), where the only variable considered was the one correlated to the filter type branch. For example, considering the mass filter branch during iteration one, the 1,000 design alternatives were parsed using the specified clustering method considering only the mass variable. Therefore, the clustering would occur only in relation to the mass variable and the alternatives presented as part of the visual dissimilarity task, three would be presented from one cluster and the fourth would be presented from a different cluster. The design alternative attribute data was not displayed as part the visual dissimilarity task since the objective was to determine if participants could visually discern the difference between clusters. Each visual dissimilarity task page had a timer that started once all design alternatives were loaded, thereby removing any data impact associated with the variability of network connections. The accuracy of the visual dissimilarity task was evaluated by comparing a participant's selection against the single design alternative that was from a unique cluster in relation to the other three design alternatives. Participant design alternative selection and time to select a design alternative were recorded in the application database. Figure 22 below presents an example of the visual dissimilarity task page.

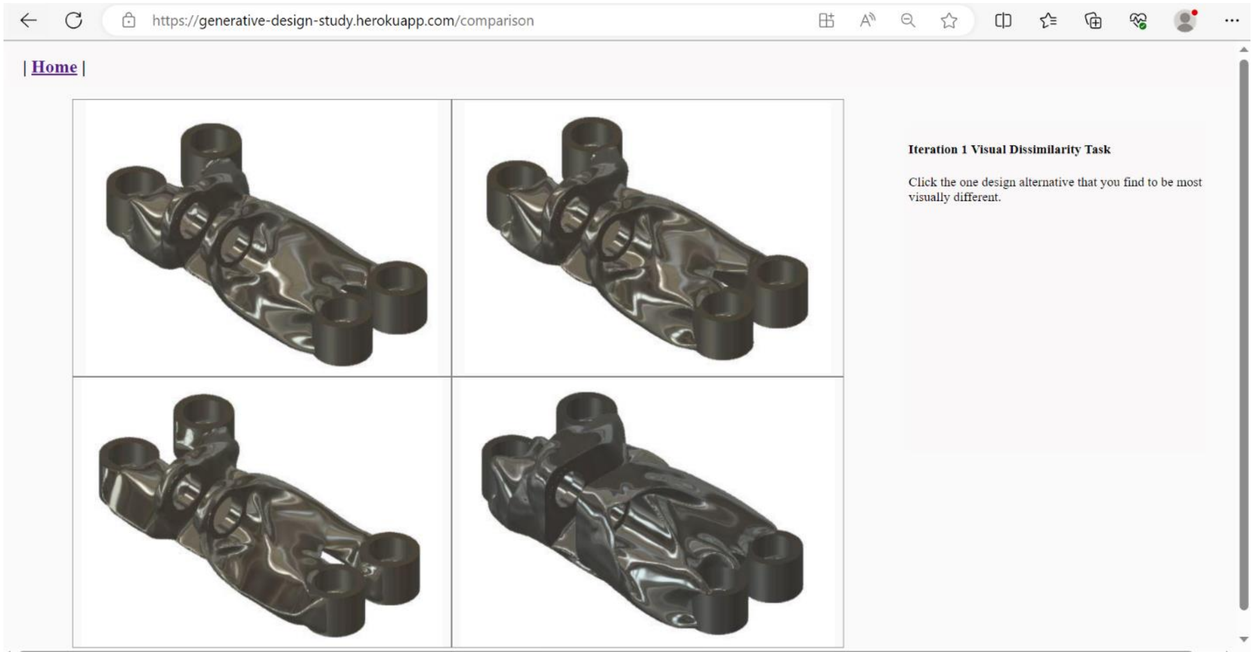


Figure 22: Visual Dissimilarity Task Page

4.5.6 Visual Dissimilarity Task Survey Page(s)

Upon the completion of the visual dissimilarity task, for each iteration, participants were routed to the visual dissimilarity task survey page. A survey was presented subsequently upon completion of visual dissimilarity task such that the task experience could be recorded immediately. Participants were asked a question on a 5-point Likert scale ranging from strongly disagree to strongly agree. To answer this question, participants selected a radio button that corresponded with their response (e.g., strongly agree). Figure 23 below presents the 5-point Likert scale survey question that was presented for each visual dissimilarity task survey.

Figure 23: Visual Dissimilarity Task Survey Page

After the Likert scale questions, participants were presented with a set of NASA Task Load Index (TLX) questions that were used to collect perceived workload associated with the visual dissimilarity tasks. To answer these questions, participants were presented with a number line from zero to 100 in increments of five where zero was very low or poor and 100 was very high or excellent. Participant survey question responses were recorded in the application database. Figure 24 below presents the NASA TLX questions that were presented for each visual dissimilarity task survey.

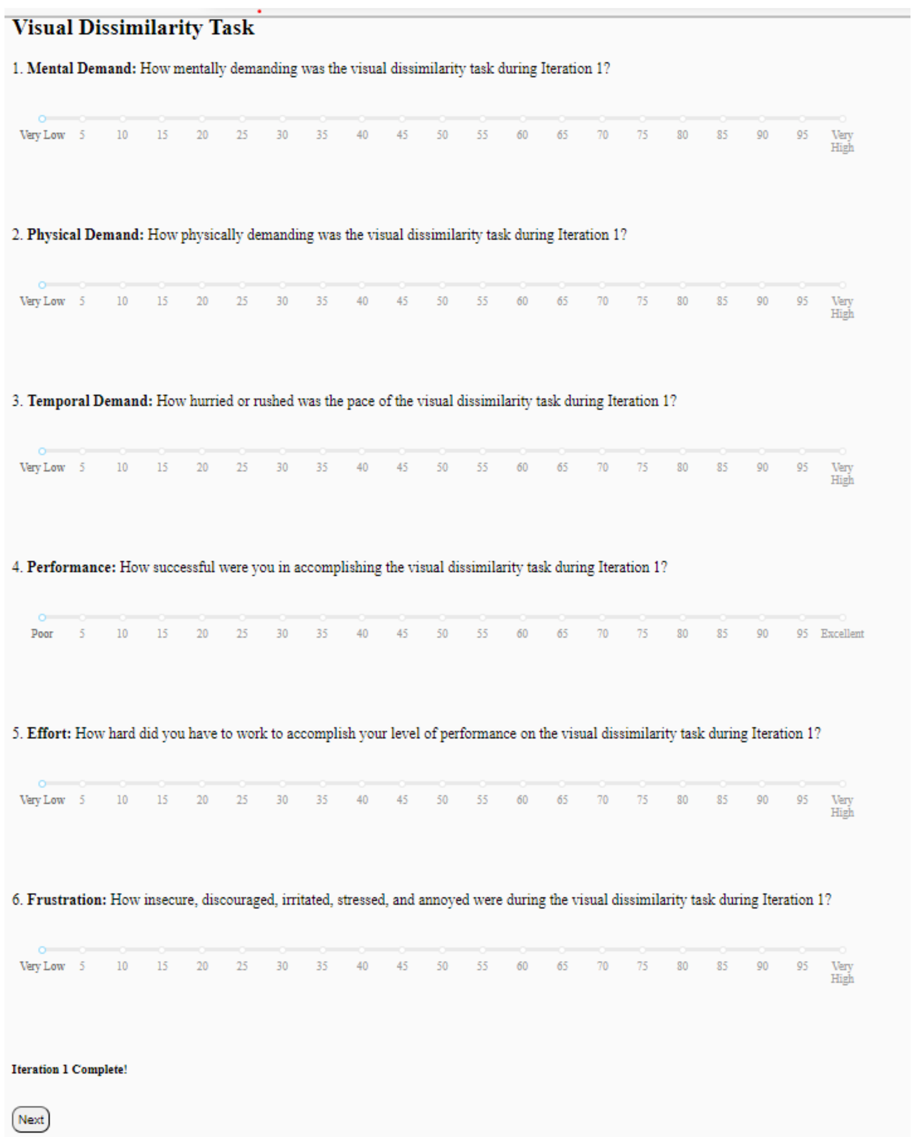


Figure 24: Visual Dissimilarity Task NASA TLX Questions

4.5.7 Iteration *n* Complete Page

The study was comprised of seven iterations. Upon the completion of each iteration, participants were routed to an iteration completion page. This page indicated that iteration *n* was successfully completed, and that the next iteration would begin once the 'Next' button was clicked. Once the 'Next' button was clicked, iteration $n + 1$ would begin, and the loop would continue until iteration seven was completed. For each iteration, after iteration one, participants would then be routed to the design selection task for the subsequent iteration (e.g., iteration two) and complete the same sequence as indicated in Figure 13 (Section 4.5). Figure 25 below displays an iteration *n* complete page.

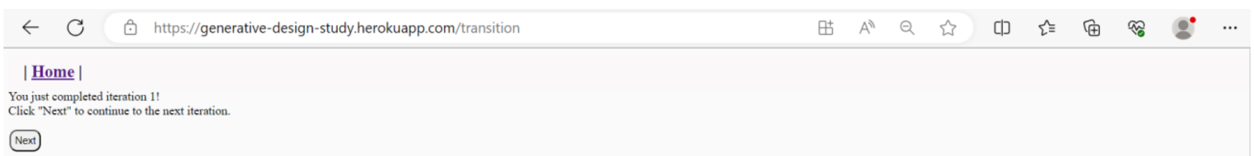


Figure 25: Iteration *n* Complete Page

4.5.8 Pairwise Comparisons Page

As part of the NASA TLX methodology, a set of 15 pairwise combinations were presented to participants after the completion of iteration seven to select an option from each pair that had the greatest effect on the workload during study as a whole. Participant pairwise comparison responses were recorded in the application database. Figure 26 below displays the pairwise comparisons that were presented to participants.

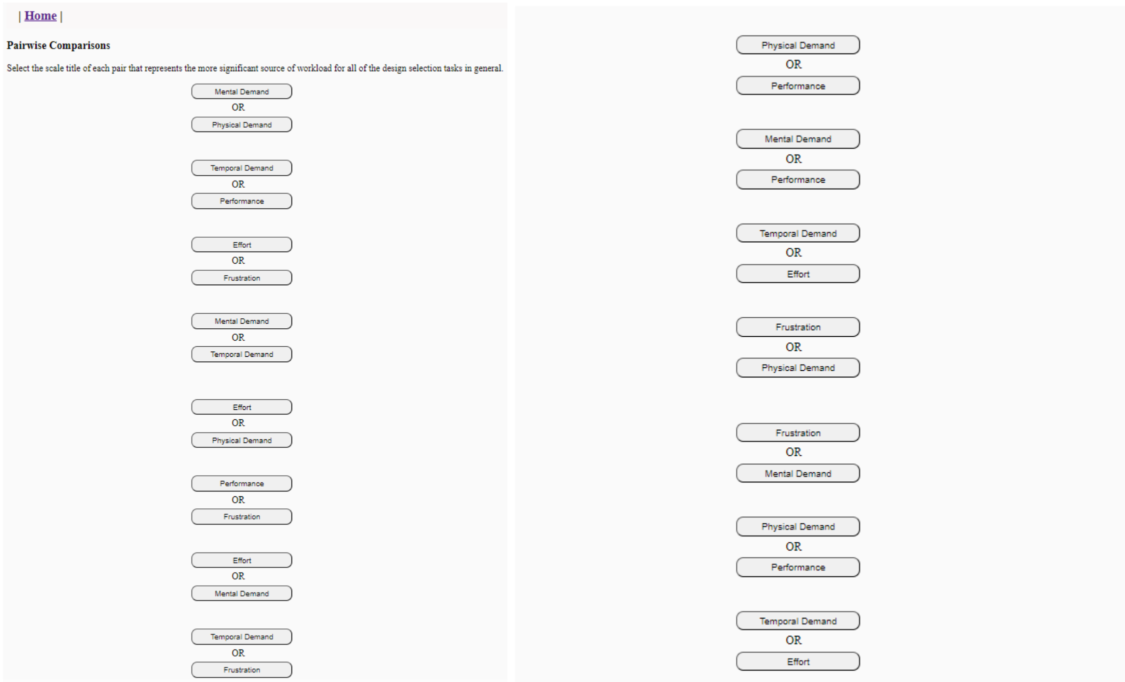


Figure 26: Pairwise Comparisons Page

4.5.9 Generative Design Study – Application Completion

Once the participants completed the pairwise comparisons for their assigned filter type branch, the 'Finish Study' button was clicked, which then routed participants back to the application login page described in Section 4.5.1. Upon returning to the application login page, participants completed the experiment again, as required, given another token (e.g., filter type branch) provided by the researchers.

4.6 Generative Design Study Limitations

The design of the generative design study yielded potential limitations, including (1) system-of-system constraints, (2) participant fatigue, and (3) participant learning. Each potential limitation is addressed further in this section.

4.6.1 System-of-System (SoS) Constraints

Individual components are a part of a larger system, which in turn can be part of an even larger system. This is known as a system of systems (SoS), where individual components perform a specialized function that enable the overall system to execute additional desired capabilities greater than simply the sum of the individual component functions. Generative design systems themselves are primarily concerned with the development of an individual component, however the requirements used to establish constraints for the component being developed contain SoS considerations. The objective of this study was to evaluate generative design at the component level, however it can be assumed that the requirements presented to the participant during the design selection task encompass overall SoS considerations. Furthermore, each participant was given context of the overall system (e.g., background on the design problem) being designed such that trade-off considerations could occur as part of the design selection task.

4.6.2 Participant Fatigue

Participants completed the study in a single session, so there was the possibility of fatigue as participants progress through each iteration. To mitigate potential impact on the data, each participant was assigned a filter type prior to beginning the study, thereby reducing 49 possible iterations to complete to seven iterations. Furthermore, every participant completed the entire study under the same circumstances, therefore any fatigue experienced was more or less evenly distributed between participants. Since there was more than one participant in every filter type group, individual levels of fatigue were further minimized due to the quantity of participants and the randomization of token (e.g., filter type branch) assignments.

4.6.3 Participant Learning

Due to the magnitude of the study, participants were asked to complete the study two times, therefore bringing up the concern of participant learning leading to skewed data, specifically improved task times because of participants learning study structure, not participants being influenced by study variables. This concern was addressed by having participants complete the study for different filter types in different filter type pairwise combinations (e.g., mass and volume, mass and material, etc.), instead of repeating the same combination. Furthermore, since participants performed the study multiple times, any potential data skewness was viewed as noise since it was distributed across all filter types.

4.7 Generative Design Study Outcomes & Transferability

Outcomes of this study included (1) a recommended generative design solution space size and an associated filter type for generative design tools that will subsequently result in a more efficient systems engineering process and (2) a structured approach that captures how to parse a generative design solution space. Both outcomes have transferability to generative design systems (e.g., Autodesk Fusion 360) and to design applications outside of solely structural design tasks. Generative design systems are being applied to other engineering domains, including electrical design, fluid-flow optimization, optics, and acoustics (Brossard et al., 2020). In fact, there are generative design applications in in almost every creative discipline including art, music, software, and engineering design (BuHamdan et al., 2021). This is driving a need for a generalized approach to generative design given application to complex optimization problems arising across multiple domains (Nagaraj & Werth, 2020).

4.7.1 Recommended Generative Design Solution Space Size and Filter Type Combination

Current generative design systems place a significant cognitive workload on the designer due to the magnitude of data and design alternatives presented, particularly during conceptual design when requirements are evolving and unrefined. To streamline the systems engineering process, by reducing time lost due to design cognitive overload, it is imperative that generative design systems are structured to produce output that is easily digestible by the designer. This study evaluated designer cognitive performance at various design quantity levels and filter types, such that an optimized combination of the two was identified, resulting in a more efficient systems engineering process.

4.7.2 Generative Design Solution Space Parsing Method

The main output of this study was a generative design solution space parsing method that describes how to parse the output of a generative design problem. Specifically, a generative design solution space parsing method was developed using the Gower distance matrix and partitioning around medoids (PAM). Application of this method resulting in the creation of multiple subsets of the initial generative design solution space that retained novel design alternatives given the comprehensive range of design alternative variables within each subset (e.g., material, manufacturing method, etc.). This method can be integrated into generative design systems that already exist, such as Autodesk Fusion 360, by allowing the designer to specify the desired quantity of solutions to be displayed (e.g., 100). The generative design system can then implement the developed clustering algorithm internally, providing the requested subset of solutions to the designer.

4.7.3 Transferability to Other Domains & Applications

The generative design solution parsing method was completed in the R statistical software program, which is free to download and compatible across operating systems. While this method can easily be implemented into a generative design software package (e.g., Autodesk Fusion 360), it also is a highly accessible method that can be implemented at no cost to a user by using R. This no cost method would allow any user with a solution space of any size and any mix of variable types to reduce a sample space to a desired size. Moreover, this method is innovative in that the algorithm is not restrictive of properties or inputs. System properties could be of any variable type, such as qualitative or quantitative, which is unique to partitioning around the medoids compared to other clustering methods. This PAM method could easily be paired with user inputs of design restrictions or requirements, such that the data would filter solutions that only met the user specifications and then cluster accordingly. Hence, this approach provides flexibility to software developers implementing the algorithm and users using the algorithm.

Given the flexibility of the parsing method, it can be applied to other domains and applications outside of generative design where a large set of data needs to be evaluated to identify relationships or data regions. Specifically, the aerospace domain could benefit from the implementation of the developed parsing method during the systems engineering process. The systems engineering process in aerospace involves multiple stakeholders and subsequently multiple design criteria. By applying the parsing method from this study, engineers will be able to have a structured approach to parsing a large set of data with differing variable types. This will translate to a more efficient systems engineering process, resulting in cost and schedule savings. Additive manufacturing is becoming increasingly prevalent for aerospace application, including space missions, providing capabilities including innovation, rapid development optimization, and affordability (Clinton et al., 2018). Abstract aerospace scenarios, such as printing ad hoc tools for emergency situations during deep space missions, could apply the parsing method from this study for generative design applications and other big data applications. Leveraging this parsing method will enable a more efficient use of scarce resources and allow the users (e.g., astronauts) to efficiently parse and locate a generative design alternative for their mission need(s).

CHAPTER 5: CLUSTERING METHODOLOGY DISCUSSION

5.1 Introduction

This chapter presents a novel approach to generative design solution space reduction by offering a flexible, efficient, and accessible method. Since generative design is most powerful in the conceptual design phase of the systems engineering life cycle, problem requirements are often ambiguous, allowing for the creation of a larger quantity of design alternatives. Therefore, it was imperative that a method for reducing the quantity of generative design solutions was created. The presented method of generative design solution space reduction leverages the clustering analysis technique with a combination of the Gower distance matrix and partitioning around medoids in an iterative process. This iterative generative design solution space reduction method retains the originality of unique design solutions, while simultaneously reducing the quantity of design solutions presented. Design originality is maintained since the clustering process groups similar designs into clusters, from which a systematic reduction of similar designs can be achieved, thereby leaving novel solutions from the design envelope. This chapter presents the effectiveness of this method. An original solution space of 1,000 aircraft engine loading brackets was downselected to 10 unique and representative solutions, whose attributes were summarized and compared to the initial data set to further highlight design originality. This method is innovative in that there are no strict requirements for the data input, unlike other clustering methods, such that design properties can be quantitative and/or qualitative and there are no restrictions to the number of properties clustered.

The objective of this chapter is to validate how the proposed clustering method of partitioning around medoids via the Gower distance matrix is able to systematically reduce a generative design system solution space to a handful of novel design alternatives. The research questions evaluated were, (1) How can cluster analysis be used to reduce a generative design solution space with mixed variable type (i.e., qualitative and quantitative) properties, and (2) Does the reduced solution space offer unique solutions while retaining the originality of the original solution space? Subsequent sections present a case study of a solution space with 1,000 designs and implements the cluster analysis technique of partitioning around medoids to downselect the solution space iteratively to 10 solutions. Then, the properties of these final 10 solutions are evaluated to determine their uniqueness between each other and breadth across the original solution space.

5.2 Cluster Analysis Overview

Clustering is a data analysis tool that facilitates grouping a data set into several groups such that, under a definition of similarity (e.g., element parameters, etc.), similar elements are grouped in the same cluster and dissimilar elements are grouped in different clusters (Guha et al., 2003). Clustering facilitates the identification and definition of patterns between data elements. The realization of patterns via clustering subsequently allows significant conclusions to be made that may have not been apparent prior to the clustering effort. Furthermore, cluster analysis groupings are based on the similarity of the entire data set, as opposed to the individual variables that comprise each observation, thereby providing a comprehensive analysis of the entire data set (Leonard & Droege, 2008).

5.2.1 Clustering Algorithms

Clustering involves selecting an algorithm that defines the method used for determining the distance between the collected data points and subsequently grouping data points in close proximity to each other (e.g., density of points, intervals, and statistical distributions) into clusters. The main approaches to clustering include (1) density-based clustering (e.g., DBSCAN), (2) hierarchical clustering (e.g., divisive and agglomerative), and (3) partition-based clustering (e.g., k-means, partitioning around medoids [PAM]).

(1) Density-Based Clustering. Density-based spatial clustering (DBSCAN) is a method best for analyzing arbitrarily shaped clusters and detecting outliers. DBSCAN algorithms are efficient at locating high-density regions, therefore are utilized frequently in use cases that require anomaly detection. DBSCAN relies on two input parameters, which are (1) eps (e.g., distance that specifies neighbors, where two data points are considered neighbors if the distance between them is equal to or less than the eps) and (2) minPts (e.g., minimum quantity of data points needed to define a cluster). With this method, a point will be assigned to a cluster if it is close to many points from that cluster. Aside from analyzing arbitrarily shaped clusters and outliers, DBSCAN is advantageous since the quantity of clusters does not need to be specified beforehand.

(2) Hierarchical Clustering. Hierarchical clustering is a method that seeks to construct a hierarchically arranged sequence of partitions for some given data set (Köhn & Hubert, 2015). Similar to other clustering methods, hierarchical clustering is an iterative method that repeats a calculation of distance measures (e.g., Euclidean distance) between objects, and between clusters once objects begin to be grouped into clusters. Hierarchical clustering produces a unique set of clusters by sequentially pairing variables (e.g., categorical or continuous), clusters, or both (Köhn & Hubert, 2015). Every iteration begins with the correlation matrix, all clusters and variables are tried in all possible pairs, and that pair that produces the highest average inter-correlation within the trial cluster is chosen as the new cluster (Bridges, 1966). The outcome of

hierarchical clustering is a dendrogram, similar to a tree-plot, where iteration of hierarchical clustering is depicted as a fusion of two branches (e.g., clusters) of the tree into a single one (Bridges, 1966). The two main approaches to hierarchical clustering are (1) agglomerative (e.g., sequentially merging similar clusters) and (2) divisive (e.g., starting with one cluster and successively splitting subsequent clusters), although the latter is rarely used in practical applications. Unlike other clustering methods in which a single set of mutually exclusive and exhaustive clusters is formed, hierarchical clustering produces smaller, less inclusive clusters through larger more inclusive clusters until all variables are clustered into a single group (Köhn & Hubert, 2015).

(3a) Partition-Based Clustering: K-Means. K-means clustering is a method that partitions a data set into *k* clusters such that data points in one cluster are similar and data points in another cluster are farther apart, where the similarity of two points is calculated as the distance between them. K-Means clustering focuses on minimizing the distances between data points within a cluster while simultaneously maximizing the distance between clusters (e.g., Euclidean, Manhattan, and Gower). K-Means clustering is iterative, leveraging an expectation-maximization algorithm to (1) randomly select centroids for each cluster specified, (2) calculate the distance of all data points to the centroids, (3) assign data points to the closest cluster, and (4) identify new cluster centroids by taking the mean of all data points in that cluster. The steps are then repeated until all points converge and the cluster center location remains unchanged between iterations. Unlike DBSCAN, k-means clustering is sensitive to outliers and requires the number of clusters to be determined in advance, with the elbow method being a common choice for selecting the optimal quantity of clusters for a given data set. However, k-means clustering is advantageous for large data sets with continuous variables and the method guarantees convergence.

(3b) Partition-Based Clustering: Partitioning Around Medoids. The Partitioning Around Medoids (PAM) algorithm is a clustering method that maps a distance matrix into a specified number of clusters (Van der Laan et al., 2003). A major advantage of the PAM algorithm is that it enables clustering relative to any specified distance matrix (e.g., Gower distance matrix), thereby allowing it to be less sensitive to outliers (Van der Laan et al., 2003). The objective of the PAM algorithm is to minimize the average dissimilarity of objects to their closest selected object by identifying a sequence of objects (e.g., medoids) that are centrally located in clusters. PAM is considered a more robust algorithm when compared to k-means since data points are used as medoids, therefore they are not introduced randomly. Once the optimal number of clusters is identified (e.g., *k* value), the first medoid is assigned as the data point that has the smallest distance to all other data points, so it is in the center of the data set. The subsequent medoid is introduced such that the total distance of each data point to their nearest medoid is reduced. All subsequent medoids in the building phase follow a similar cycle. Then, different data points are tested as medoids and if a different point reduces the total within cluster distance the PAM method swaps the medoid with that point (Manvailer, 2019). All possible combinations in the given data set are tested, therefore only one solution is possible.

5.2.2 Distance Matrices

When presented with a data set comprised on multiple observations, prior to classifying the observations into clusters, it is crucial to identify the dissimilarity between individual observations. The computation of the degree of dissimilarity between observations results in a distance matrix, which can subsequently be leveraged via a prescribed method (e.g., k-means algorithm) to identify clusters specific to the initial data set. The selected method for measuring distance influences the data analysis process since it defines how similarity is calculated, thereby influencing the shape of the clusters. Common distance measures include (1) Euclidean, (2) Manhattan, and (3) Gower distances. Additional dissimilarity measures exist, such as correlation-based distances for gene expression data analysis, however for brevity they will not be addressed here.

(1) Euclidean Distance. The Euclidean distance is a clustering distance measure that is used to calculate distance between data points that consist of continuous variables. Prior to executing any calculations, the data set is column standardized to remove differences due to measurement units and scale. Euclidean distance is computed using the formula below (Equation 1) via squares of distances such that computing square roots is avoided and subsequent theorems and algorithms are simplified.

Equation 1: Euclidean Distance

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{1}$$

Although simple to calculate, this method is a nonproportional distance measure which assumes variables are uncorrelated and it loses sensitivity quicker than other distance measures as data heterogeneity increases.

(2) Manhattan Distance. The Manhattan distance is a clustering distance measure that is also used to calculate distance between data points that consist of continuous variables. The Manhattan distance is computed using the formula below (Equation 2) via summing the horizontal and vertical distances between observations.

Equation 2: Manhattan Distance

$$d_{man}(x, y) = \sum_{i=1}^{n} |(x_i - y_i| \tag{2}$$

When compared to the Euclidean distance, the Manhattan distance gives less weight to outliers since the calculation does not utilize squared differences and it retains sensitivity as data heterogeneity increases. However, the Manhattan distance is still a nonproportional distance measure.

(3) Gower Distance. Gower distance is a clustering distance measure that is used to calculate distance between two data points that have a mix of categorical and numerical elements (Gower, 1971). Gower distance between two points is calculated by comparing each element and computing a term. If a data element is numeric, the term is the absolute value of the difference divided by the range (Gower, 1971). If a data element is non-numeric (e.g., categorical) the term is 1 if the elements are different or the term is 0 if the elements are the same (Gower, 1971). The Gower distance is then computed using the formula below (Equation 3) as the average of partial dissimilarities across data points where each partial dissimilarity (e.g., Gower distance) ranges in [0 1] (Gower, 1971).

Equation 3: Gower Distance

$$d(i,j) = \frac{1}{p}\sum_{i=1}^{p} d_{ij}^{(f)} \qquad (3)$$

Once distance metrics are scaled (e.g., fall between zero and one) for each variable type, a linear combination using user-specified weights (e.g., an average) is calculated to create the final distance matrix. The subsequent Gower distance matrix of similarities between all pairs of sample units has been shown to be positive semidefinite (Gower, 1971).

5.2.3 Selecting Optimal Number of Clusters

A common approach to selecting the number of clusters is to first compute the distance matrix on the data set, and then use a clustering algorithm (e.g., hierarchical, k-means) with 1 to

n (e.g., 10) clusters. The clustering with 1 to n groups are then evaluated, most often via (1) Elbow Method or (2) Silhouette Width to select the optimal number of clusters from 1 to n.

(1) Elbow Method. The elbow method aims to identify the optimal quantity of clusters of a given data set by exploring the percentage of variance explained as a function of the number of clusters (Bholowalia & Kumar, 2014). The percentage of variance explained by the clusters is plotted against the number of clusters (Bholowalia & Kumar, 2014). The primary clusters will add substantial information, but at some point, the marginal gain of adding additional clusters will drop dramatically, giving an angle (e.g., elbow) in the graph. After the drop, subsequent higher values (e.g., more clusters) will plateau or only provide marginal gain, which would provide no additional value to the cluster analysis. This initial decrease on the graph indicates the elbow of the graph, which correlates to the optimal number of clusters (e.g., k value) for the given data set.

(2) Silhouette Width. The silhouette width method, best used with the PAM method, aims to identify the optimal quantity of clusters of a given data set by providing a visualization of how well each data point lies within its cluster. The silhouette width value is calculated using the formula below (Equation 4) with output values ranging from -1 to 1.

Equation 4: Silhouette Width

$$s = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4}$$

Positive silhouette width values are preferable since they indicate that the sample is far away from the neighboring clusters (Batool & Hennig, 2021). Therefore, the k value that corresponds to the largest silhouette width indicates the optimal quantity of clusters for the given data set (Batool & Hennig, 2021).

The following sections describe the set-up of the case study used to create the generative design solution space for the subsequent proposed systematic approach for generative design solution space reduction.

5.3 Application to Generative Design Study

A case study was used to implement the proposed systematic solution space reduction. Generative design was first used to create the case study's solution space of 1,000 aircraft engine loading brackets. Cluster analysis was then used across six iterations to downselect the solution space to 10 unique aircraft engine loading brackets.

5.3.1 Creation of Original Solution Space

Modern aircraft engines utilize loading brackets to support the weight of the engine during operations without breaking or warping. Although they may only be used periodically, loading brackets remain installed on the engine, even during flight. Creating a new loading bracket via generative design processes would improve engine efficiency since generative design tools leverage unique geometry to optimize component weight and other parameters (e.g., volume), given any identified requirements. As is the case with various aerospace components, it is crucial to develop a design that has minimal weight without trading off much strength and performance.

To simulate the operational environment, the following requirements were input into Autodesk Fusion 360 prior to the generation of design alternatives:

- Maximum static linear load of 8,000 pounds vertical.
- Maximum static linear load of 8,500 pounds horizontal.

- Maximum static linear load of 9,500 pounds 42 degrees from vertical.
- Maximum static torsional load of 5,000 lb-in horizontal at intersection of centerline of pin and midpoint between clevis arms.
- Any machine bolt interface (0.375-24 AS3239-26) is to be comprised of nut face 0.405 inches maximum inside diameter and 0.558 inches minimum outside diameter. For the study, the bolts are to be considered infinitely stiff.
- The pin interface is to be 0.75 inches in diameter pin. For the study, the pin is to be considered infinitely stiff.

With these initial requirements input into Autodesk Fusion 360, the generative design solution space was created. Minor adjustments, such as factors of safety values, were iteratively performed to generate additional generative design solutions needed to arrive at the desired quantity from which to begin this case study.

5.3.2 Iterative Downselect

As mentioned previously, the generative design solution space was initially comprised of 1,000 generative design solutions. All design options were then transposed into a separate repository that only contained this specified design space, thereby creating a constrained design space (e.g., finite number of elements) from which the proposed systematic generative design solution space reduction method could be applied. From there, the proposed systematic approach to generative design solution space reduction iteratively reduced the initial 1,000 solution set generative design space to 500, then 250, 100, 50, 25, ultimately concluding with 10 design alternatives. These design alternative quantities were selected to replicate various levels of the design selection process to closely resemble the manual generative design solution space reduction a designer would follow to arrive at a single design alternative.

Since every iteration of the case study has an increasingly smaller subset of solutions from the design envelope, the objective of the downselect process was to systematically reduce the quantity of generative design solutions while retaining the originality of solutions presented in the entire data set. In other words, the downselect process aimed at partitioning the design envelope into regions that contained similar solutions and proportionally reducing the quantity of solutions retained from every region for each subsequent iteration. Clustering was the chosen method to analyze the generative design solution space given that clustering is a data analysis tool that facilitates grouping a data set into several groups such that, under a definition of similarity (e.g., element parameters, etc.), similar elements are grouped in the same cluster and dissimilar elements are grouped in different clusters (Guha et al., 2003). Leveraging the clustering data analysis method allowed for an objective and systematic partitioning of the solution space for the downselect process.

5.3.3 Data Structure

Prior to selecting the clustering method (e.g., algorithm), it was imperative to consider the entirety of the given data set and the associated data types. Data types are classified into four broad categories, including (1) nominal, (2) ordinal, (3) discrete, and (4) continuous data. Nominal data is qualitative and is used to label data without a preference for ordering or hierarchy (e.g., operating system, computer type, etc.). Ordinal data is also qualitative, however, it is differentiated from nominal data since order matters (e.g., rankings, superiority, etc.). Discrete data is quantitative and only includes integers, meaning discrete data cannot be further fractionalized (e.g., number of participants in a study). Continuous data is also quantitative, however, it can be meaningfully partitioned and can be measured on a continuum (e.g., temperature, height, etc.). In the context of this case study, each design alternative contained an associated set of attributes including material type (nominal variable), manufacturing method (nominal variable), volume (continuous variable), mass (continuous variable), and stress (continuous variable). Therefore, the clustering method selected had to be able to consider the impact of both categorical and continuous data types. Previously mentioned clustering methods, such as k-means clustering, have limitations as they are only applicable to continuous data types. Since generative design output solutions have both categorical and continuous data, a combination of the Gower distance matrix and the PAM algorithm was the clustering method chosen for this case study.

5.4 Results from Application to Generative Design Study

The statistical software R (version 4.1.0) was used to cluster and downselect the original entire 1,000 generative design solution space of the aircraft engine loading brackets across six iterations to eventually downselect to 10 solutions, which was based on the five attributes (two nominal and three continuous) of each loading bracket captured in Table 4 below.

Table 4: Variable Definitions for Entire Solution Space Properties

Variable	**Definition**	**Used in Cluster Analysis**
Outcome ID	Unique solution ID, 1 to 1000	No
Material	Aluminum 7175 T73 0 Hot Formed *(N = 4)* Aluminum AlSi10Mg *(N = 8)* Cobalt Chrome *(N = 152)* Inconel 625 *(N = 147)* Inconel 718 *(N = 168)* Inconel 718 Plus *(N = 73)* Iron, Cast *(N = 6)* Iron, Ductile *(N = 29)* Iron, Malleable *(N = 54)*	Yes

	Stainless Steel 17-4 PH *(N = 147)* Stainless Steel AISI 304 *(N = 8)* Stainless Steel AISI 440C, Welded *(N = 40)* Stainless Steel 440C *(N = 22)* Titanium 6A1-4V *(N = 142)*	
Manufacturing Method	2 Axis Cutting 2.5 *(N = 61)* 2.5 Axis Milling *(N = 49)* 3 Axis Milling *(N = 109)* 5 Axis Milling *(N = 36)* Additive *(N = 550)* Die Casting *(N = 60)* Unrestricted *(N = 135)*	Yes
Volume *(in^3)*	[5.25 – 57.94]; $\mu = 16.52$; $\sigma = 7.62$	Yes
Mass *(lbmass)*	[0.84 – 17.67]; $\mu = 4.51$; $\sigma = 2.36$	Yes
Max von Mises Stress *(psi)*	[29,008 – 127,998]; $\mu = 53{,}759$; $\sigma = 16{,}167.53$	Yes

5.4.1 Reduction from 1,000 to 10 Solutions

The solution space reduction was completed across six iterations in R, where iteration 1 downselected from 1,000 to 500 solutions, iteration 2 from 500 to 250, iteration 3 from 250 to 100, iteration 4 from 100 to 50, iteration 5 from 50 to 25, and iteration 6 from 25 to 10. Note, that for mixed data, variable structures should be of numeric and factor type. Within each iteration, there were three primary steps described below.

Step 1: Find Optimal K. Gower's coefficient was used to compute pairwise distances, or dissimilarities, between each solution, which were stored in a dissimilarity matrix. Next, average silhouette widths were computed using partitioning around the medoids on the dissimilarity matrix for 2 through 10 clusters of the data, a for loop was used to execute this. To assist in the silhouette width analysis, these silhouette widths were plotted to visualize the distance between the cluster groups. For this, a larger average silhouette width is ideal, representing a larger distance between the cluster groups. Hence, the *k* with the greatest silhouette width value was selected.

Step 2: Use K to Form Clusters. Based on the *k* selected in step 1, a final partitioning (i.e., clustering) around the medoids fit was conducted on the data and the cluster group number was added as a column to the data.

Step 3: Downselect. A for loop was used to randomly select an even number from each cluster, such that the number randomly selected multiplied by the number of clusters equaled the size of the targeted downselect. For example, in downselecting to 500 solutions in iteration 1, 50 solutions were randomly sampled from each of the 10 clusters. Note, in cases where there were not enough solutions in a cluster, which happened a few times as the solution space got smaller, as many solutions as possible were selected from the small cluster and a larger number of solutions were sampled from the other clusters to still achieve the desired downselect quantity.

An example of the R code used in Iteration 1 is provided in Figure 27, which can be used as a template for future applications of this method.

```r
## Initiate Environment ##
library(cluster); library(tidyverse) # for cluster analysis and data wrangling
set.seed(8) # for consistency in random downselect
data <- read.csv('DesignSpace_1000Solutions.csv')
## Step 1: Find Optimal K ##
dissimilarityMatrix <- daisy(data[, -1], metric = 'gower') # pairwise dissimilarities
silAvgWidth <- vector() # create vector to store for loop values
for(i in 2:10) { # compute silhouette widths for 2 to 10 clusters
  pam.fit <- pam(dissimilarityMatrix, k = i, diss = TRUE) # fit for cluster i
  silAvgWidth[i] <- pam.fit$silinfo$avg.width # save silhouette width i
}
plot(1:10, silAvgWidth, # higher avg. silhouette width better
     type = 'b', pch = 19, col = 'blue',
     xlab = 'Number of Clusters (k)', ylab = 'Average Silhouette Width')
## Step 2: Use k to Form Clusters ##
pam.fit <- pam(dissimilarityMatrix, k = 10, diss = TRUE) # use k from previous plot
clusplot(data[, -1], pam.fit$clustering, color = TRUE, main = '') # visualize clusters
data <- cbind(data, cluster = pam.fit$clustering)
## Step 3: Downselect ##
data500 <- data.frame() # create data frame to store data
for (i in 1:10) {
  temp <- sample_n(data %>% filter(cluster == i), 50) # randomly select 50 per cluster
  data500 <- rbind(data500, temp) # save solutions from cluster i
}
data <- data500 %>% select(-c('cluster')) # remove cluster variable
rm(list = ls()[!(ls() %in% c('data'))]) # remove all non-data objects from environment
## …repeat steps 1 through 3 five more times to iterate to 10 final solutions ##
```

Figure 27: R Code for Solution Space Reduction Using Partitioning Around Medoids with Gower Distance

A visualization of the clusters formed around the data points (i.e., generative design solution alternatives) can be useful to see the partitions and variability explained by the cluster analysis. However, with a very large sample space such as the original 1,000 solutions, this visualization may not show the clusters in a two-dimensional space as distinctly in Figure 28.

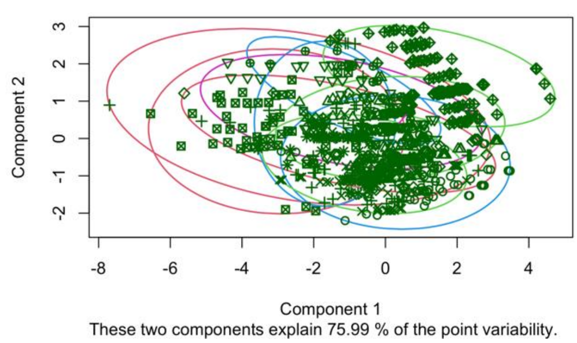


Figure 28: Clusters for 1000 Solution Space with First Two Principal Components

In each downselecting iteration, it should be expected that a different number of clusters will likely fit the data best. For example, in this case study, the number of clusters selected for each iteration were as follows: iteration 1 (k = 10), iteration 2 (k = 9), iteration 3 (k = 10), iteration 4 (k = 9), iteration 5 (k = 9), and iteration 6 (k = 6). These average silhouette width plots used to select each iteration's k value are provided in Figure 29.

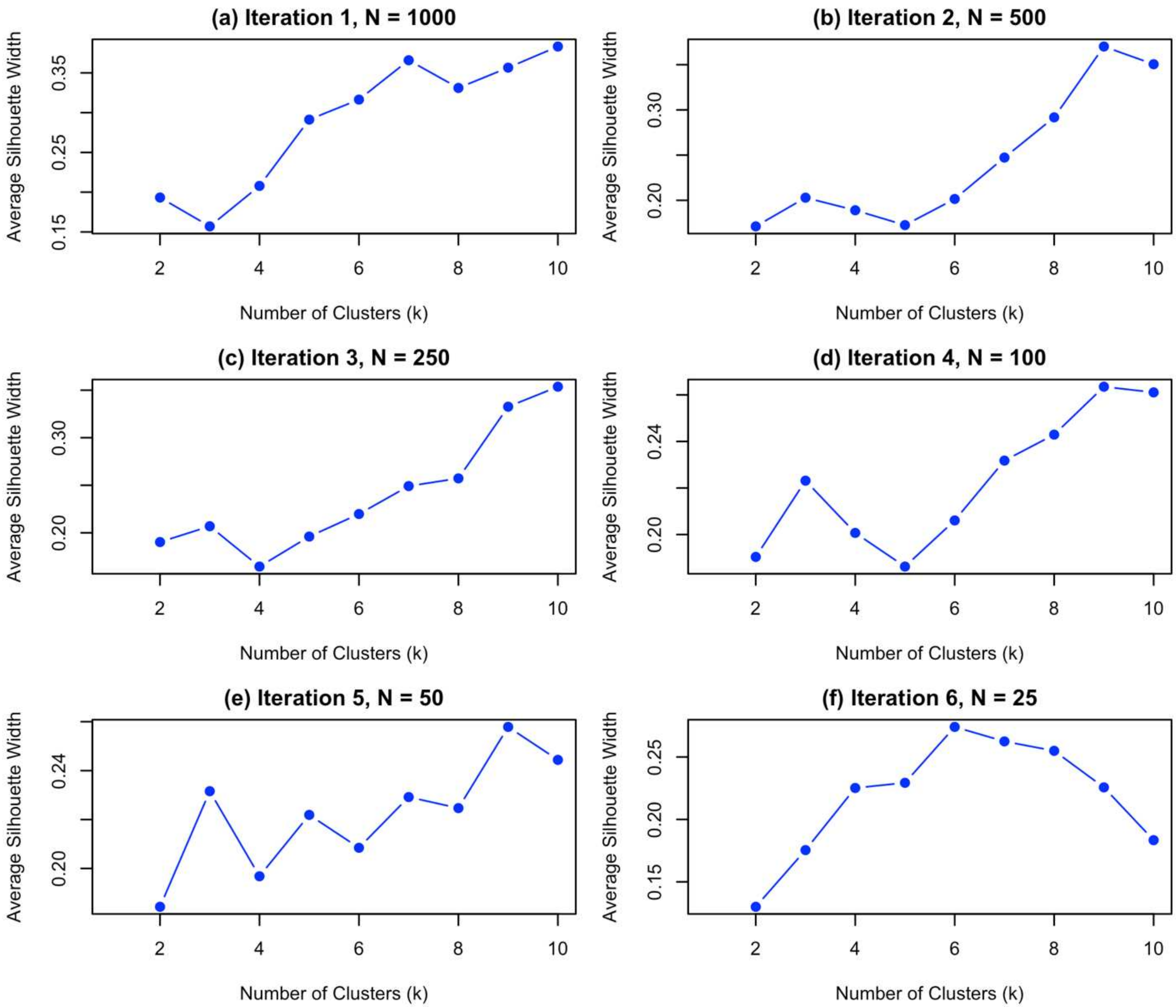


Figure 29: Average Silhouette Widths for k = [2, 10] for (a) 1000, (b) 500, (c) 250, (d) 100, I 50, and (f) 25 Solution Space

5.4.2 Validating Final Solution Space

The resulting solution space consisted of 10 loading brackets, which varied visually from one another as indicated by Figure 30. Each loading bracket had a unique shape clearly differentiating it from the other 10 solutions.

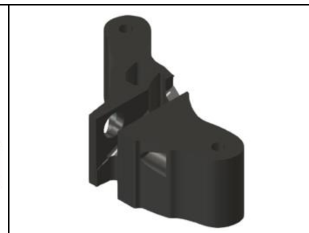
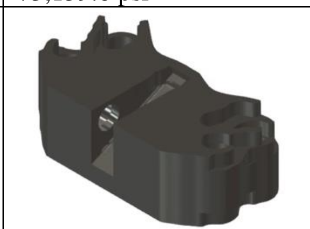
			
Outcome ID 32 Inconel 625 3 Axis Milling 33.00 in^3 10.069 lbs 37,185.2 psi	**Outcome ID 52** Inconel 718 3 Axis Milling 12.64 in^3 3.692 lbs 74,639.7 psi	**Outcome ID 220** Iron, Ductile Unrestricted 10.83 in^3 2.796 lbs 58,812.8 psi	**Outcome ID 315** Stainless Steel, 440C 2 Axis Cutting 21.70 in^3 6.077 lbs 49,964.8 psi
			
Outcome ID 361 Inconel 625 Unrestricted 10.26 in^3 3.127 lbs 61,961.3 psi	**Outcome ID 525** Titanium 6Al-4V Additive 9.79 in^3 1.567 lbs 73,139.6 psi	**Outcome ID 660** Stainless Steel 17-4 PH 2.5 Axis Milling 24.86 in^3 7.005 lbs 47,038.9 psi	**Outcome ID 665** Iron, Malleable Additive 23.38 in^3 6.040 lbs 32,581.8 psi
			
Outcome ID 729 Stainless Steel 17-4 PH Additive 15.81 in^3 4.456 lbs 40,469.0 psi	**Outcome 803** Cobalt Chrome 2.5 Axis Milling 40.67 in^3 12.180 lbs 43,584.1 psi		

Figure 30: Final 10 Loading Bracket Solution Space with Outcome ID, Material, Manufacturing Method, Volume, Mass, and Max von Mises Stress

Table 5 provides a comparison of the final 10 solution properties to the original solution space. These final 10 solutions varied within each of the five properties clustered upon and sufficiently maintained the original range of values of each property. For example, for the qualitative variables, the reduced solution space represented 8 of the original 14 material options and 5 of the original 7 manufacturing methods. For the quantitative variables, the ranges slightly reduced while the means and standard deviations remained similar.

Table 5: Properties of Final 10 Solutions

Variable	Final 10 Solutions	Original Solution Space
Material	Aluminum 7175 T73 0 Hot Formed *(N = 0)*	Aluminum 7175 T73 0 Hot Formed *(N = 4)*
	Aluminum AlSi10Mg *(N = 0)*	Aluminum AlSi10Mg *(N = 8)*
	Cobalt Chrome *(N = 1)*	Cobalt Chrome *(N = 152)*
	Inconel 625 *(N = 2)*	Inconel 625 *(N = 147)*
	Inconel 718 *(N = 1)*	Inconel 718 *(N = 168)*
	Inconel 718 Plus *(N = 0)*	Inconel 718 Plus *(N = 73)*
	Iron, Cast *(N = 0)*	Iron, Cast *(N = 6)*
	Iron, Ductile *(N = 1)*	Iron, Ductile *(N = 29)*
	Iron, Malleable *(N = 1)*	Iron, Malleable *(N = 54)*
	Stainless Steel 17-4 PH *(N = 2)*	Stainless Steel 17-4 PH *(N = 147)*
	Stainless Steel AISI 304 *(N = 0)*	Stainless Steel AISI 304 *(N = 8)*
	Stainless Steel AISI 440C, Welded *(N = 0)*	Stainless Steel AISI 440C, Welded *(N = 40)*
	Stainless Steel 440C *(N = 1)*	Stainless Steel 440C *(N = 22)*
	Titanium 6A1-4V *(N = 1)*	Titanium 6A1-4V *(N = 142)*
Manufacturing Method	2 Axis Cutting 2.5 *(N = 1)*	2 Axis Cutting 2.5 *(N = 61)*
	2.5 Axis Milling *(N = 2)*	2.5 Axis Milling *(N = 49)*
	3 Axis Milling *(N = 2)*	3 Axis Milling *(N = 109)*
	5 Axis Milling *(N = 0)*	5 Axis Milling *(N = 36)*
	Additive *(N = 3)*	Additive *(N = 550)*
	Die Casting *(N = 0)*	Die Casting *(N = 60)*
	Unrestricted *(N = 2)*	Unrestricted *(N = 135)*
Volume *(in^3)*	[9.79 – 40.67]; μ = 20.06; σ = 10.39	[5.25 – 57.94]; μ = 16.52; σ = 7.62
Mass *(lbmass)*	[1.57 – 12.18]; μ = 5.63; σ = 3.32	[0.84 – 17.67]; μ = 4.51; σ = 2.36
Max von Mises Stress *(psi)*	[32,582 – 74,640]; μ = 51,937; σ = 14,659.7	[29,008 – 127,998]; μ = 53,759; σ = 16,167.53

The representativeness of the final 10 solutions to the breadth of the original sample space properties is further depicted in Figure 31, which shows the spread across the five properties for the original sample space and for the final 10. As can be seen in this figure, the final 10 downselected solutions range across the five properties such that there are no groupings of solutions within any region of the plots.

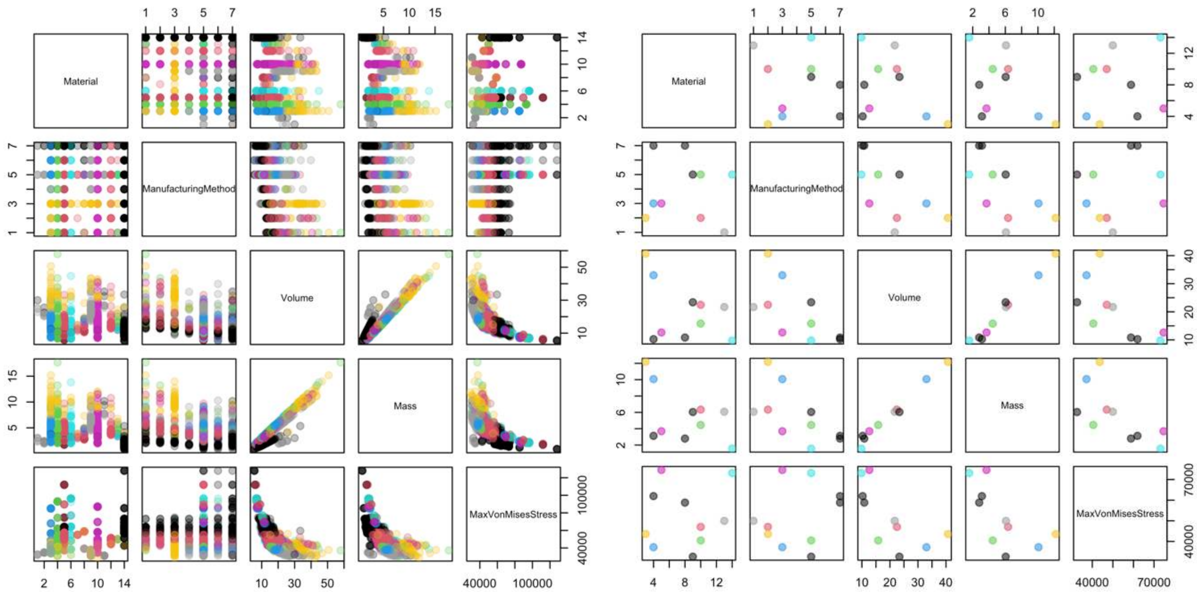


Figure 31: Original Solution Space (left) and final Solution Space (right), where Color Represents Cluster Group

5.5 Case Study Conclusions

The objective of this case study was to validate how the proposed clustering method of partitioning around medoids via the Gower distance matrix is able to systematically reduce a generative design system solution space with a variety of property types to a handful of unique and representative design alternatives. Based off the results of this case study, the Gower distance matrix with the PAM clustering method is a valid systematic approach to generative design solution space reduction. This approach successfully parsed a generative design solution space initially consisting of 1,000 aircraft engine loading bracket designs to smaller subsets (i.e., 500, 250, 100, 50, 25, and 10), while retaining the originality associated with different regions of the design space. As was summarized in Table 5, the attributes of the final 10 design alternatives are different from each other, while retaining the range for each property from the initial design space. Although there was a finite quantity of variables in this study (i.e., five attributes), the same methodology (i.e., Gower distance matrix and PAM) can be leveraged with a greater quantity of design variables due to the inherent structure of the clustering method since it does not have a quantitative limit of variables allowed. Therefore, this method can be transferred to other design problems that have multiple design alternatives comprised of a mix of categorical and numerical data. In addition, generative design systems provide design alternatives that are validated against the initial inputs provided by the designer, so any quantity of inputs a designer chooses to incorporate can be leveraged via the proposed method.

CHAPTER 6: GENERATIVE DESIGN STUDY RESULTS AND ANALYSIS

6.1 Introduction

Generative design systems aim to enhance the creativity of the designer by exploring search spaces in an innovative and efficient way to produce novel solutions (Bentley & Corne, 2002). The overall design process focuses on the emergence of shape and form in response to objectives (e.g., requirements) (Oxman, 2002). Generative design systems provide the greatest benefits during the conceptual design phase of the systems engineering life cycle by creating valid solutions that meet a set of requirements that encompass mixed variable types (e.g., categorical, continuous, etc.).

However, given the ambiguous nature of requirements during the conceptual design phase, generative design often produces thousands of design alternatives, which inherently places a significant cognitive workload on the user (McKnight, 2017). Due to the limitations of human cognitive ability, the user is only able to evaluate a limited number of design solutions without cognitive fatigue (Bentley & Corne, 2002). As a result of increased workload due to the evaluation of myriad generative design alternatives, users are subject to decision fatigue, which describes a phenomenon in which the limited reserve of stamina for making decisions becomes drained, which leads to poor self-control and impaired judgement (Polman & Vohs, 2016). Decision fatigue can further impair user ability to make trade-offs due to pursuing a passive role in the decision-making process, leading to choices that seem impulsive or irrational (Pignatiello et al., 2020).

The need to determine an optimal quantity of novel generative design alternatives would also help eliminate the complexity associated with the paradox of choice. Where the paradox of choice is a phenomenon in which the result of too many choices or a lack of meaningful choices leaves the human less happy, less satisfied, and occasionally paralyzed (Piasecki & Hanna, 2011). Left unmitigated, these factors lead to an inefficient systems engineering process with a negative user experience and improper design selection. Therefore, there is a need to identify parsing methods for a generative design solution space that indicate quantitative and qualitative improvements in user efficiency.

Given these inherent deficiencies in the generative design process, it becomes imperative that generative design systems leverage parsing methods that methodologically reduce the quantity of design options that are presented to the user, while retaining novel designs from distinct solution space regions. However, since the user would only be presented with a subset of the total number of generative design alternatives, it is crucial that the design alternatives be widely dispersed within the viable generative design solution space, such that each generated instance may be taken to represent a region of design possibilities (Krish, 2011).

Specifically, a clustering method using partitioning around medoids (PAM) has proven to be an effective parsing tool given increased robustness to noise and outliers in comparison to other clustering methods (Bhat, 2014). Since a generative design solution space encompasses all possible solutions given a set of requirements, outliers at extremes of the design envelope are possible. Therefore, the PAM approach can still identify accurate data relationships by minimizing the impact outliers have on the data analysis. Furthermore, PAM produces clusters that are dependent on similarity, thereby facilitating the detection of patterns from a large dataset (Ali et al., 2021). Since generative design variables are mixed, the ability to produce clustered subsets based on parameter similarity is crucial in the reduction of a generative design solution space to emphasize the uniqueness of individual clusters. Previous literature has used PAM clustering to reduce generative design solution spaces (Botyarov & Miller, 2022). This current study builds on previous PAM clustering literature, by evaluating usability from the designer's perspective of down selecting a generative design solution space parsed using these clustering algorithms.

Although parsing a solution space can yield smaller subsets of unique design alternatives, it is also imperative to consider how the subsets are presented to the user. Generative design systems are leveraged to produce novel design alternatives to create a product (e.g., furniture, aircraft loading bracket, etc.). The design process itself constructs a description of a product that satisfies a functional specification, meets certain performance criteria and resource limitations, is realizable, and satisfies criteria such as simplicity, testability, manufacturability, and reusability (Chandrasegaran et al., 2013). Generative design systems create design alternatives based on a set of these requirements, which have corresponding variables, such as material type, manufacturing method, volume range, mass range, and so forth. Therefore, it is logical to consider potential efficiencies in presenting a parsed generative design solution space based off a single or multi-variable filter. Performing design alternative tradeoffs hinges on the user's ability to visualize sub-shapes (e.g., components) within a design that can be modified. Recognizing emergent sub-shapes is a natural capacity of the human designer (Oxman, 2002). Therefore, variable filters that are visually discernable to a user, such as mass or manufacturing method, may further improve process efficiency than simply reviewing a subset of a generative design solution space by reducing workload associated with the selection of a design alternative since a user can discern design differences easier.

Currently, there exists a gap in the literature and in practice regarding generative design solution space parsing and human factors considerations of decision fatigue and usability. For a system to be usable and accessible, individuals should be able to use it to achieve their goals in an acceptable amount of time and be satisfied with the results (Bevan, 2008). However, many generative design tools and processes place significant burden on the user by overwhelming them with design alternatives.

6.2 Study Research Hypotheses

The overall objective of this study is to evaluate the user performance, workload, and experience for a generative design solution space selection process, where assessment considers both the quantity and filtering of alternatives presented using cluster analysis. In this study, we evaluated the following hypotheses:

- H_1: subset of the generative design solution space, parsed with a unique variable filter, will allow the user to select a single design more effectively (quickly, accurately) than if the entire solution space was presented with no variable filter available.
- H_2: a parsed subset of all generated design solutions will result in decreased cognitive overload when compared to an entire generated design set. This is provided that the parsed design options are significantly different, thereby originating from multiple unique regions of the design space.
- H_3: a parsed subset of all generated design solutions will result in increased perceived user performance when compared to an entire generated set. This is provided that the parsed design alternatives are significantly different, thereby originating from multiple unique regions of the design space.

6.3 Data Modeling

To account for the variability associated with each participant, such as priming effects and decision fatigue due to participants completing various instances of the study several times, generalized linear mixed models (GLMMs) were used to fit the data. Mixed models are used to analyze non-independent grouped data and are comprised of fixed and random effects, where fixed effects are model elements with a finite quantity of levels (e.g., iterations, filter types) and random effects have a theoretically infinite number of levels (e.g., participant variability) (Zuur, 2016). In this study, measurements from the same participant are likely to be more similar than measurements from different participants, and participants from the same study grouping are likely to be more similar than participants from different groupings. Therefore, it was determined that GLMMs would be appropriate for analyzing the data.

6.4 Data Analysis

Data cleaning and analysis was conducted using the R statistical software program (version 4.3.0). Statistical significance was assessed at $\alpha = 0.05$. Generalized linear mixed models (GLMMs) were used to evaluate the data. Linear mixed models (i.e., identify link function) were used to predict task duration, DRT response time, TLX workload, satisfaction, and confidence. GLMMs using the logit link function were fit on the binary outcome variables for correct selection of design alterative and DRT hit/miss. In each model, the fixed effects were filter type and iteration number, and participant number was treated as a random effect to account for the random variation across individuals.

6.5 Time to Complete Tasks (Task Duration)

Two linear mixed models were fit, one for design selection and one for visual dissimilarity, on time to complete task, see Table 6. For the design selection task, there was no significant difference between filter types, although it is noted that clustering on all variables of a design alternative (i.e., parameter similarity), yielded the quickest completion time with an average time to select a design alternative of 182.88 seconds compared to the next quickest average completion time of 206.16 for the mass filter and 208.47 seconds for the volume filter. All other filters had a longer average time for design selection completion than the control (i.e., no filter). It is also worth noting that mass and volume are variables that directly impact the shape of a design. Therefore, it is plausible that variables that influence visual characteristics of a design will have the greatest impact on user cognition since fluctuation in these variables are universally easier to discern, especially for users that lack familiarity with other generative design variables with less discernable impact on shape (e.g., material, stress, etc.). An increased sample size may further emphasize these nuances and produce greater statistical significance.

For the visual dissimilarity task, there was no statistically significant advantage in discerning visual differences between design alternatives given not clustering (i.e., no filter) when compared to clustering by all variables (i.e., parameter similarity). However, task completion times significantly increased when clustering was conducted on only one variable (e.g., material, stress, etc.). Specifically, clustering on all variables (i.e., parameter similarity) resulted in an average task completion time of 14.00 seconds, compared to an average completion time ranging from 15.88 seconds clustering by the manufacturing method variable to 24.10 seconds clustering by the volume variable.

In both the design selection and visual dissimilarity tasks, task durations tended to decrease significantly from iteration one to each subsequent iteration, with iteration seven resulting in the quickest average completion times (i.e., 60.74 seconds for design selection and 2.42 seconds for visual dissimilarity). This was observed more prominently in the design selection task as compared to the visual dissimilarity task. This indicates that the more a generative design solution space is reduced, meaning the smaller quantity of design alternatives per region are presented, the quicker a design selection decision can be made. A caveat to consider is design selection satisfaction, discussed in subsequent sections, meaning although a quicker design selection response time is achieved, the confidence and satisfaction in the selection could be lower in iteration seven than previous iterations given the lower quantity of design alternatives. However, the data in Table 6 indicates that design selections are made significantly quicker when less than 100 design alternatives are presented (i.e., average of 75.75 seconds when 100 alternatives are presented, 74.07 seconds with 50 design alternatives, 80.25 seconds for 25 alternatives, and 60.74 seconds with 10 alternatives) compared to an average of 228.08 seconds when 1,000 design alternatives are presented.

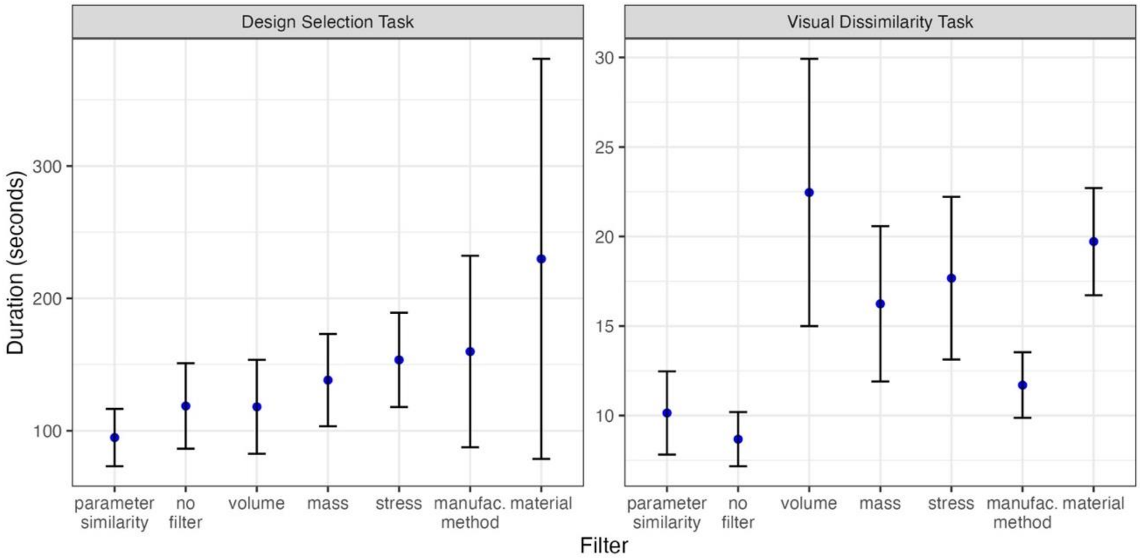
Moreover, a lower time to complete the visual dissimilarity task from iterations four to seven, when compared to iteration one, indicates that the design alternatives presented were substantially more visually different. When compared to an average completion time of 9.79 seconds in iteration one, average completion times were 5.14 seconds for iteration four, 7.91 seconds for iteration five, 6.19 seconds for iteration six, and 2.42 seconds for iteration seven. This indicates that a reduction in the design solution space results in more distinct clusters of solutions that are more visually different, specifically when the design solution space has 100 (i.e., iteration four) or less design alternatives.

Table 6: Model Summary Predicting Duration (seconds) for Design Selection Task (left) and Visual Dissimilarity Task (right)

Variable	**LMM on Design Selection**				**LMM on Visual Dissimilarity**			
	Coeff	**SE**	**t**	**p**	**Coeff**	**SE**	**t**	**p**
(Intercept)	228.08	56.23	4.06	< .001	9.79	3.08	3.18	.002
Filter (reference: no filter)								
Parm. Sim.	-45.20	63.82	-0.71	ns	4.21	3.27	1.29	ns
Manf. Met.	28.69	58.14	0.49	ns	6.09	2.95	2.06	.040
Material	99.21	60.23	1.65	ns	11.25	3.08	3.65	< .001
Stress	12.15	60.85	0.20	ns	10.06	2.97	3.38	< .001
Mass	-21.92	63.65	-0.34	ns	6.78	3.21	2.12	.035
Volume	-19.61	61.45	-0.32	ns	14.31	3.03	4.72	< .001
Iteration (reference: 1)								
2	10.65	50.22	0.21	ns	2.50	2.30	1.08	ns
3	-79.20	50.41	-1.57	ns	-1.25	2.31	-0.54	ns
4	-152.33	50.56	-3.01	.003	-4.65	2.31	-2.01	.044
5	-154.01	51.82	-2.97	.003	-1.88	2.32	-0.81	ns
6	-147.83	52.41	-2.82	.005	-3.60	2.34	-1.54	ns
7	-167.34	53.25	-3.14	.002	-7.37	2.34	-3.15	.002
Model Fit	*AIC*	*LL*	*Lratio*	*p*	*AIC*	*LL*	*Lratio*	*p*
Model	8109.1	-4039.6	147.8	< .001	5582.9	-2776.5	94.3	< .001
Null	8232.9	-4113.4			5652.9	-2823.5		

$N_{\text{design sel.}}$ = 566; $N_{\text{visual dis.}}$ = 658; AIC = Akaike Information Criterion; LL = log-likelihood; Lratio = likelihood ratio

The mean time to completion for each task by filter type is provided in Figure 32, which aggregates across all iterations for visual clarity. As shown in the figure, organized by parameter similarity (i.e., clustering on all variables) and no filter (i.e., clustering on no variables) resulted in quicker task completion than clustering on just one variable for both the design selection task and visual dissimilarity task.



Mean with 95% CI for Task Duration by Filter

6.6 Accuracy of Task Responses

A binary logistic regression was conducted predicting correct selection of design alternative for each the design selection task and the visual dissimilarity task, see Table 7. For the design selection task, the p-values ($p > .05$) of all filters, aside from volume, suggest that there is no statistical correlation between the impact of filter and the likelihood of a successful design selection. However, the positive coefficient for the volume filter variable indicates that it is an important factor for predicting a successful design selection. Specifically, filtering by volume increases the likelihood of a successful design selection by 352.7% (i.e., $e^{\beta} - 1$). Although not statistically significant, all other filters aside from manufacturing method had a positive coefficient. As such, the increased likelihood of a successful design selection filtering by parameter similarity could be 2.0%, material could be 107.5%, stress could be 93.5%, and mass could be 27.1%. An increased sample size would likely further validate the importance of filtering on other filters, not solely volume, for improving the likelihood a successful design selection when compared to no filter.

Further, for the design selection task, iteration three, iteration four, iteration six, and iteration seven are significant factors for predicting a successful design selection. Since these statistically significant iteration coefficients are positive, parsing a design solution space by presenting fewer design alternatives increases the probability of a successful design selection when compared to no filter. Specifically, the likelihood of a successful design selection when 250 design alternatives are presented (i.e., iteration three) increases by 425.9%. When 100 design alternatives are presented (i.e., iteration four), the likelihood of a successful selection increases by 380.7%. When 25 alternatives are presented (i.e., iteration four), the likelihood of a successful selection increases by 646.3%. When 10 alternatives are presented (i.e., iteration four), the likelihood of a successful selection increases by 969.7%.

For the visual dissimilarity task, the binary logistic regression predicts that when presented with four design alternative images the user visually identified the design that was from a different cluster. Given $p > .05$ for all filter types, there is not statistical significance between the filter and the impact it has on the likelihood of visually identifying the design alternative from a different cluster (e.g., intra- versus inter-cluster differences). Additionally, all iterations aside from iteration five, had $p < .05$, indicating that iteration (i.e., design solution space size) is an important factor in predicting the likelihood of the user visually identifying the design that was from a different cluster. Therefore, iteration two is associated with 74.6% decrease in the likelihood of a correct design selection compared to iteration one; iteration three is 79.0%, iteration four is 67.7%, iteration six is 58.1%, and iteration seven is 72.8%. This suggests that although less design alternatives are presented in subsequent iterations, than iteration one (1,000), intra-cluster differences increase such that is becomes more difficult to visually discern which design is from a unique cluster.

Table 7: Model Summary Predicting Correct Selection of Design Alternative for Design Selection Task (left) and Visual Dissimilarity Task (right)

Variable	**Binary Logit – Design Selection**				**Binary Logit – Visual Dissimilarity**			
	Coeff	**SE**	**t**	**p**	**Coeff**	**SE**	**t**	**p**
(Intercept)	-3.19	0.56	-5.66	< .001	2.28	0.44	5.18	< .001
Filter (reference: no filter)								
Parm. Sim.	0.02	0.52	0.04	ns	-0.25	0.39	-0.66	ns
Manf. Met.	-0.25	0.50	-0.50	ns	-0.16	0.37	-0.44	ns
Material	0.73	0.47	1.56	ns	-0.11	0.38	-0.28	ns
Stress	0.66	0.48	1.36	ns	0.49	0.41	1.18	ns
Mass	0.24	0.51	0.47	ns	-0.16	0.39	-0.40	ns
Volume	1.51	0.47	3.18	0.001	0.04	0.40	0.09	ns
Iteration (reference: 1)								
2	-17.99	3205.16	-0.01	ns	-1.37	0.41	-3.38	< .001
3	1.66	0.50	3.28	0.001	-1.56	0.40	-3.87	< .001
4	1.57	0.51	3.09	0.002	-1.13	0.41	-2.73	.006
5	-0.60	0.74	-0.81	ns	-0.77	0.43	-1.79	ns
6	2.01	0.51	3.97	< .001	-0.87	0.43	-2.04	.041
7	2.37	0.51	4.70	< .001	-1.30	0.41	-3.16	.002
Model Fit	*AIC*	*LL*	*Lratio*	*p*	*AIC*	*LL*	*Lratio*	*p*
Model	450.6	-211.3	112.2	< .001	724.5	-348.2	27.0	.008
Null	538.8	- 267.4			727.5	-361.7		

$N_{design\ sel.}$ = 566; $N_{visual\ dis.}$ = XX; AIC = Akaike Information Criterion; LL = log-likelihood; Lratio = likelihood ratio; link function = logit

The total percent correct for each filter by task is shown in Figure 33. It is noteworthy that for the visual dissimilarity task, the accuracy of selecting the most visually different design alternative was over 70% for each filter, indicating that there is a discernible visual difference between filter clusters. This means that clustering based on various design variables leads to visually discernible differences between the clusters, indicating the clustering method leveraged parses the generative design solution space into clusters that vary visually.

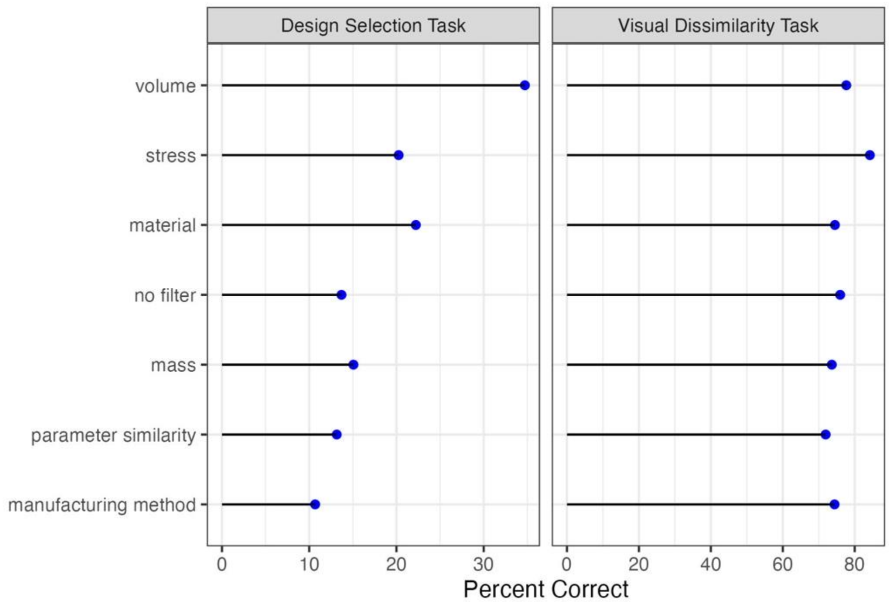


Figure 32: Total Proportion of Correct Selections of Design Alternative by Filter

6.7 Detection Response Task (DRT) Performance

The detection response task yielded a measure for response time (modeled using a linear mixed model) and hit/miss (modeled using a binary logit model), see Table 8. DRT response times for any filter, aside from the mass filter, ranged from 2.10 to 2.51 seconds, which indicated a quicker response time than 2.79 seconds if no filter was applied. This indicates that filtering design alternatives by a variable would result in slightly decreased cognitive workload given the quicker response times to the DRT.

Distraction response times, per iteration, do not indicate a statistically significant variation in time to respond to the distraction. The data collected indicates that although the quantity of design alternatives presented to the participants decreased as iterations progressed, the cognitive workload associated with evaluating design alternatives remained approximately the same, ranging from quickest average DRT response times in iteration three of 2.63 seconds to slowest average DRT response times in iteration seven 2.89 seconds.

For the binary logistic regression, the hit target variable is a successful (yes/no) response to the DRT (i.e., click red box) within 10 seconds or less of DRT appearing. Independent variables include filter category and iteration. Given $p < .05$ of all filters but mass, the filter variables are important factors for predicting a successful hit or not. Since statistically significant filter coefficients are positive, filtering a design solution space by a variable will increase the probability of a successful DRT hit when compared to no filter, implying a reduced cognitive workload. Specifically, filtering by parameter similarity increases the likelihood of a DRT response in 10 seconds or less (i.e., successful hit) by 32.3% (i.e., $e^{\beta} - 1$). Following the same analysis process, the increased likelihood of a successful hit filtering by manufacturing method is 141.1%, material is 82.2%, stress is 150.9%, and volume is 228.7%.

Since $p < .05$ for iteration five, six, and seven, these iteration variables are important factors for predicting a successful hit or not. Successful hits, meaning response to the DRT within 10 seconds or less, have a statistically significant negative coefficients, iteration five is associated with a 13.9% decrease in the likelihood of a successful hit compared to iteration one; the decreased likelihood of a successful hit during iteration six is 17.3% and iteration seven is also 17.3%. This suggests that although less design alternatives are presented in iteration five (50), iteration six (25), and iteration seven (10), than iteration one (1,000), users may have an increased cognitive workload due to limited design alternative options, resulting in an increased likelihood of an unsuccessful DRT response.

Table 8: Model Summary Predicting Response Time (seconds) of Hits (left) and Hit (right) for Detection Response Task

Variable	LMM for Response Time				Binary Logit for Hit			
	Coeff	**SE**	**t**	**p**	**Coeff**	**SE**	**t**	**p**
(Intercept)	2.79	0.08	33.34	< .001	0.04	0.08	0.44	ns
Filter (reference: no filter)								
Parm. Sim.	-0.28	0.08	-3.64	< .001	0.28	0.07	4.18	< .001
Manf. Met.	-0.69	0.07	-9.32	< .001	0.88	0.07	12.56	< .001
Material	-0.40	0.08	-4.80	< .001	0.60	0.08	7.77	< .001
Stress	-0.40	0.08	-5.23	< .001	0.92	0.07	12.50	< .001
Mass	0.16	0.10	1.54	ns	-0.07	0.09	-0.84	ns
Volume	-0.41	0.07	-5.81	< .001	1.19	0.07	17.24	< .001
Iteration (reference: 1)								
2	-0.03	0.07	-0.45	ns	0.01	0.08	0.09	ns
3	-0.16	0.07	-2.27	0.023	0.10	0.08	1.21	ns
4	0.01	0.07	0.09	ns	-0.02	0.08	-0.32	ns
5	0.07	0.07	0.95	ns	-0.15	0.08	-1.89	0.058
6	0.08	0.07	1.16	ns	-0.19	0.08	-2.50	0.012
7	0.10	0.07	1.39	ns	-0.19	0.08	-2.56	0.01
Model Fit	*AIC*	*LL*	*Lratio*	*p*	*ResDF*	*ResDev*	*Dev*	*p*
Model	32385	-16178	14.7	< .001	13213	16618	-591.7	< .001
Null	32535	-16265			13225	17210		

N_{RT} = 8,528; N_{Hit} = 13,226; AIC = Akaike Information Criterion; LL = log-likelihood; Lratio = likelihood ratio; ResDF = Residual Degrees of Freedom; ResDev = Residual Deviance; Dev = Deviance

A comparison of the mean DRT response times by filter is provided in Figure 34, where no filter and mass filter have noticeably higher response times. Given that the mass data is not statistically significant, from Table 8, it can be concluded that filtering a generative design solution space will result in improved DRT response times when compared to no filter.

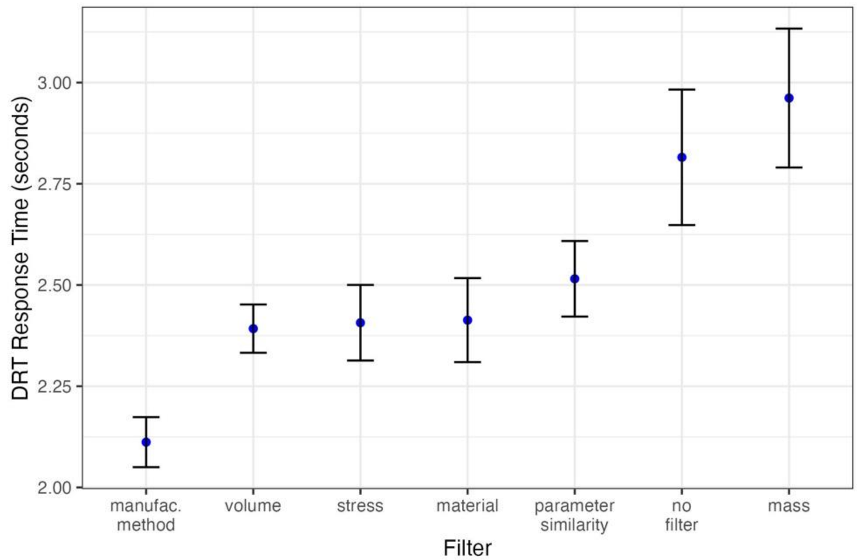


Figure 33: Mean with 95% CI for Response Time of Hits to Detection Response Task, Where Higher Values Represent Higher Cognitive Workload

The mean hit rate, as shown in Figure 35, for the detection response task indicates that for most generative design solution space filters, aside from mass, the hit count was over 50%. This indicates that if generative design solutions are logically parsed by a variable (e.g., volume, material, etc.), then cognitive workload is decreased, thereby enabling a more efficient design selection process. The variability of the mass filter can be attributed to a small sample size, meaning analysis of a larger sample size could lead to more accurate results in which the mass alternative has a higher hit rate than the no filter alternative.

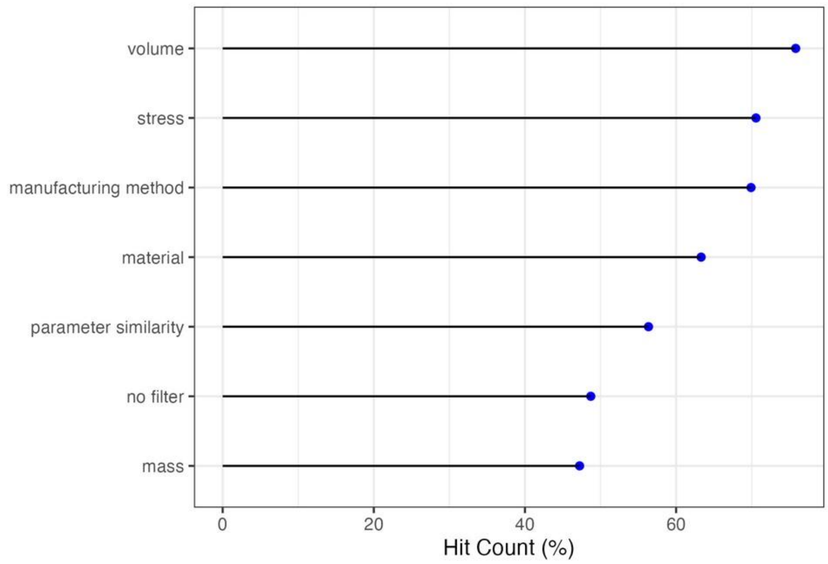


Figure 34: Mean Hit Rate for Detection Response Task, Where Lower Values Represent Higher Cognitive Workload

6.8 Perceived Workload

Table 9 summarizes the results of two linear mixed models fit on participant NASA TLX scores for each task. The design selection task perceived workload, per clustering filter, indicates a somewhat high cognitive workload (37.06 – 42.17) for participants across all filters. However, the parameter similarity filter indicates the lowest TLX score (37.06) compared to any other filter. This suggests that parsing a generative design solution space by clustering on all design variables would result in a reduced cognitive workload. The filters with the next lowest TLX scores are mass (37.99) and volume (38.06). Interestingly, perceived workload results correlate with time to complete task results in Table 6, where the quickest average time to select a design alternative was achieved with the visual similarity filter at 182.88 seconds and the subsequent quickest average times to select a design alternative where the mass filter and volume filter, at 206.16 seconds and 208.47 seconds, respectively. This further provides evidence that the previous claim stating variables that influence visual characteristics of a design will have the greatest impact on user cognition since fluctuation in the values of these variables are universally easier to discern.

The design selection task perceived workload, per iteration, indicates a steady decrease in cognitive workload from iteration three to iteration seven. Specifically, TLX scores decrease from 42.04 (iteration one) to 38.61 (iteration three) to 36.54 (iteration four) to 36.38 (iteration five) to 35.86 (iteration six) and 33.69 (iteration seven). This decrease in cognitive workload, correlated to fewer design alternatives in the generative design solution space, brings the design selection task TLX score closer to the minimum cognitive workload for visual search activities from literature at 28.98 (minimum). Overall, the cognitive workload decreased from 42.91 in iteration two to 33.69 in iteration seven. This suggests that the quantity of generative design solutions has a greater impact on improving cognitive workload than solely a filter (e.g., lowest iteration TLX score of 33.69 compared to lowest filter [parameter similarity] TLX score of 37.06). Therefore, it can be inferred that cognitive workload for a design selection task would be lowest with a combination of parsing the generative design solution space via parameter similarity (e.g., clustering considering all variables) and presenting the designer with 10 design alternatives since both resulted in the lowest TLX scores of 37.06 and 33.69, respectively.

The visual dissimilarity task perceived workload, per clustering filter, does not yield any statistically significant results. The TLX scores ranged from 25.13 to 30.36, indicating medium to somewhat high cognitive workload. However, the mass filter resulted in the lowest cognitive workload with a TLX score of 25.13. This indicates that physical variables of a design that influence a design alternative’s shape, such as mass, could be easier to discern visually than other characteristics, such as material (e.g., TLX score of 30.36) or manufacturing method (e.g., TLX score of 29.46), especially if the designer is not familiar with the nuances of the latter.

The visual dissimilarity task perceived workload, per iteration, TLX scores range from 27.25 to 32.07, indicating medium to somewhat high cognitive workload. When compared to iteration one TLX score of 27.25, the data collected indicates that perceived cognitive workload increased from medium to somewhat high in iteration five to 30.07, in iteration six to 32.07, and in iteration seven to 29.94. This could be attributed to the fact that as the generative design solution space decreased, the variability of design within the same cluster grouping also increased due to fewer design alternatives to form a cluster, causing additional effort to be placed on identifying visual similarities and differences intra- and inter-cluster. This could imply that design alternatives within a cluster grouping are more visually similar the larger the generative design space is, yielding more efficient parsing results with a larger solution space.

Table 9: Model Summary Predicting TLX Workload for Design Selection Task (left) and Visual Dissimilarity Task (right)

Variable	**LMM on Design Selection**				**LMM on Visual Dissimilarity**			
	Coeff	**SE**	**t**	**p**	**Coeff**	**SE**	**t**	**p**
(Intercept)	42.04	2.23	18.84	< .001	27.25	2.40	11.36	< .001
Filter (reference: no filter)								
Parm. Sim.	-4.98	2.07	-2.41	0.016	2.22	2.08	1.07	ns
Manf. Met.	-0.01	1.87	0.00	ns	2.21	1.88	1.18	ns
Material	0.13	1.93	0.07	ns	3.11	1.95	1.60	ns
Stress	-2.29	1.90	-1.20	ns	-0.81	1.86	-0.44	ns
Mass	-4.05	2.08	-1.94	0.053	-2.12	2.06	-1.03	ns
Volume	-3.98	1.88	-2.12	0.034	1.68	1.89	0.89	ns
Iteration (reference: 1)								
2	0.87	1.38	0.63	ns	0.91	1.40	0.65	ns
3	-3.43	1.39	-2.46	0.014	1.71	1.40	1.22	ns
4	-5.50	1.40	-3.94	< .001	2.32	1.40	1.65	ns
5	-5.66	1.44	-3.94	< .001	2.82	1.41	2.00	.046
6	-6.18	1.45	-4.26	< .001	4.82	1.42	3.40	< .001
7	-8.35	1.48	-5.66	< .001	2.69	1.42	1.90	.058
Model Fit	*AIC*	*LL*	*Lratio*	*p*	*AIC*	*LL*	*Lratio*	*p*
Model	4221.9	-2095.9	105.8	< .001	4932.1	-2451.1	53.9	< .001
Null	4303.7	-2148.8			4962.1	-2478.0		

$N_{design\ sel.}$ = 565; $N_{visual\ dis.}$ = 651; AIC = Akaike Information Criterion; LL = log-likelihood; Lratio = likelihood ratio

These mean scores for each filter are further shown in Figure 36, where similar trends in TLX scores were observed across the filters in both tasks in Table 9. Specifically, the mean of parameter similarity TLX scores for the design selection task indicate a medium cognitive workload, with a mean of less than 30. This contrasts with the remaining filters for the design selection task that indicate a mean TLX score of greater than 30, which correlates to a somewhat high cognitive workload. The mean TLX scores for the visual dissimilarity task for the parameter similarity, material, volume, and mass filters indicate a medium cognitive workload, with a mean of less than 30. This further provides evidence that variables that influence visual characteristics of a design, such as volume and mass, will have the greatest impact on user cognition since fluctuation in the values of these variables are universally easier to discern. In this case, this results in lower TLX scores, which correlate to a lower cognitive workload.

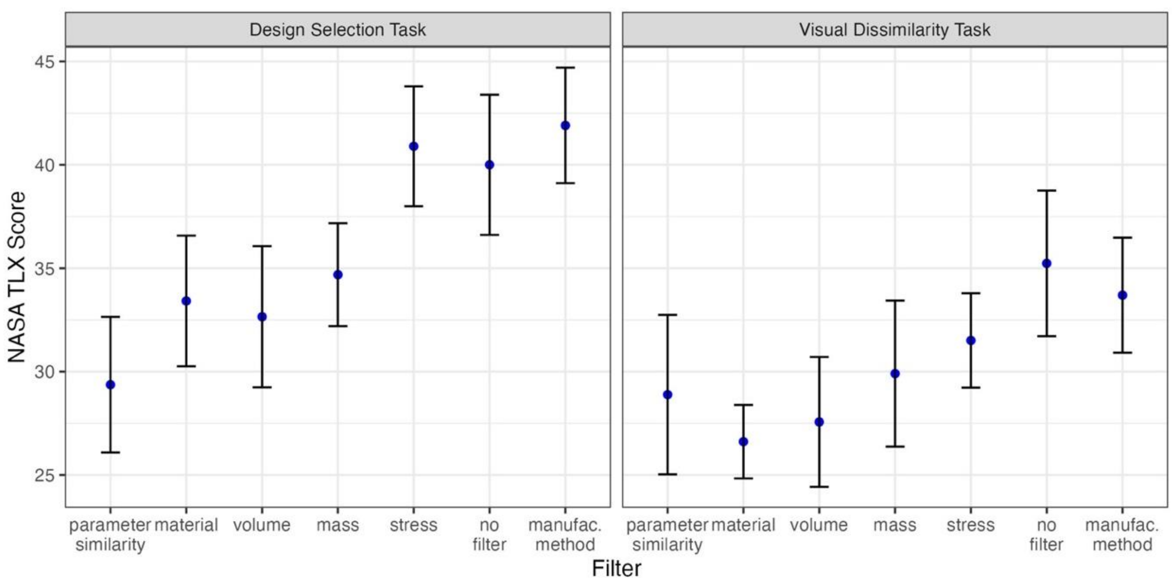


Figure 35: Mean with 95% CI for Task TLX Workload by Filter

6.9 Confidence and Satisfaction in Design Selection Task Outcome

The last set of linear mixed models fit on the data was one for self-reported confidence and self-reported satisfaction in the design selection task outcome, see Table 10. The design selection task confidence, per clustering filter, does not yield any statistically significant results. The collected data indicates that participant confidence in selecting a design alternative was high (e.g., rated a four on a scale of one [not confident] to five [extremely confident]). Confidence scores per clustering filter ranged from a low of 3.89, for the manufacturing method filter, to a high of 4.16 for the volume filter.

The design selection task confidence, per clustering iteration, indicates that participant perceived confidence in their selected design alternative remained consistent between iteration one and five, ranging from 4.02 to 4.09, but decreased to 3.7 for iteration six and 3.4 for iteration seven. This indicates that participants were unconfident with their design alternative selection with limited design alternatives, specifically when the generative design solution space consisted of 25 design alternatives in iteration six and 10 design alternatives in iteration seven.

The design selection task satisfaction, per clustering filter, indicates that participant perceived satisfaction with the presented generative design alternatives was higher for filters than the no filter alternative. Specifically, with no filter the satisfaction score was 3.46, whereas the satisfaction scores for any other filter ranged from 3.46, for manufacturing method filter, to 3.94, for the stress filter. This indicates that presenting parsed data, by clustering on design variables, increases the perceived satisfaction of participants.

The design selection task satisfaction, per clustering iteration, indicates that participant perceived satisfaction with the presented generative design alternatives remained consistent between iteration one and five, ranging from 3.46 to 3.63, but decreased to 3.28 for iteration six and 3.05 for iteration seven. This indicates that participants were unsatisfied with the design alternatives presented, specifically with a reduced generative design solution space that consisted of 25 design alternatives in iteration six and 10 design alternatives in iteration seven.

Table 10: Model Summary Predicting Confidence (left) and Satisfaction (right) for Design Selection Task

Variable	**LMM on Confidence**				**LMM on Satisfaction**			
	Coeff	**SE**	**t**	**p**	**Coeff**	**SE**	**t**	**p**
(Intercept)	4.02	0.16	24.39	< .001	3.46	0.15	22.42	< .001
Filter (reference: no filter)								
Parm. Sim.	-0.03	0.17	-0.20	ns	0.11	0.17	0.69	ns
Manf. Met.	-0.13	0.15	-0.88	ns	0.00	0.15	-0.01	ns
Material	0.12	0.16	0.74	ns	0.47	0.15	3.06	.002
Stress	0.07	0.16	0.45	ns	0.48	0.15	3.17	.002
Mass	0.03	0.17	0.16	ns	0.10	0.16	0.59	ns
Volume	0.14	0.15	0.93	ns	0.33	0.15	2.18	.030
Iteration (reference: 1)								
2	0.07	0.11	0.59	ns	0.05	0.11	0.40	ns
3	0.05	0.12	0.47	ns	0.17	0.11	1.44	ns
4	0.06	0.12	0.56	ns	0.15	0.11	1.31	ns
5	0.04	0.12	0.35	ns	0.11	0.12	0.98	ns
6	-0.32	0.12	-2.70	.007	-0.18	0.12	-1.49	ns
7	-0.62	0.12	-5.05	< .001	-0.41	0.12	-3.41	< .001
Model Fit	*AIC*	*LL*	*Lratio*	*p*	*AIC*	*LL*	*Lratio*	*p*
Model	1474.9	-722.5	26.2	.010	1455.8	-712.9	26.8	.008
Null	1477.1	-735.6			1458.6	-726.3		

$N_{confidence}$ = 573; $N_{satisfact.}$ = 573; AIC = Akaike Information Criterion; LL = log-likelihood; Lratio = likelihood ratio

Figure 37 specifically shows the mean response across participants for each filter type, indicating that mean confidence response scores were, on average, higher than the mean satisfaction response scores. The discrepancy between confidence and satisfaction could be attributed to lack of participant familiarity with generative design systems, specifically with the quality and quantity of design alternatives presented when compared to other design processes.

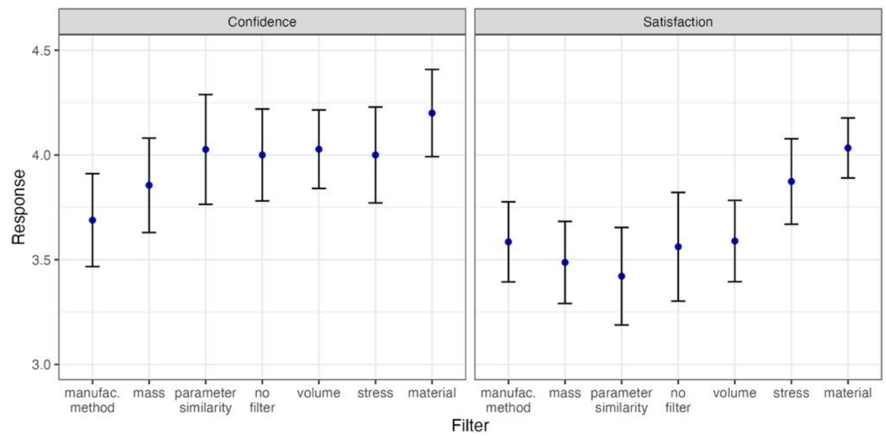


Figure 36: Mean with 95% CI for Self-Reported Confidence in Selection, and Satisfaction of the Design Space

6.10 Evaluation of Hypotheses

H1: subset of the generative design solution space, parsed with a unique variable filter, will allow the user to select a single design more effectively (quickly, accurately) than if the entire solution space was presented with no variable filter available. An evaluation of objective data (i.e., time and accuracy) indicated that clustering a generative design solution space on all variables associated with a design alternative (e.g., categorical [material, manufacturing method] and continuous variables [volume, mass, stress]) results in quicker design alternative selection when compared to clustering by only a single variable or no clustering at all. Furthermore, the data indicates that presenting a subset of the entire generative design solution space results in quicker design alternative selection, specifically the quantity of design alternatives presented is between 10 and 100. Correct selection of a design alternative, given unique design requirements, further improved as the generative design solution space was parsed to present fewer design alternatives, specifically between 10 and 250.

H2: a parsed subset of all generated design solutions will result in decreased cognitive overload when compared to an entire generated design set. This is provided that the parsed *design options are significantly different, thereby originating from multiple unique regions of the design space.* An evaluation of objective data (i.e., DRT response time and hits) indicated the time to discern a visual difference between generative design alternatives decreased as smaller clustered subsets of the generative design solution space were evaluated. Lower cognitive workload, correlated to quicker times to accurately discern a visual difference between alternatives, was greatest when the generative design solution space was parsed to include between 10 and 100 design alternatives. When clustering a generative design solution space, visual differences between design alternatives are easier to discern when clusters are created considering all design alternative variables than solely clustering on a single variable. This is further supported via distraction analysis, where designers exhibited a higher hit rate (e.g., responding to the distraction), indicating a lower cognitive workload, when the design alternatives were parsed considering all design variables than no parsing of the generative design solution space.

H3: a parsed subset of all generated design solutions will result in increased perceived user performance when compared to an entire generated set. This is provided that the parsed design alternatives are significantly different, thereby originating from multiple unique regions of the design space. An evaluation of subjective survey data (i.e., NASA TLX, confidence and satisfaction) was performed. The subjective NASA TLX survey data is consistent with the objective data in that clustering a generative design solution space considering all design alternative variables resulted in a lower perceived workload when compared to clustering by a single variable or no clustering at all. Perceived workload also decreased as the generative design solution space decreased, with the lowest perceived workload reported when the solution space consisted of 10 to 100 design alternatives. However, evaluation of subjective data associated with the ability to discern a design alternative that is most visually different along with confidence and satisfaction associated with a design decision, indicates that if the generative design solution space is parsed too much, specifically if few design alternatives are presented, then cognitive workload increases, and design selection confidence along with satisfaction with the design alternatives presented decreases. Cognitive workload increases since when 10 to 25 design alternatives are presented, they are all visually different, that it becomes difficult to discern which one is most different from the others. Furthermore, when 10 to 25 design alternatives are presented, designers feel constrained, meaning that creativity of the design process is stifled given the limited pool of design alternatives that confidence with a selected design and satisfaction with the entire design solution envelope are diminished. Generative design systems attempt to enhance the creativity of the designer by exploring search spaces in an innovative and efficient way to produce novel solutions (Bentley & Corne, 2002). Therefore, it is imperative that reduction of the design envelope does not constrain designer creativity by presented too limited a subset of design alternatives.

CHAPTER 7: CONTRIBUTIONS & FUTURE WORK

7.1 Research Contributions

It has been noted that the generative design domain is lacking framework for design evaluation given iterative processes are still not so common in literature and outcome selection frameworks are almost absent (Pilagatti et al., 2022). This study has provided a structured approach to produce an optimized parsing of spatially different generative design solutions, derived from generative design systems, such that human cognitive performance during the design process is improved. Given the evaluation of the presented hypotheses, it is recommended that parsing of a design solution space involves a clustering method in which all design variables, both categorical and continuous, are considered. Furthermore, it is recommended that a reduced subset of an overall design solution space is presented to designers such that cognitive workload is reduced. Objective data suggests that cognitive workload is lowest when 10 to 100 design alternatives are presented for evaluation in the subset of the overall design solution space. However, subjective data indicates a caution when limiting the subset of designs presented, since design selection confidence and satisfaction may be decreased the more limited the design alternative selection becomes. Given these subjective considerations, it is recommended that a design solution space consists of 50 to 100 design alternatives, with a clustering parsing method that considers all design alternative variables.

The results from this dissertation have been organized into three journal articles for disseminating the results. The articles are as follows:

- Botyarov, M., & Miller, E. (under review). Generative design solution space parsing: An evaluation of user experience, workload, and performance. *Journal of Systems Science and Systems Engineering*.
- Botyarov, M., & Miller, E.E. (2022). Partitioning around medoids as a systematic approach to generative design solution space reduction. *Results in Engineering*, 100544. doi.org/10.1016/j.rineng.2022.100544
- Botyarov, M., & Miller, E. (2021). Evaluating usability of generative design process for human-centered design. *International Journal of Development Research, 11*(3), pp 45148-45152.

7.2 Conclusions and Future Work

Although generative design systems must allow room for designer creativity, there is potential for artificial intelligence (AI) to further enhance the generative design process. AI has produced several powerful tools including knowledge-based systems, fuzzy logic, inductive learning, neural networks, and genetic algorithms (Pham & Pham, 1999). Genetic algorithms have been used in optimization applications across multiple domains (Singh & Gu, 2012). Specifically, the usage of genetic algorithms in generative design applications has become increasingly prevalent, both in the generation of design alternatives and user interfaces (Troiano & Birtolo, 2014). In addition to AI applications, generative design is expanding into the domain of machine learning (ML). Clustering is considered an unsupervised learning technique used in ML such that a system can learn automatically and continuously improve from self-experience without being explicitly programmed (Nti et al., 2022). ML in the context of engineering design ultimately focuses on developing autonomous systems that replicate human abilities and support human decision-making (Panchal et al., 2019). Therefore, in the generative design realm, ML assists with the creation and evaluation of design alternatives to facilitate human decision-making via the exploration of hybrid design spaces (Khan & Awan, 2018). An advantage of the Gower distance matrix and partitioning around medoids clustering method, used in this study, as an approach to parsing a generative design solution space, is the ability to consider both categorical and continuous variables as part of the parsing process of the design solution space, thereby providing a structured approach to parsing generative design system output. This method is transferrable across generative design systems since clustering could be integrated as part of an AI approach in creating an optimized design space for subsequent evaluation. Furthermore, application of this method should be researched outside of the generative design realm where clustering could be used as part of data analysis to discover relationships within a given data set. With big data becoming increasingly widespread in engineering applications, a comprehensive clustering method has the potential to uncover unique regions and relationships within a broad data set. Further research can be conducted into the efficiency of the specified clustering method given a varying quantity of variables than in this study and varying variable types. The quantity of variables can further be evaluated against cognitive workload in processing the parsed results following the executed clustering method.

Since generative design is intended for use in the conceptual design phase, it is crucial that the designer has sufficient cognitive ability to evaluate trade-offs that would result in a balanced design that satisfies requirements. This study focused on a method of analyzing generative design output, however further research should be conducted into the integration of human preference into the generative design process. Specifically, recognition and identification of aesthetic preference is indispensable in design (Chew et al., 2016). The digitalization associated with generative design is contributing to Industry 4.0, delivering real-time decision making, enhanced productivity, flexibility and agility (Ghobakhloo, 2020). Industry 4.0, also known as the Industrial Internet of Things (IIoT), yields extensive cross-domain opportunities via five distinct research areas, including human resource management, implementation, business models, supply chain management, and law and ethics (Kiel, 2020). Comprehensive methods from several Industry 4.0 research areas should be investigated in the context of generative design to further consider design aesthetics. Ongoing research is involved with establishing preference-based measurement of user aesthetics using electroencephalogram (EEG) signals for virtual 3D shapes with motion (Chew et al., 2016). In addition to analyzing brain signals for decision-making, eye-tracking is becoming one of the most commonly used sensor modalities in affective computing (Lim et al., 2020). In the context of generative design, eye-tracking allows the aesthetic exploration of vast numbers of design alternatives that could be refined based off lingering on a design thereby the generative design system would present similar options based on the intuitive reaction of the designer (Mountstephens & Teo, 2020). If such a use case was expanded on in the future, results from this research could provide the parsing methodology for a generative design system to extract similar design alternatives based off eye tracking, thereby further improving the efficiency of the human component of generative design systems.

REFERENCES

Adams, F., & Aizawa, K. (2001). The bounds of cognition. *Philosophical Psychology*, *14*(1). https://doi.org/10.1080/09515080120033571

Adams, K. M. G., & Keating, C. B. (2011). Overview of the systems of systems engineering methodology. *International Journal of System of Systems Engineering*, *2*(2–3). https://doi.org/10.1504/IJSSE.2011.040549

Agkathidis, A. (2015). Generative Design Methods. *Real Time: Extending the Reach of Computation, ECAADe 33*, *2*.

Ali, N. G., Abed, S. D., Shaban, F. A. J., Tongkachok, K., Ray, S., & Jaleel, R. A. (2021). Hybrid of K-Means and partitioning around medoids for predicting COVID-19 cases: Iraq case study. *Periodicals of Engineering and Natural Sciences*, *9*(4). https://doi.org/10.21533/pen.v9i4.2382

Allen, C. (2017). On (not) defining cognition. *Synthese*, *194*(11). https://doi.org/10.1007/s11229-017-1454-4

Anderson, C. (2001). Swarm Intelligence: From Natural to Artificial Systems . Eric Bonabeau , Marco Dorigo , Guy Theraulaz . *The Quarterly Review of Biology*, *76*(2). https://doi.org/10.1086/393972

Andreasen, M. M., Hansen, C. T., & Cash, P. (2015). Conceptual design: Interpretations, mindset and models. In *Conceptual Design: Interpretations, Mindset and Models*. https://doi.org/10.1007/978-3-319-19839-2

Batool, F., & Hennig, C. (2021). Clustering with the Average Silhouette Width. *Computational Statistics and Data Analysis*, *158*. https://doi.org/10.1016/j.csda.2021.107190

Bentley, P. J. (1999). From Coffee Tables to Hospitals: Generic Evolutionary Design. In *Evolutionary Design by Computers*.

Bentley, P. J., & Corne, D. W. (2002). An introduction to Creative Evolutionary Systems. In *Creative Evolutionary Systems*. https://doi.org/10.1016/b978-155860673-9/50035-5

Bevan, N. (2008). 【Meaningful Measures】-Classifying and selecting UX and usability measures. *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement (VUUM)*.

Bhat, A. (2014). K-Medoids Clustering Using Partitioning Around Medoids for Performing Face Recognition. *International Journal of Soft Computing, Mathematics and Control*, *3*(3). https://doi.org/10.14810/ijscmc.2014.3301

Bholowalia, P., & Kumar, A. (2014). EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications*, *105*(9).

Bisantz, A. M., Roth, E., Brickman, B., Gosbee, L. L., Hettinger, L., & McKinney, J. (2003). Integrating cognitive analyses in a large-scale system design process. *International Journal of Human Computer Studies*, *58*(2). https://doi.org/10.1016/S1071-5819(02)00130-1

Blanchard, B. S., & Fabrycky, W. (2014). Systems Engineering and Analysis Pearson New International Edition (5th Edition). In *Prentice-Hall International Series in Industrial and Systems Engineering*.

Blomberg, O. (2011). Conceptions of cognition for cognitive engineering. *International Journal of Aviation Psychology*, *21*(1). https://doi.org/10.1080/10508414.2011.537561

Blum, C., & Li, X. (2008). Swarm Intelligence in Optimization. In *Swarm Intelligence*. https://doi.org/10.1007/978-3-540-74089-6_2

Bolker, B. M., Brooks, M. E., Clark, C. J., Geange, S. W., Poulsen, J. R., Stevens, M. H. H., & White, J. S. S. (2009). Generalized linear mixed models: a practical guide for ecology and evolution. In *Trends in Ecology and Evolution* (Vol. 24, Issue 3). https://doi.org/10.1016/j.tree.2008.10.008

Botyarov, M., and Miller, E. (2021). Evaluating usability of generative design process for human-centered design. *International Journal of Development Research,* 11(3), 45148-45152.

Botyarov, M., & Miller, E. E. (2022). Partitioning around medoids as a systematic approach to generative design solution space reduction. *Results in Engineering*, *15*. https://doi.org/10.1016/j.rineng.2022.100544

Bridges, C. C. (1966). Hierarchical Cluster Analysis. *Psychological Reports*, *18*(3). https://doi.org/10.2466/pr0.1966.18.3.851

Brossard, M., Gatto, G., Gentile, A., Merle, T., & Wlezien, C. (2020). How generative design could reshape the future of product development. *Mckinsey Co*.

BuHamdan, S., Alwisy, A., & Bouferguene, A. (2021). Generative systems in the architecture, engineering and construction industry: A systematic review and analysis. In *International Journal of Architectural Computing* (Vol. 19, Issue 3). https://doi.org/10.1177/1478077120934126

Buonamici, F., Carfagni, M., Furferi, R., Volpe, Y., & Governi, L. (2020). Generative design: An explorative study. *Computer-Aided Design and Applications*, *18*(1). https://doi.org/10.14733/cadaps.2021.144-155

Cagan, J., Grossmann, I. E., & Hooker, J. (1997). A conceptual framework for combining artificial intelligence and optimization in engineering design. *Research in Engineering*

Design - Theory, Applications, and Concurrent Engineering, *9*(1). https://doi.org/10.1007/BF01607055

Caldas, L. (2006). GENE_ARCH: An evolution-based generative design system for sustainable architecture. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4200 LNAI*. https://doi.org/10.1007/11888598_12

Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., & Gao, W. (2013). The evolution, challenges, and future of knowledge representation in product design systems. *CAD Computer Aided Design*, *45*(2). https://doi.org/10.1016/j.cad.2012.08.006

Chase, S. C. (2005). Generative design tools for novice designers: Issues for selection. *Automation in Construction*, *14*(6). https://doi.org/10.1016/j.autcon.2004.12.004

Chew, L. H., Teo, J., & Mountstephens, J. (2016). Aesthetic preference recognition of 3D shapes using EEG. *Cognitive Neurodynamics*, *10*(2). https://doi.org/10.1007/s11571-015-9363-z

Chien, S. F., & Flemming, U. (2002). Design space navigation in generative design systems. *Automation in Construction*, *11*(1). https://doi.org/10.1016/S0926-5805(00)00084-4

Clinton, Raymond, J., Prater, T., Morgan, K., & Ledbetter, F. (2018). NASA Additive Manufacturing Initiatives for Deep Space Human Exploration. *69th International Astronautical Congress (IAC);MSFC-E-DAA-TN61658;IAC-18C29*, *October*.

Dahmann, J., & Roedler, G. (2016). Moving towards standardization for system of systems engineering. *2016 11th Systems of Systems Engineering Conference, SoSE 2016*. https://doi.org/10.1109/SYSOSE.2016.7542953

Dahmann, J. S., & Baldwin, K. J. (2008). Understanding the current state of US defense systems of systems and the implications for systems engineering. *2008 IEEE International Systems Conference Proceedings, SysCon 2008*. https://doi.org/10.1109/SYSTEMS.2008.4518994

Dean, C. B., & Nielsen, J. D. (2007). Generalized linear mixed models: A review and some extensions. *Lifetime Data Analysis*, *13*(4). https://doi.org/10.1007/s10985-007-9065-x

Di Filippo, A., Lombardi, M., Marongiu, F., Lorusso, A., & Santaniello, D. (2021). Generative design for project optimization. *Proceedings - DMSVIVA 2021: 27th International DMS Conference on Visualization and Visual Languages*. https://doi.org/10.18293/dmsviva21-014

Ely, J. W., Dawson, J. D., Mehr, D. R., & Burns, T. L. (1996). Understanding logistic regression analysis through example. *Family Medicine*, *28*(2).

Eres, M. H., Bertoni, M., Kossmann, M., & Scanlan, J. (2014). Mapping customer needs to engineering characteristics: an aerospace perspective for conceptual design. *Journal of Engineering Design*, *25*(1–3). https://doi.org/10.1080/09544828.2014.903387

Fischer, T., & Herr, C. (2001). Teaching generative design. *... of the 4th Conference on Generative Art*.

Forsberg, K., & Mooz, H. (1992). The relationship of systems engineering to the project cycle. *EMJ - Engineering Management Journal*, *4*(3). https://doi.org/10.1080/10429247.1992.11414684

Gero, J. S., & Kazakov, V. (2001). A genetic engineering approach to genetic algorithms. *Evolutionary Computation*, *9*(1). https://doi.org/10.1162/10636560151075121

Gersh, J. R., McKneely, J. A., & Remington, R. W. (2005). Cognitive engineering: Understanding human interaction with complex systems. In *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)* (Vol. 26, Issue 4).

Ghobakhloo, M. (2020). Industry 4.0, digitization, and opportunities for sustainability. In *Journal of Cleaner Production* (Vol. 252). https://doi.org/10.1016/j.jclepro.2019.119869

Gower, J. C. (1971). A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, *27*(4). https://doi.org/10.2307/2528823

Grier, R. A. (2015). How high is high? A meta-analysis of NASA-TLX global workload scores. *Proceedings of the Human Factors and Ergonomics Society*, *2015-January*. https://doi.org/10.1177/1541931215591373

Gu, N., & Behbahani, P. A. (2021). Shape Grammars: A Key Generative Design Algorithm. In *Handbook of the Mathematics of the Arts and Sciences*. https://doi.org/10.1007/978-3-319-57072-3_7

Guha, S., Meyerson, A., Mishra, N., Motwani, R., & O'Callaghan, L. (2003). Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, *15*(3). https://doi.org/10.1109/TKDE.2003.1198387

Harris, J. K. (2021). Primer on binary logistic regression. *Family Medicine and Community Health*, *9*. https://doi.org/10.1136/fmch-2021-001290

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, *52*(C). https://doi.org/10.1016/S0166-4115(08)62386-9

Hedeker, D. (2005). Generalized Linear Mixed Models. In *Encyclopedia of Statistics in Behavioral Science*. https://doi.org/10.1002/0470013192.bsa251

Hollnagel, E., & Woods, D. D. (2005). Joint cognitive systems: Foundations of cognitive systems engineering. In *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering*.

Johnson, B. R. (2016). Design computing: An overview of an emergent field. In *Design Computing: An Overview of an Emergent Field*. https://doi.org/10.4324/9781315680057

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks - Conference Proceedings*, *4*. https://doi.org/10.4018/ijmfmp.2015010104

Khan, S., & Awan, M. J. (2018). A generative design technique for exploring shape variations. *Advanced Engineering Informatics*, *38*. https://doi.org/10.1016/j.aei.2018.10.005

Kiel, D. (2020). What do we know about "industry 4.0" so far? *26th International Association for Management of Technology Conference, IAMOT 2017*.

Knoll, D., Fortin, C., & Golkar, A. (2018). Review of concurrent engineering design practice in the space sector: State of the art and future perspectives. *4th IEEE International Symposium on Systems Engineering, ISSE 2018 - Proceedings*. https://doi.org/10.1109/SysEng.2018.8544387

Köhn, H., & Hubert, L. J. (2015). Hierarchical Cluster Analysis. In *Wiley StatsRef: Statistics Reference Online*. https://doi.org/10.1002/9781118445112.stat02449.pub2

Krish, S. (2011). A practical generative design method. *CAD Computer Aided Design*, *43*(1). https://doi.org/10.1016/j.cad.2010.09.009

Krupa, G. P. (2019). Application of agile model-based systems engineering in aircraft conceptual design. *Aeronautical Journal*, *123*(1268). https://doi.org/10.1017/aer.2019.53

Langdon, W. B., Poli, R., McPhee, N. F., & Koza, J. R. (2008). Genetic programming: An introduction and tutorial, with a survey of techniques and applications. *Studies in Computational Intelligence*, *115*. https://doi.org/10.1007/978-3-540-78293-3_22

Lapins, M. (1997). Aerospace engineering education: An industry view from a preliminary design perspective. *International Journal of Engineering Education*, *13*(5).

Leonard, S. T., & Droege, M. (2008). The uses and benefits of cluster analysis in pharmacy research. *Research in Social and Administrative Pharmacy*, *4*(1). https://doi.org/10.1016/j.sapharm.2007.02.001

Li, X., Demirel, H. O., Goldstein, M. H., & Sha, Z. (2021). *Exploring Generative Design Thinking for Engineering Design and Design Education*. https://doi.org/10.18260/1-2-1125.1153-38349

Lim, J. Z., Mountstephens, J., & Teo, J. (2020). Emotion recognition using eye-tracking: Taxonomy, review and current challenges. In *Sensors (Switzerland)* (Vol. 20, Issue 8). https://doi.org/10.3390/s20082384

Maguire, M. (2001). Methods to support human-centred design. *International Journal of Human Computer Studies*, *55*(4). https://doi.org/10.1006/ijhc.2001.0503

Manvailer, V. (2019). Partition around medoids (PAM) and silhouette plots. *University of Alberta*. https://sites.ualberta.ca/~ahamann/teaching/graphics/LabPAM.pdf

Marsh, A. (2008). Generative and Performative Design : A Challenging New Role for Modern Architects. *Design*, *2008*(c).

Mccormack, J., Dorin, A., & Innocent, T. (2004). Generative design : a paradigm for design research. *Proceedings of Futureground*.

McKnight, M. (2017). Generative Design: What it is? How is it being used? Why it's a game changer. *KnE Engineering*, *2*(2). https://doi.org/10.18502/keg.v2i2.612

Militello, L. G., Dominguez, C. O., Lintern, G., & Klein, G. (2010). The role of Cognitive Systems Engineering in the systems engineering design process. *Systems Engineering*, *13*(3). https://doi.org/10.1002/sys.20147

Miller, E. E., Boyle, L. N., Jenness, J. W., & Lee, J. D. (2018). Voice Control Tasks on Cognitive Workload and Driving Performance: Implications of Modality, Difficulty, and Duration. *Transportation Research Record*, *2672*(37). https://doi.org/10.1177/0361198118797483

Mountstephens, J., & Teo, J. (2020). Progress and challenges in generative product design: A review of systems. In *Computers* (Vol. 9, Issue 4). https://doi.org/10.3390/computers9040080

Nagaraj, D., & Werth, D. (2020). Towards a Generalized System for Generative Engineering. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3378184.3378227

Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J., & Peleska, J. (2015). Systems of systems engineering: Basic concepts, model-based techniques, and research directions. *ACM Computing Surveys*, *48*(2). https://doi.org/10.1145/2794381

Nti, I. K., Adekoya, A. F., Weyori, B. A., & Nyarko-Boateng, O. (2022). Applications of artificial intelligence in engineering and manufacturing: a systematic review. In *Journal of Intelligent Manufacturing* (Vol. 33, Issue 6). https://doi.org/10.1007/s10845-021-01771-6

Núñez, R., Allen, M., Gao, R., Miller Rigoli, C., Relaford-Doyle, J., & Semenuks, A. (2019). What happened to cognitive science? In *Nature Human Behaviour* (Vol. 3, Issue 8). https://doi.org/10.1038/s41562-019-0626-2

Osmundson, J. S., Huynh, T. V., & Langford, G. O. (2008). Emergent behavior in systems of systems. *18th Annual International Symposium of the International Council on Systems Engineering, INCOSE 2008*, *4*.

Oxman, R. (2002). The thinking eye: Visual re-cognition in design emergence. *Design Studies*, *23*(2). https://doi.org/10.1016/S0142-694X(01)00026-6

Özkar, M., & Kotsopoulos, S. (2008). Introduction to shape grammars. *ACM SIGGRAPH 2008 Classes*. https://doi.org/10.1145/1401132.1401182

Panchal, J. H., Fuge, M., Liu, Y., Missoum, S., & Tucker, C. (2019). Special issue: Machine learning for engineering design. In *Journal of Mechanical Design* (Vol. 141, Issue 11). https://doi.org/10.1115/1.4044690

Parish, Y. I. H., & Müller, P. (2001). Procedural modeling of cities. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001*. https://doi.org/10.1145/383259.383292

Pham, D. T., & Pham, P. T. N. (1999). Artificial intelligence in engineering. International *Journal of Machine Tools and Manufacture*, 39(6), 937-949.

Piasecki, M., & Hanna, S. (2011). A redefinition of the paradox of choice. *Design Computing and Cognition ’10*. https://doi.org/10.1007/978-94-007-0510-4_19

Pidaparti, R. M. (2023). *Detail Design*. https://doi.org/10.1007/978-3-031-25969-2_7

Pignatiello, G. A., Martin, R. J., & Hickman, R. L. (2020). Decision fatigue: A conceptual analysis. In *Journal of Health Psychology* (Vol. 25, Issue 1). https://doi.org/10.1177/1359105318763510

Pilagatti, A. N., Vecchi, G., Atzeni, E., Iuliano, L., & Salmi, A. (2022). Generative Design and new designers' role in the manufacturing industry. *Procedia CIRP*, *112*. https://doi.org/10.1016/j.procir.2022.09.010

Polman, E., & Vohs, K. D. (2016). Decision fatigue, choosing for others, and self-construal. *Social Psychological and Personality Science*, *7*(5). https://doi.org/10.1177/1948550616639648

Prabaswari, A. D., Basumerda, C., & Utomo, B. W. (2019). The Mental Workload Analysis of Staff in Study Program of Private Educational Organization. *IOP Conference Series: Materials Science and Engineering*, *528*(1). https://doi.org/10.1088/1757-899X/528/1/012018

Prusinkiewicz, P., & Lindenmayer, A. (1990). Algorithmic Beauty of Plants. *Springer-Verlag*.

Ranganathan, P., Pramesh, C., & Aggarwal, R. (2017). Common pitfalls in statistical analysis: Logistic regression. *Perspectives in Clinical Research*, *8*(3). https://doi.org/10.4103/picr.PICR_87_17

Rao, V. M., & Francis, R. A. (2022). New conceptual approaches to cognition in systems engineering: applying the 4 E's of cognition. *Systems Engineering*, *25*(6). https://doi.org/10.1002/sys.21637

Ryen, E. (2008). Overview of the System Engineering Process. *NDDOT - North Dakota Department of Transportation*, *March*.

Scheidl, R., & Winkler, B. (2010). Model relations between conceptual and detail design. *Mechatronics*, *20*(8). https://doi.org/10.1016/j.mechatronics.2010.04.008

Schulz, A., Wang, H., Grinspun, E., Solomon, J., & Matusik, W. (2018). Interactive exploration of design trade-offs. *ACM Transactions on Graphics*, *37*(4). https://doi.org/10.1145/3197517.3201385

Schumann, H., Wendel, H., Braukhane, A., Berres, A., Gerndt, A., & Schreiber, A. (2010). Concurrent Systems Engineering in Aerospace: From Excel-based to Model Driven Design. *8th Conference on Systems Engineering Research (CSER)*, *Idm*.

Sebastian, P., & Ledoux, Y. (2009). Decision support systems in preliminary design. In *International Journal on Interactive Design and Manufacturing* (Vol. 3, Issue 4). https://doi.org/10.1007/s12008-009-0077-5

Shea, K., Aish, R., & Gourtovaia, M. (2005). Towards integrated performance-driven generative design tools. *Automation in Construction*, *14*(2 SPEC. ISS.). https://doi.org/10.1016/j.autcon.2004.07.002

Sheard, S. A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering*, *12*(4). https://doi.org/10.1002/sys.20124

Sigmund, O., & Petersson, J. (1998). Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural Optimization*, *16*(1). https://doi.org/10.1007/BF01214002

Singh, V., & Gu, N. (2012). Towards an integrated generative design framework. *Design Studies*, *33*(2). https://doi.org/10.1016/j.destud.2011.06.001

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, *7*(3). https://doi.org/10.1068/b070343

Stojmenova, K., & Sodnik, J. (2018). Detection-response task—uses and limitations. In *Sensors (Switzerland)* (Vol. 18, Issue 2). https://doi.org/10.3390/s18020594

Troiano, L., & Birtolo, C. (2014). Genetic algorithms supporting generative design of user interfaces: Examples. *Information Sciences*, *259*. https://doi.org/10.1016/j.ins.2012.01.006

Tuerlinckx, F., Rijmen, F., Verbeke, G., & De Boeck, P. (2006). Statistical inference in generalized linear mixed models: A review. In *British Journal of Mathematical and Statistical Psychology* (Vol. 59, Issue 2). https://doi.org/10.1348/000711005X79857

Tyflopoulos, E., Flem, D. T., Steinert, M., & Olsen, A. (2018). State of the art of generative design and topology optimization and potential research needs. *Proceedings of NordDesign: Design in the Era of Digitalization, NordDesign 2018*.

Van der Laan, M. J., Pollard, K. S., & Bryan, J. (2003). A new partitioning around medoids algorithm. *Journal of Statistical Computation and Simulation*, *73*(8). https://doi.org/10.1080/0094965031000136012

Verma, M., & Wood, W. H. (2001). Form follows function: Case-based learning over product evolution. *Proceedings of the ASME Design Engineering Technical Conference*, *3*. https://doi.org/10.1115/detc2001/dfm-21182

Walden, D. D., Roedler, G. J., & Forsberg, K. (2015). INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners. *INCOSE International Symposium*, *25*(1). https://doi.org/10.1002/j.2334-5837.2015.00089

Weisstein, E. (2021). String Rewriting System. MathWorld-A Wolfram Web Resource. https://mathworld.wolfram.com/StringRewritingSystem.html

Zhang, Z., & Chu, X. (2010). A new approach for conceptual design of product and maintenance. *International Journal of Computer Integrated Manufacturing*, *23*(7). https://doi.org/10.1080/09511921003736766

Zuur, A. F. (2016). Bayesian data analysis in ecology using linear models with R, BUGS and Stan. *Animal Behaviour*, *111*. https://doi.org/10.1016/j.anbehav.2015.10.020