MULTIOBJECTIVE TOPOLOGY OPTIMIZATION FOR PRELIMINARY DESIGN USING GRAPH THEORY AND L-SYSTEM LANGUAGES

A Dissertation

by

BRENT RYAN BIELEFELDT

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Darren Hartl
Committee Members,	Ergun Akleman
	Richard Malak
	John Whitcomb
Head of Department,	Rodney Bowersox

May 2020

Major Subject: Aerospace Engineering

ABSTRACT

Topology optimization is a powerful tool that, when employed at the preliminary stage of the design process, can determine potential structural configurations that best satisfy specified performance objectives. However, the use of conventional topology optimization approaches such as density-based and level set methods requires a fair amount of user knowledge of or intuition for both the design problem being considered and the desired result. While straightforward for simple structural problems with a relatively small design space, advancements in the area of smart materials and a growing interest in developing structures with increased multifunctionality may begin to render these methods as ineffective. Thus, there is a growing need for an inherently multiobjective preliminary design tool capable of exploring a vast design space to identify well-performing solutions to problems with which users have little/no intuition or experience.

This work proposes the use of a heuristic alternative to conventional topology optimization approaches which couples a genetic algorithm with a parallel rewriting system known as a Lindenmayer System (L-System). The L-System encodes design variables into a string of characters that, when interpreted by a deterministic algorithm, governs the development of the topology. In particular, this work explores two distinct L-System interpretation approaches. The first is a geometry-based approach known as turtle graphics, which tracks its spatial position and orientation at all times and constructs line segments between specified coordinates. The second is a newly-developed graph-based approach referred to as Spatial Interpretation for the Development of Reconfigurable Structures (SPIDRS). This algorithm is based on the nodes, edges, and faces of a planar graph, allowing for an edge- and face-constructing agent to move more freely around the design space and introduce deliberate and natural topological modifications. This graph-based approach can also be extended to consider a three-dimensional structural design domain, the first known demonstration of 3-D L-System topology optimization. It will be demonstrated that the proposed L-System topology optimization framework effectively explores the physical design space and results in configurations comparable to both known optimal or ideal solutions as well as those found using conventional topology optimization methods, but with the advantage of straightforward multiobjective/multiphysical extension. The implementation of a sizing optimization scheme to determine optimal structural member thicknesses for SPIDRS-generated topologies will also be discussed, and several potential multiphysical design applications will be introduced.

DEDICATION

To Mom, Dad, and Jessie,

I couldn’t have done this without you.

"It is not knowledge, but the act of learning, not possession but the act of getting there, which grants the greatest enjoyment."

-*Carl Friedrich Gauss*

ACKNOWLEDGMENTS

I would be remiss if I did not first thank God for blessing me with His gifts and continually opening doors to allow me to do what I love - learn. May this work and all future works serve to glorify Your name.

I could not have accomplished any of this without the love and support of my parents. I can imagine it’s challenging at times to have a child that wants to spend ten years in college, four of those working on a “video game” that may or may not be working depending on the day, but there hasn’t been a single moment where they’ve stopped encouraging me. Part of what has made me the researcher I am today has come from watching them, seeing the importance of working hard, doing your best, and (most importantly) having fun. I could not ask for better parents.

I would also like to thank my wife and best friend Jessica for the incredible amount of encouragement she’s given me over the past five years. From practically bullying me into staying for my PhD to listening to my defense presentation about a dozen times, she’s been there supporting me for every step of this journey. The fact that she’s been able to do all this while simultaneously getting two degrees and starting a full-time job is truly incredible. Although she jokes about it, I truly view this as “our PhD”.

I am not where I am today without the direction of my advisor, Dr. Darren Hartl, who plucked me from Alabama as a recent graduate who knew nothing about the world of research (or Python), was patient with me as I adjusted to the rigorous demands of graduate school, and then somehow saw enough in the first two years to offer me an opportunity to pursue a doctorate. Since then, he has helped me achieve my goal of doing “just structures research” (in Python), kept me upbeat when I was banging my head against a problem for weeks, kept me grounded when I had achieved the latest world-changing result, and never gotten too upset after catching me watching football or soccer highlights. His expectations have pushed me further than I ever thought I could go and molded me into a self-reliant researcher with high standards and lofty ambitions.

I would like to thank my committee members for their influence and input on this work.

SPIDRS doesn't get off the ground without the help of Dr. Ergun Akleman, who views things in an entirely different way than we did and helped clarify many of the abstract foundations of graph theory. Interacting with Dr. Rich Malak has taught me so much about how to view optimization and helped shape my understanding of the benefits and drawbacks of traditional topology optimization techniques. Finally, comments provided by Dr. John Whitcomb during my preliminary exam drove me to TA AERO 306 and indirectly added an entire chapter to this work.

Of course, this work has not strictly been a Texas A&M effort. I would like to thank Dr. Greg Reich, who has been a fantastic mentor for me during my time as a doctoral student. Under his supervision, what was supposed to be a 1-year side project while I studied for Quals turned into four years of exploration and a true research passion. I have learned numerous lessons from him during my time as a doctoral student, from "A's want to work with A's" to "work hard, play hard". Interacting with him as well as Drs. Phil Beran, Alex Pankonien, and Josh Hodson during three separate internships at the Air Force Research Laboratory contributed to this work a great deal and gave me insight into the career path I want to pursue. I also want to thank Dr. Marcelo Kobayashi from the University of Hawaii for inspiring such a fascinating area of study and development with his work on the map L-System.

My experience at Texas A&M has been greatly enriched by those students who have been a part of the M^{2}AESTRO Laboratory over the last four years. I can honestly say that I probably don't make it through Quals without studying with Pedro Leal and Will Scholten. Patrick Walgren, Jacob Mingear, and Allen Davis have been great sounding boards in the office as I talked out parts of my research and were always willing to critique a figure or slide. I've enjoyed working with Madalyn Mikkelsen on some of the more "fun" SPIDRS application problems, and her questions have given me an opportunity to flush out some of the philosophical reasoning behind different SPIDRS operations. Holly Patterson enabled an entirely new chapter of this dissertation with one summer's worth of work, while recent work done by Mason Ward has been phenomenal and probably deserves far more discussion in Chapter 6 than I was able to provide.

Thank you all.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professors Darren Hartl (advisor) and John Whitcomb of the Department of Aerospace Engineering, Professor Ergun Akleman of the Department of Visualization, and Professor Richard Malak of the Department of Mechanical Engineering.

All work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a combination of Texas A&M University and Air Force Office of Scientific Research (AFOSR) Lab Task 16RQCOR284 funding.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

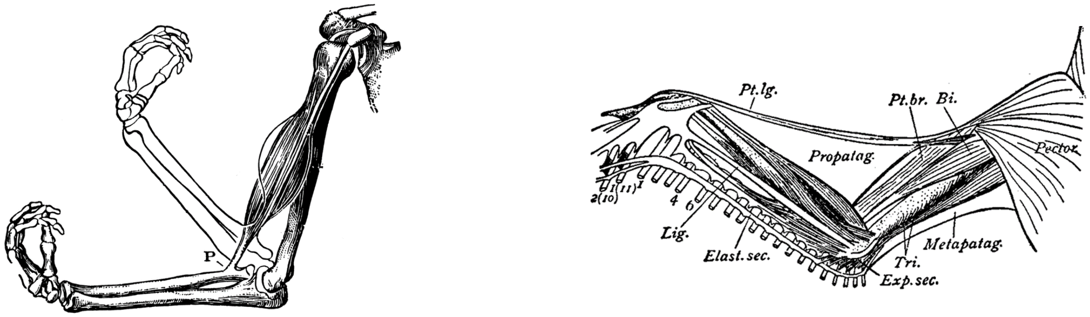
LIST OF TABLES

TABLE Page

1. INTRODUCTION

When considering the design of an engineering structure, it is crucial to evaluate the effect that material type and placement has on structural functionality. A simple example of this found in nature is shown in Figure 1.1. The human arm (Figure 1.1a) consists of rigid skeletal structures connected to cartilage, tendons, ligaments, and muscles, which, when combined together in a specific configurations, enable a range of motions and capabilities. A bird wing (Figure 1.1b) consists of the same intricate and complex multi-membered structures but demonstrates completely different functionalities because these materials are laid out in a different configuration. Thus, determining the material placement and type that best utilizes material properties to achieve specified objectives is an extremely important aspect of the design process.

In the fields of structural design and optimization, the design of material distributions inside a given domain subjected to boundary conditions and constraints such that specified performance metrics of the structure are maximized is known as topology optimization. Routinely deployed at the conceptual or preliminary stage of the design process, it is a powerful tool that can determine potential system configurations and greatly influence the performance of the finalized design [2].



(a) Human arm.

(b) Bird wing [1].

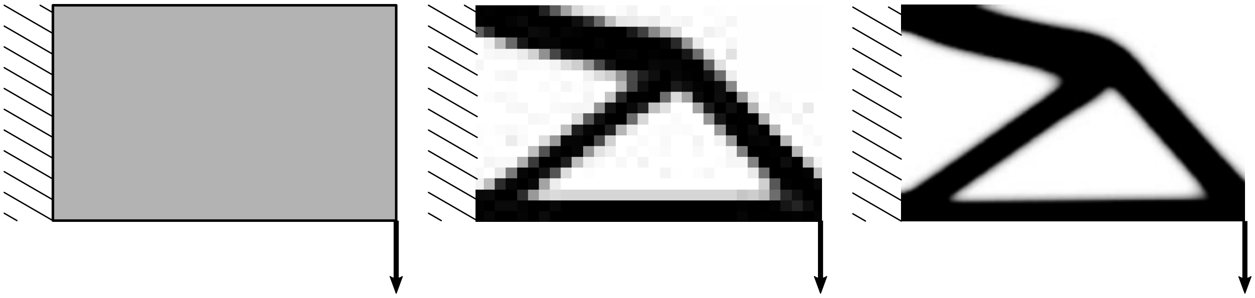
Figure 1.1: Illustrations of structural topology using the anatomy of the human arm and a bird wing. Both structures consist of the same materials distributed in different topologies, allowing for completely different functionalities.

Topology optimization has become a popular area of research in the design community in recent years due to the refinement of classical topology optimization methodologies as well as the development of novel promising techniques. Additionally, topology optimization has been adapted to consider not only structural responses, but also those from heat transfer, fluid flow, acoustics, and other multiphysical disciplines [3].

1.1 Review of Common Topology Optimization Methodologies

The most widely-accepted methodologies for structural topology optimization are classified as *density-based* methods, which include the popular Solid Isotropic Material with Penalization (SIMP) method (Figure 1.2). Density-based methods consider a dense domain of finite elements with the goal of minimizing an objective function by identifying whether a given element should consist of solid material or void (i.e., design variables have discrete “0/1” values to represent “solid/void” regions) [4, 5]. This represents an *explicit* topological representation method, as structural boundaries are explicitly defined by the design variables of the problem. However, given that each 2-D pixel or 3-D voxel is associated with its own design variable, the dimensionality of the design space increases rapidly when considering that a higher resolution of the design space (which results in more accurate simulated results) requires an increased number of pixels/voxels [3, 6]. Density-based methods also suffer from a phenomenon known as “checkerboarding”, or the formation of adjacent solid-void elements arranged in a checkerboard pattern [3, 4, 7], as well as mesh dependencies, where different topologies can result from identical design domains of different discretization sizes. Furthermore, these methods can result in singularities in associated finite element matrices [3]; such difficulties can be avoided by replacing discrete design variables with continuous variables, but results in the formation of grey “transition” material between solid and void regions [8, 9, 10, 11]. Various *regularization* techniques have been created to prevent numerical issues and control the quality of final results, such as filtering methods [7, 12, 13, 14], length-scale constraints [8, 15, 16], and projection schemes [8, 9, 10, 11, 17].

Another well-known topology optimization methodology is the level set method, which represents structural boundaries (i.e., the interface of solid and void) as the zero-level curve of a



(a) Initial design domain.

(b) Solution with a volume fraction of 0.5.

(c) Solution after applying sensitivity filtering [12].

Figure 1.2: Example of density-based topology optimization applied to a cantilever beam design problem (adapted from Andreassen et al. [18]).

scalar level set function [3, 19, 20, 21] and is illustrated in Figure 1.3. This method is an example of an *implicit* topological representation method, as structural boundaries are defined based on an implicit parameterized function rather than explicitly by the design variables. This allows for the convenient treatment of topological changes, as structural boundaries can be modified by using the physical problem and optimization conditions to control the output of the level set function. A level set function can be parameterized using finite element method (FEM) basis functions [20, 22, 23, 24, 25], radial basis functions (RBFs) [26, 27, 28, 29], or Fourier series [30], which determine the design freedom and level of detailedness of the material boundaries. Performance analysis requires the mapping of a parameterized level set function to a structural model, which can have an impact on the accuracy of the structural response and the quality of the final result of the optimization process. Currently utilized methods for this mapping procedure include conforming discretization (most accurate but computationally expensive) [31, 32], immersed boundary techniques (accurate but difficult to implement) [25, 29], and density-based mapping (accurate with regularization techniques) [20, 22, 26]. Additionally, the level set method also suffers from poor rate of convergence, convergence to local minima, and difficulties in dealing with constraints [21], some of which can be prevented by using regularization techniques such as sensitivity and density filtering [25, 33, 34].

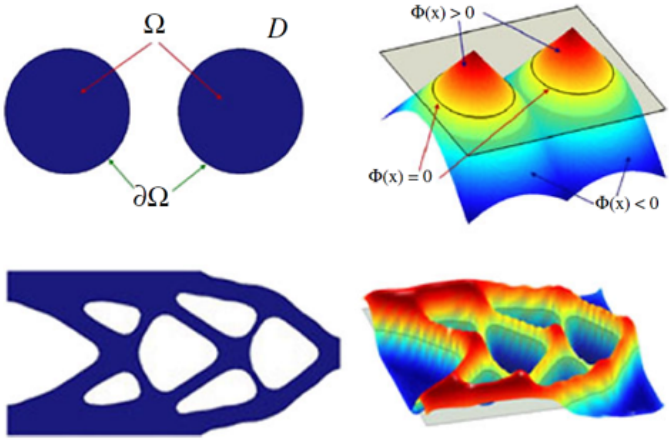


Figure 1.3: Example of level set topology optimization, illustrating both 2-D topology (left) and corresponding level set function (right) (adapted from Luo et al. [35]).

In addition to the more widely-used approaches described above, several other interesting methodologies have been proposed in recent years. Evolutionary Structural Optimization (ESO) is a "hard-kill" method based on the idea that optimal structures are made up of members that are all equivalently stressed [36]. ESO gradually removes "inefficient" material from the design domain based on heuristic criteria such as elemental stress [36] or other sensitivity value [37]. ESO differs from density-based methods in that the discrete design space is not relaxed, meaning that ESO does not allow transition material between solid and void regions. A variant known as bi-directional ESO (BESO) also allows for material to be added in regions where elements have a high criterion value [38, 39]. In addition to potential numerical instabilities due to their strictly discrete nature, ESO/BESO implementations have been shown to break down for simple structural problems [40] and utilize heuristic criteria that may not relate to the objective function [41]. The ground structure method (GSM) represents the initial domain as a series of perfect, slender structural members that connect every member in the set of user-defined nodes to every other member in that set. The topology of the domain is then evolved by varying the cross-sectional areas of each member in the structure, with an area of zero signifying that a member has been removed from the structure [42, 43, 44]. While GSMs are adept at finding solutions to truss problems, the quality of optimized results depend on the initial locations and number of user-defined nodes that influence the connectivity of structural members, which can quickly increase the number of design variables required as well as the computational time needed to analyze initially dense configurations [43, 45]. Similar to BESO, a method known as the growing ground structure method (GGSM) allows for both the addition and removal of structural members, allowing for the initial ground structure to remain relatively sparse [45].

1.2 Methods for Multiobjective Topology Optimization

While determining which methodology can best represent the optimal topology for a given problem is a challenge in its own right, this process is made considerably more difficult when that problem has *multiple* performance objectives. These objectives are often dissimilar, leading to a compromised topology if pursued simultaneously [46]. Thus, multiobjective topology optimization problems rarely possess a single "optimal" solution but instead exhibit a series of solutions that are classified as *Pareto-optimal*. Solutions are Pareto-optimal if no other solution exists that is better with respect to one objective and is as good with respect to other objectives [47]. Such solutions combine to form the *Pareto frontier*, which can provide the user with insight into the characteristics of the problem [48].

Pareto-optimal solutions for topology optimization problems can be obtained using both gradient and non-gradient optimization approaches. Gradient methods compute/estimate the gradients of objectives to changes in design variables, then adjust the design variables in small increments towards perceived solutions that minimize those gradients. These methods are capable of considering a large number of design variables, and the computational cost of computing objective gradients is often relatively low, meaning one can quickly converge to an optimal solution [49, 50]. Gradient methods are the most popular method for topology optimization and have have been used extensively with the SIMP [4, 18], level set [20, 25], and GSM [42, 43, 45] methodologies. However, these methods have difficulties with problems containing multiple local minima, discontinuous design spaces, or discrete variables [50]. Additionally, gradient methods seek to minimize/maximize a single scalar function, meaning that for multiobjective problems the objectives must be combined into a single objective. This is commonly achieved using the method of weighted sums, where the objective function is the sum of each objective multiplied by some predetermined weight based on importance or other factors [51, 52]. Variants of the weighted sums method known as compromise programming [46] and physical programming [53] have also been used for multiobjective topology optimization. However the determination of suitable weights is non-trivial [54, 55, 56] and often require a general intuition for the problem beforehand [48]. The single objective function approach of gradient methods also dictates that only one topology can be generated per optimization, meaning that each solution on the Pareto front must be generated by an individual optimization process using individual objective weights. Furthermore, gradient methods using weighted objective functions have been shown to fail in capturing Pareto-optimal solutions for problems with non-convex Pareto frontiers [46, 54, 55, 56]

As the name suggests, non-gradient optimization approaches do not require gradient information and instead solely use function evaluations of the objective function(s) to converge to a solution. One of the most common non-gradient methods, and the one most commonly utilized in topology optimization, is known as a genetic algorithm. Genetic algorithms are motivated by the principles of natural selection and construct an optimization procedure that requires minimal amounts of information about the problem [48, 50]. Natural chromosomes are made up of a series of genes that control an organism's observable characteristics or traits; similarly, in genetic algorithms designs are represented by pseudo-chromosomal representations of design parameters. Genetic algorithms begin with a randomly generated set (or "population") of design solutions that are evaluated, assigned a fitness value based on their performance, and then modified by three operators borrowed from the fundamental ideas of genetics (reproduction, crossover, and mutation) to create a new and hopefully better population [57]. This process continues until some specified termination criterion is met.

Genetic algorithms are attractive when considering multiobjective topology optimization because they work with a population of solutions rather than a single solution, meaning that the set of solutions comprising the Pareto frontier can be obtained simultaneously. Their stochastic behavior also generally allows for these algorithms to better search the global design space, thereby avoiding convergence to local minima. Additionally, the lack of reliance on gradients makes genetic algorithms an appealing option when dealing with discrete design variables and discontinuous design spaces [50]. Genetic algorithms have been successfully coupled with various topology optimization methodologies, including level set [58, 59], ESO/BESO [60, 61], and GSM [62], as well as used in conjunction with gradient-based methods on topology optimization problems [63, 64]. Sigmund has argued against their use in problems where the topology is represented explicitly by the design variables (e.g., SIMP, ESO/BESO) [49], citing their computational cost relative to gradient-based methods and the necessity of using coarse meshes that are unable to correctly represent the underlying physical response of the structure. However, he does concede that genetic algorithms are useful when applied to implicitly-represented topologies or problems with a discontinuous design space.

1.3 Topology Optimization in Preliminary Design

As previously mentioned, topology optimization is routinely implemented at the preliminary stage of structural design due to its ability to determine optimal system configurations without well-defined initial conditions. This is exemplified in Figure 1.4, which illustrates the design process of a generic structural part [65]. Given a design domain (A) and a series of predefined boundary conditions (denoted in red), topology optimization can be used to determine an optimal distribution of material within the design domain that maximizes the performance of the structure while satisfying the specified boundary conditions. Typically, the objectives that quantify structural performance include minimizing mass while maximizing the stiffness or durability of the part. The resulting "optimal" structural configuration (B) is then combined with safety factors or other metrics to generate a preliminary design (C). This preliminary design is refined using variables such as geometric features (D) and the thicknesses or cross-sectional areas of structural members (E) to arrive at a finalized design (F). Thus, employing topology optimization early in the preliminary design phase can significantly impact the performance of the final structure.

Regardless of the tools utilized, the preliminary stage of structural design is largely influenced

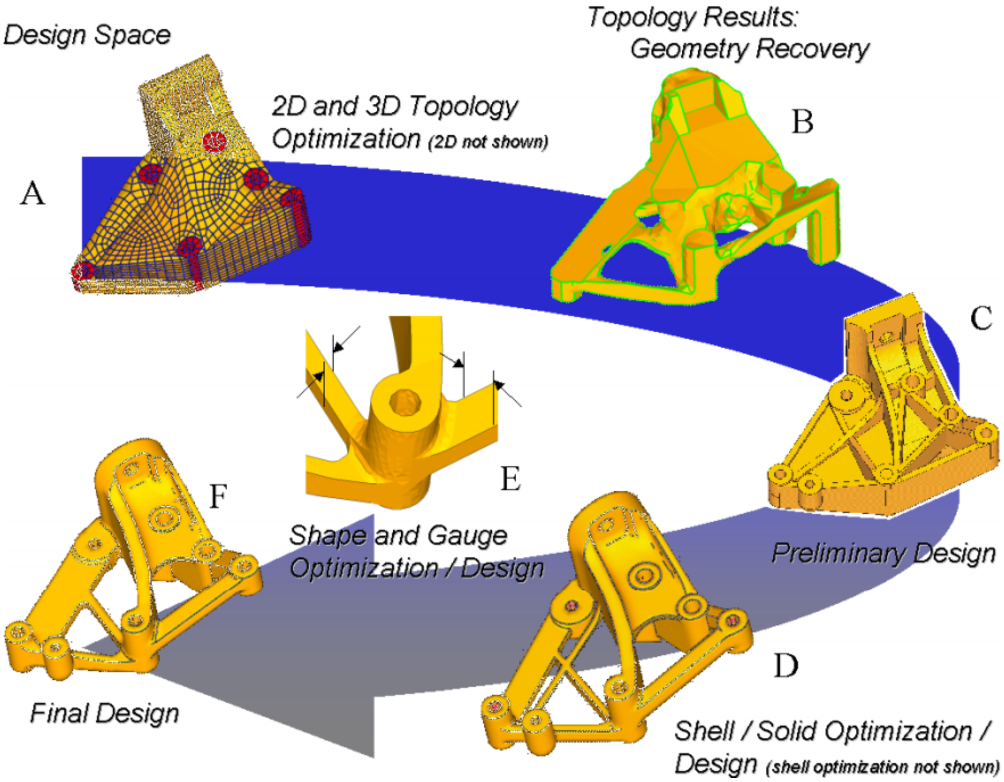


Figure 1.4: Illustration of the design process of a generic structural part (adapted from [65]).

by designers' previous experience with or intuition for the problem at hand [66, 67]. A closer inspection of several of the methods discussed above reveal that topology optimization is no different. SIMP, level set, and GSM implementations all require an initial volumetric constraint, meaning that designers must have a knowledge of approximately how much material they want to use. Each of the three methods are also typically reliant upon gradient-based optimization approaches. As discussed in Section 1.2, this means that for multiobjective problems the objectives must be combined into a single scalar objective using the method of weighted sums or some other method. However, the determination of these weights is non-trivial and requires that designers have a general intuition of the problem [48]. SIMP and GSM implementations are also explicit topological representations, meaning that the design variables directly define the topology of the structure. Given that both the accuracy of the analysis and the resulting topology are functions of the domain discretization, both methods are plagued by rapidly increasing design space dimensionality and require designers to determine the best trade-off between accuracy and computation time.

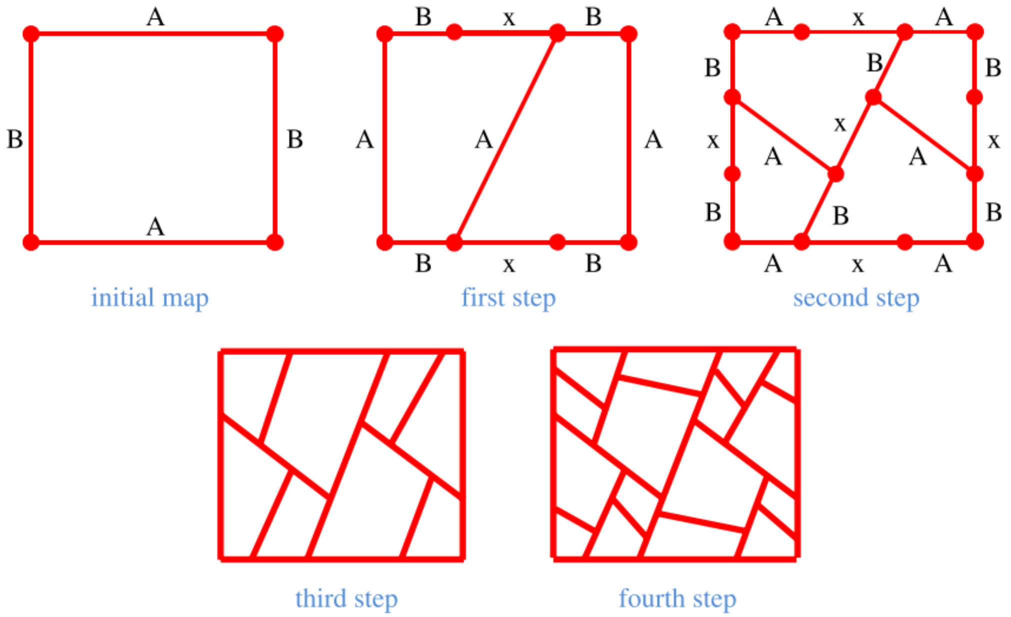
For simple structural problems with a limited design space such as the example shown in Figure 1.4, which often come with a general knowledge or intuition, such topology optimization methodologies are sufficient. However, the maturation of adaptive material technology and a growing interest in developing structures with increased multifunctionality may begin to render these methods as ineffective. The goal of designing structures with multiple functionalities is inherently multiobjective, and the vast design space associated with these problems makes developing a knowledge of/intuition for them increasingly difficult. Thus, designers are capable of exploring only a small subset of potential solutions by implementing a volumetric constraint or weighting objectives within a single function. Furthermore, these systems typically respond to a certain stimulus such as stress, heat, electrical current/voltage, magnetic field, moisture, or light by altering some of their physical and/or chemical properties. The complex physics models required to accurately analyze these responses greatly complicate the derivative calculations necessary for gradient-based topology optimization approaches. Finally, the vast design space is only exacerbated by the increasing design space dimensionality present in both SIMP and GSM implementations. Therefore, there is a growing need for an inherently multiobjective preliminary design tool capable of exploring a vast design space to identify well-performing solutions to problems for which designers have little/no intuition or experience.

1.4 L-System Topology Optimization

Recently, a novel class of bio-inspired topological representation methodologies capable of generating discrete and continuum-like structures was developed as an alternative to the more rigorous mathematical approaches described above. These methods utilize a biological model known as a Lindenmayer System (L-System) which, when coupled with an interpreter, creates and executes a series of tasks that develop the topology in stages. It should be noted that these methods are classified as implicit topological representations, as the topology is realized indirectly via the encoding from the L-System. The evolution of the L-System encoding, and by extension the topology of the structure, is governed by a genetic algorithm [68], which imitates the process of natural selection by stochastically choosing well-performing designs and modifying the "genome" (i.e., set of L-System design variables) of each design via crossover or mutation to arrive at a group of optimal designs. Thus, the proposed methodology represents biological-inspired design in the truest sense, as a biological model capable of simulated growth is evolved using natural selection to find optimal designs for a set of competing objectives.

Research on applying an L-System to topology optimization problems has been pioneered by Kobayashi et al. and focused primarily on the map L-System. The map L-System is inspired by the process of cellular division in living organisms and evolves the structural topology by executing a series of tasks that subdivides the domain in stages as shown in Figure 1.5 [6, 69]. This results in a diverse set of topological designs generated with relatively few design variables. Map L-System topology optimization has been demonstrated using interesting application problems involving flapping wing mechanisms [69, 70] and aircraft wing structures [71, 72, 73]. However, this method has been marginally applied to benchmark topology optimization problems [74], and its use often requires the application of an osmotic equilibrium [74] or shape morphing [71, 73] post-processing step. Furthermore, the map L-System is currently unable to consider truly 3-D structural design domains, which may hinder its use in real-world applications that cannot be resolved into 2-D abstractions.

The goal of this dissertation is to develop a topology optimization framework capable of serving as a multiobjective preliminary design tool for highly novel design problems with which a designer has little/no intuition for or knowledge of. The result of this framework would not be a single deterministic solution, but rather the exploration of the design space for non-intuitive solutions, the identifications of trends and commonalities between well-performing designs, and the gaining of insights into specific solutions that conventional design methods may not consider. As shown in Table 1.1, L-System topology optimization methods are an attractive option for such a framework, as they are classified as implicit topological representations, which limits the number of design variables needed, and do not require an initial volumetric constraint to obtain solutions.



initial map

first step

second step

third step

fourth step

Figure 1.5: Example of how the map L-System evolves the topology of a design domain (from [69]).

Additionally, their reliance on genetic algorithms, while admittedly increasing computation time, allows for both the convenient treatment of problems with multiple objectives and their application to multifunctional structural design problems that consider complex physical fields for which gradient calculations become exceedingly difficult. In an effort to further broaden the utility of this framework, the L-System topology optimization method to be employed should also be able to consider 3-D structural design domains.

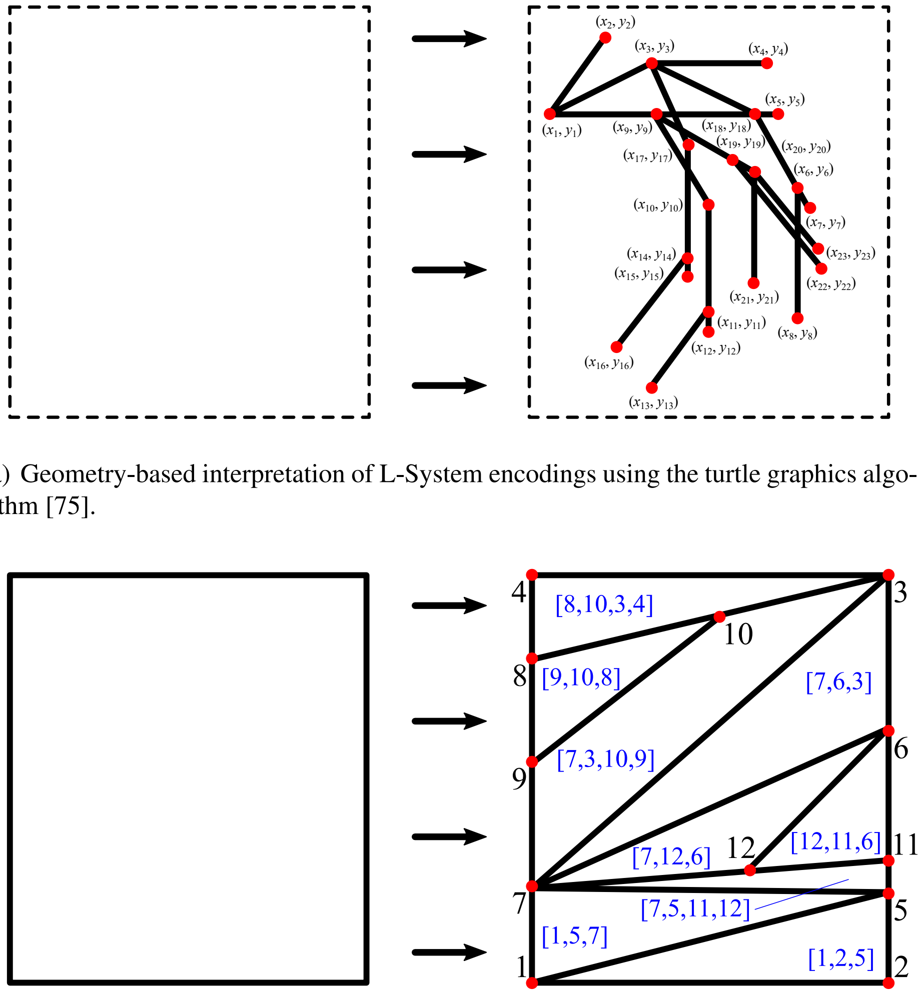
Table 1.1: Overview of the desired criteria for an improved topology optimization framework employed at the preliminary stage of the design process.

	Gradient-Based	Level Set	GSM	L-System
Lack of Volume Constraint				✓
Implicit Topological Representation		✓		✓
Inherently Multiobjective				✓
Fundamentally Multifunctional				✓
3-D Capability	✓	✓	✓	✓*

*In this work; currently not possible with Map L-System implementations.

To identify a potential L-System topology optimization method for this preliminary design framework, this effort considers two distinct methods of interpreting L-System encodings. First, the use of a geometry-based interpretation method using an algorithm known as turtle graphics [75] is explored, which constructs continuous, straight line segments by tracking the spatial position and orientation of a line-constructing agent inside a given 2-D structural design domain as shown in Figure 1.6a. This method is capable of generating topologies inspired by ramified, or branched, structures found in numerous natural systems, including leaf [76] and insect wing venation [77, 78] and mammalian respiratory [79, 80] and vascular systems [5, 81]. The potential of turtle graphics-interpreted L-System topology optimization has been demonstrated for simple non-structural problems such as temperature [82, 83] and fluid transport [6], which led to the preliminary investigations into its use in structural problems that have motivated this work [84, 85]. Second, this work proposes the use of a newly-developed interpreter known as Spatial Interpretation for the Development of Reconfigurable Structures, or SPIDRS. SPIDRS is a graph-based algorithm which interprets L-System commands as a function of the nodes, edges, and faces (i.e., enclosed portions) of a graph, regardless of the spatial location of these features. When coupled with a parameterized L-System, this method enables an edge-constructing agent to freely move about the graph, generating new connected features as illustrated in Figure 1.6b. Fundamentally, the proposed SPIDRS algorithm is a network design tool in that it simply defines connections between a set of nodes. In the context of structural design, these connections are assumed to represent the placement of high aspect ratio structural members (i.e., beam or frame elements). The nature of the SPIDRS algorithm also allows for several design extensions, such as the optimization of the size of the physical design domain and the conversion of closed edge sets into faces and of faces into solid volumes, as well as the ability to extend into 3-D space, which is not possible with the geometry-based turtle graphics.

For any novel design method to gain acceptance and be applied to the complex multifunctional design problems previously mentioned, it must first demonstrate a capability of determining optimal system configurations for benchmark problems. Thus, the bulk of this work focuses on the



(a) Geometry-based interpretation of L-System encodings using the turtle graphics algorithm [75].

(b) Graph-based interpretation of parameterized L-System encodings using the SPIDRS algorithm.

Figure 1.6: Illustration of how the two L-System interpretation methods explored in this work evolve the topology of a given design domain.

rigorous analysis of the two proposed L-System interpretation methods when considering simple, well-known topology optimization problems. For each design problem considered, efforts will be made to compare the performance of resulting designs both to mathematically known optimal or ideal solutions as well as those generated using conventional topology optimization methodologies. Where necessary, the drawbacks and limitations of the proposed methods will be discussed and improvements will be suggested and/or implemented. It will be shown that L-System topology optimization can be considered a capable alternative to conventional topology optimization methods and is a viable preliminary design tool when considering complex multiobjective, multi-physical structural problems.

1.5 Dissertation Summary

In summary, this dissertation is organized as follows:

- Chapter 2 introduces how an L-System encodes design variables and how this encoding is interpreted using the geometry-based turtle graphics algorithm to generate 2-D structural topologies. This framework is then applied to several benchmark topology optimization problems to assess the capabilities of geometry-based interpreted L-System topology optimization. Experimental characterization of fabricated structural topologies also explores whether the proposed framework is capable of generating physically-realizable designs.

- Chapter 3 discusses the development of the graph-based SPIDRS interpretation algorithm. After considering several design problems and demonstrating several novel extensions enabled by its graph-based foundation, several key components of the SPIDRS algorithm are considered and modified to further optimize the performance of resulting designs.

- Chapter 4 details how the SPIDRS algorithm developed in Chapter 3 can be extended to consider 3-D design domains with minimal changes to both the algorithm and the L-System. Several simple frame and compliant mechanism problems are then explored in the first known demonstration of 3-D L-System topology optimization.

- Chapter 5 extends the work of Chapters 3-4 to consider the use of gradient-based sizing optimization to determine the optimal thicknesses of SPIDRS-generated structural members. The development of a hybrid optimization framework capable of coupling the topology and sizing optimization is also discussed, and the effects of where in the optimization process the sizing optimization scheme is employed are also considered.

- Chapter 6 provides a brief overview of potential applications of the SPIDRS algorithm, including tailorable stiffness structure design, thermomechanical design and optimization, and the design morphing supersonic airfoils.

- Chapter 7 summarized this work and offers thoughts an potential future research efforts.

2. TWO-DIMENSIONAL L-SYSTEM TOPOLOGY OPTIMIZATION USING GEOMETRY-BASED INTERPRETATION*

This chapter begins the dissertation by describing the development of a topology optimization framework using geometry-based interpretations of L-System encodings. To begin, Section 2.1 discusses the L-System formulation, geometry-based interpretation method, and how these tools are integrated into a structural optimization and design framework. Section 2.1 then demonstrates the performance of the proposed framework using three benchmark problems, where Pareto-optimal L-System designs are compared to known optimal/ideal solutions and, if necessary, attempts are made to understand any shortcomings this methodology might have. Finally, Section 2.3 details the validation of selected single-point designs using both high-fidelity continuum-based FEA models and experimental characterization of fabricated prototypes to ensure the proposed design methodology is capable of creating physically realizable structures.

2.1 L-System-Generated Topology Optimization

2.1.1 The L-System Approach

In 1968, biologist Aristid Lindenmayer developed a parallel rewriting system using a set of production rules to govern the evolution of a string of characters ($\boldsymbol{\omega}$). This method, now referred to as a Lindenmayer system (or L-System), was originally used to provide a mathematical description of the development of simple multicellular organisms, such as algae and fungi [86]. It has since been extended to formulate complex branching structures resembling large plants and other forms found in nature. Even as researchers have worked to advance the scientific theory behind them [87] and to understand the general mathematics of formal languages, L-Systems have found more uses, from creating computer imagery of plants [88, 89, 90, 91] to composing new music [92, 93]. However, the focus of this work is applying L-System theory to create engineering structures

*Portions reprinted with permission from “Development and Validation of a Genetic L-System Programming Framework for Topology Optimization of Multifunctional Structures” by Bielefeldt, B. R., Reich, G. W., Beran, P. S., and Hartl, D. J., 2019. *Computers & Structures*, 218, 152-169, Copyright 2019 by Elsevier.

resembling natural forms.

Generally, an L-System consists of [87]:

1. an *alphabet* of constant and/or variable characters (e.g., $\texttt{A}$, $\texttt{B}$, $\texttt{C}$, $\texttt{D}$, $\texttt{[}$, $*$, etc.) that is used to define the actions of the topology drawing algorithm (cf., Section 2.1.2) and/or assign material functionalities,

2. an *axiom*, which is a string $\boldsymbol{\omega}^0 = \omega_1\omega_2\ldots\omega_N$ of N variable characters and provides a foundation for the full topology string of a structure, and

3. a set of *production rules* applied first to the axiom $\boldsymbol{\omega}^0$ to create a new string or strings $\boldsymbol{\omega}^1$ and then recursively n times to any variable characters within the resulting string(s) $\boldsymbol{\omega}^n$, allowing for the generation of a set of instructions corresponding to potentially complex topologies.

The recursive nature of the L-System leads to self-similarity, and therefore fractal-like and branched forms are easily generated [88]. Plant models and other natural-looking forms are easily attained, as increasing the number of recursions causes the model to "grow" and generate a more complex self-similar structure.

As a short example, consider a simple L-System formulation as presented by Prusinkiewicz & Lindenmayer [87]. This consists of a binary alphabet $\boldsymbol{V} = \{\texttt{A}, \texttt{B}\}$, an axiom $\boldsymbol{\omega}^0 = \texttt{B}$, and production rules $\boldsymbol{P} = \{\texttt{A} \rightarrow \texttt{AB}, \texttt{B} \rightarrow \texttt{A}\}$. The successive application of the production rules produces a series of developmental stages defined by a string of characters. The zeroth stage is simply the axiom, or $\boldsymbol{\omega}^0 = \texttt{B}$. Subsequent stages are developed by simultaneously applying the production rules to each character in the string (as opposed to Chomsky grammars, which apply production rules sequentially [94]). Thus, the first stage $\boldsymbol{\omega}^1 = \texttt{A}$ is obtained by substituting $\texttt{B}$ with $\texttt{A}$ as defined by the production rules. Similarly, the second stage $\boldsymbol{\omega}^2 = \texttt{AB}$ is obtained by simultaneously substituting $\texttt{A} \rightarrow \texttt{AB}$ and $\texttt{B} \rightarrow \texttt{A}$. Examining the first six steps of this process, one finds:

$$\boldsymbol{\omega}^0 = \mathtt{B}$$

$$\boldsymbol{\omega}^1 = \mathtt{A}$$

$$\boldsymbol{\omega}^2 = \mathtt{AB}$$

$$\boldsymbol{\omega}^3 = \mathtt{ABA}$$

$$\boldsymbol{\omega}^4 = \mathtt{ABAAB}$$

$$\boldsymbol{\omega}^5 = \mathtt{ABAABABA}$$

$$\boldsymbol{\omega}^6 = \mathtt{ABAABABAABAAB}$$

The result is clearly a string of characters and does not have any inherent geometrical representation; thus, the resulting string must be paired with some sort of graphical approach to serve the goals of this effort [89]. This will be addressed in Section 2.1.2.

In the context of this work, we consider an alphabet that consists of the following: $i)$ a set $\boldsymbol{\alpha}$ of four *variable* (i.e., subject to production rules) letter characters $\boldsymbol{\alpha} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$ describing structural branch length and/or material assignment (functionality), $ii)$ two *constant* (i.e., not subject to production rules) characters $\{\mathtt{-}, \mathtt{+}\}$, which indicate changes in angle as line segments are drawn, and $iii)$ two constant characters $\{\mathtt{[}, \mathtt{]}\}$ that signify the beginning and end of individual branches. All other characters in the alphabet shown are inactive and reserved for future iterations of this L-System formulation. Additionally, an axiom consisting of two *variable* letter characters ($\boldsymbol{\omega}^0 = \omega_1\omega_2,\ \omega_i \in \boldsymbol{\alpha}$) will be used to initiate the recursive development of a fully compiled string of instructions.

To develop a set of production rules $\boldsymbol{P}$, we consider a *context-free* approach, meaning that the result of a given production rule as applied to its associated variable character is independent of any neighboring variable characters (i.e., $\alpha_i \rightarrow P_i$)[1]. Since each variable character requires its own

[1] Alternatively, a *context-sensitive* approach means that the result of a given production rule as applied to its associated variable is dependent on the variables both before and after it (i.e., $\alpha_l\langle\alpha_i\rangle\alpha_r \rightarrow P_i$). See [6] for more information.

single production rule (i.e., it is deterministic, see [87, 6]), this L-System formulation considers four production rules for the four variable letter characters, each consisting of 14 characters, which can be generally written as

$$P_i : \alpha_i \rightarrow \lambda_1^i \lambda_2^i \ldots \lambda_{14}^i, \;\; i = 1 \ldots 4$$

for all $\alpha_i \in \boldsymbol{\alpha}$, where $\lambda_1^i \lambda_2^i \ldots \lambda_{14}^i$ is a string of 14 characters such that $\lambda_K^i \in \Lambda_K$ and[2]

$$\Lambda_1 = \{\,\texttt{[}\,,\_\}\,,\;\; \Lambda_2 = \{\texttt{-},\_,\texttt{+}\}\,,\;\; \Lambda_3 = \{\ast,\_,/\}\,,\;\; \Lambda_4 = \{\texttt{\#},\_,\texttt{!}\}\,,\;\; \Lambda_5 = \boldsymbol{\alpha},\;\; \Lambda_6 = \boldsymbol{\alpha},$$
$$\Lambda_8 = \{\,\texttt{[}\,,\_\}\,,\;\; \Lambda_9 = \{\texttt{-},\_,\texttt{+}\}\,,\;\; \Lambda_{10} = \{\ast,\_,/\}\,,\;\; \Lambda_{11} = \{\texttt{\#},\_,\texttt{!}\}\,,\;\; \Lambda_{12} = \boldsymbol{\alpha},\;\; \Lambda_{13} = \boldsymbol{\alpha}.$$

Note that assignments for λ_7^i and λ_{14}^i are made such that the "begin branch" indicators "[" in λ_1^i and λ_8^i, if assigned, are closed. Thus, only 12 independent variables are needed to form each production rule. The characters $\star$, /, #, and ! are considered inactive in the current implementation, but could be assigned functionalities for future studies.

The recursive generation of final strings using an L-System approach is illustrated in Figure 2.1.

2.1.2 Geometry-Based Interpretation

The L-System formulation presented in Section 2.1.1 is effective in encapsulating great complexity (e.g., design diversity) into a relatively straightforward encoding. The user-specified number of iterations over which recursive operations are performed allows for further tuning of the information generated. However, as previously mentioned, the L-System must be coupled with a graphical interpreter for the encoded information to be given spatial significance.

Turtle graphics is a vector-based graphical method by which encoded instructions are interpreted to construct paths [75], which here are continuous and branched in 2-D space for simplicity. The "turtle" is an agent for constructing straight line segments based on its movement in this 2-D

[2] Note that the *underscore* character "_" here is meant to represent null string entries that are not interpreted by the graphing algorithm (and are therefore not considered characters in the alphabet). In practice, they are ignored completely and are included here for typesetting purposes only.

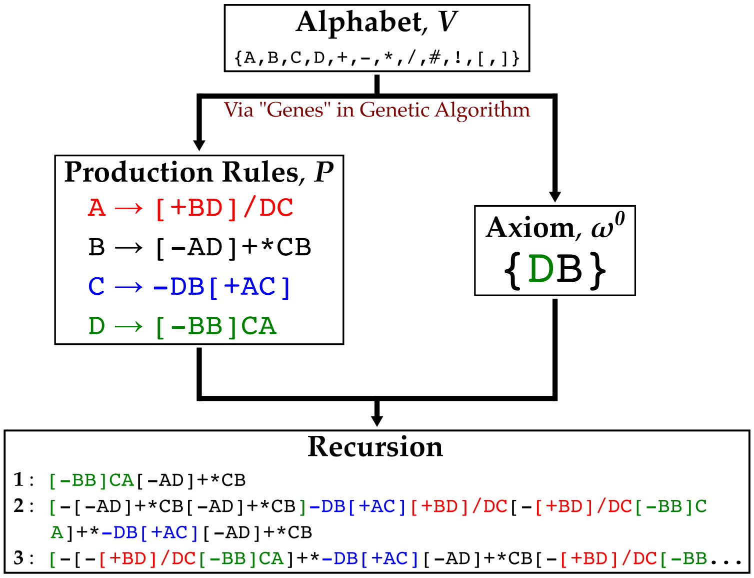

Alphabet, *V*
{A,B,C,D,+,-,*,/,#,!,[,]}
Via "Genes" in Genetic Algorithm
Production Rules, *P*
A → [+BD]/DC
B → [-AD]+*CB
C → -DB[+AC]
D → [-BB]CA
Axiom, ω^0
{DB}
Recursion
1: [-BB]CA[-AD]+*CB
2: [-[-AD]+*CB[-AD]+*CB]-DB[+AC][+BD]/DC[-[+BD]/DC[-BB]C
A]+*-DB[+AC][-AD]+*CB
3: [-[-[+BD]/DC[-BB]CA]+*-DB[+AC][-AD]+*CB[-[+BD]/DC[-BB...

Figure 2.1: Example of recursively generated final strings using an L-System approach consisting of a two-character axiom and production rules constructed with a specific alphabet. Note that A, B, C, and D represent variable characters.

space, and the state of the turtle is characterized by its current position (a set of Cartesian coordinates), its orientation (the direction it will next move), and its line drawing status ("pen up" or "pen down") [89]. Instructions interpreted by the turtle function are then i) move forward/backward, ii) turn by some angle, iii) pick up/put down pen, and iv) go to some point in 2-D space. These instructions then serve to control the motion (translational, rotational, and instantaneous placement) of the turtle in creating two-dimensional branched paths.

In particular, the translational instructions to the turtle are encoded in the variable characters $\boldsymbol{\alpha} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$, and rotational instructions (i.e., angular changes to the orientation of the turtle) are made incrementally by + (clockwise) and - (counter-clockwise) characters. Branches are created by interpreting [and] characters as instructions to "save current position" and "pick up pen; return to last saved position; put down pen", respectively. Note that branch instructions can be nested. Simple example branched topologies constructed using the L-System example in Figure 2.1 are depicted as interpreted by the turtle function in Figure 2.2. The colors associated with each variable character (and by extension each segment) represent example material assignments or functionalities that will be explained in the studies to follow. As mentioned in Section 2.1.1, increasing the number of recursions leads to progressively more complex topologies. Note that the structural utility of these merely demonstrative forms is unknown and that a large number of free segment ends result from this approach.

2.1.3 Structural Analysis and Design Framework

While the L-System framework and associated structural topologies presented in Section 2.1 have inspired research and developments in a number of areas, the main goal of this work is their use toward the topology optimization of mechanically driven multifunctional structures. This is achieved by coupling the L-System description and turtle graphics interpreter to a finite element analysis (FEA) package and genetic algorithm for the purpose of driving populations of structures toward designs that can simultaneously satisfy multiple design goals associated with multiple functions. In this way, truly bio-inspired designs can be achieved, as any biological entity is driven by evolution towards designs that meet certain criteria. Regarding the genetic optimization and recalling the L-System formulation discussed in Section 2.1.1, the following design variables are

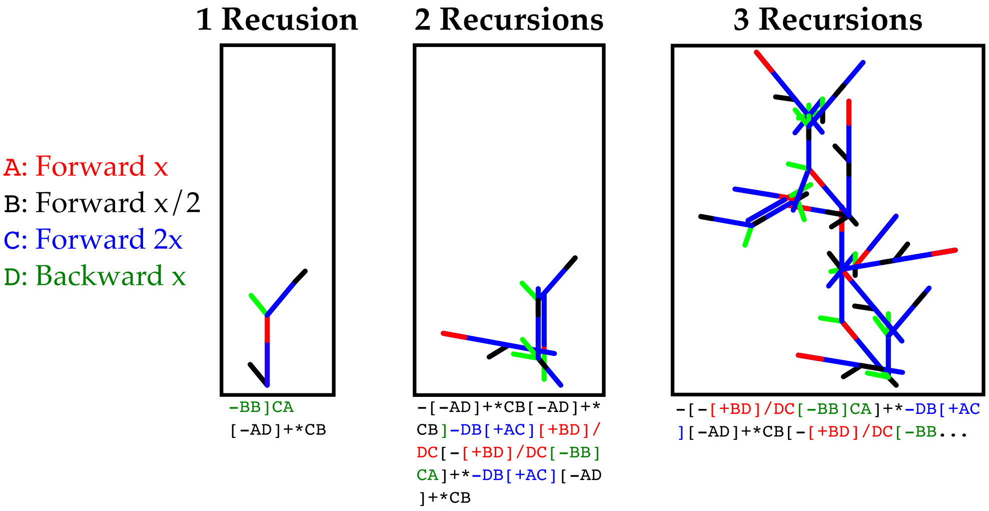


Figure 2.2: Example of branched structure generation using the L-System example in Figure 2.1. Branch length assignments are as defined in the figure, and all incremental angle changes are 40° or integer multiples thereof.

included in the set of properties that comprise the genotype:

1. the two variable characters that form the axiom $\omega_1\omega_2$ (2 discrete variables),
2. the assignments for each of the 12 *independent* characters $\lambda_k^i,\ i = 1 \ldots 14$ associated with each of the four production rules $P_i,\ i = 1 \ldots 4$ (48 discrete variables), and
3. globally applied turtle graphics length factor and angle increment definitions (2 continuous variables),

yielding a total of 52 independent design variables. Given the four variable characters in $\boldsymbol{\alpha}$ and the assigned definitions of the 14 variables (12 independent) in $\boldsymbol{\Lambda}_k$ from which the production rules are constructed, it can be shown that the 50 design parameters associated with the axiom and four production rules (i.e., not including segment length parameter and angle assignments) are sufficient to describe $\approx 5 \times 10^{24}$ different L-System generated topologies of varying complexity[3].

In this work, an open-source Python toolbox known as the *Distributed Evolutionary Algorithms in Python* (DEAP) [95] is chosen as the genetic algorithm implementation and manages the simulation process. During this initial study, the well-known NSGA-II algorithm for multiobjective optimization [68] is considered, although DEAP's modular approach to deploying a wide range of evolutionary algorithms allows for the use of other algorithms in future work. As NSGA-II is generally a quantitative algorithm considering continuous variables, each of the fifty discrete variable characters introduced in Section 2.1.1 are mapped evenly onto a real number line in the semi-open interval [-1,1)[4]. The two continuous variables are likewise mapped to the same interval via consideration of their upper and lower bounds.

[3] This value does not account for potentially identical designs generated after the graphical interpretation of the L-System and thus may not represent the number of *unique* topologies.

[4] Consider for example the continuous variable $-1.0 \leq v_1 < 1.0$ employed by the genetic algorithm to define the first character of the axiom $\omega_1\omega_2$ (i.e., the first variable of the L-System). Given the four possible characters of $\boldsymbol{\alpha}$, we have

$$\omega_1 = \begin{cases} A: & -1.0 \leq v_1 < -0.5 \\ B: & -0.5 \leq v_1 < 0.0 \\ C: & 0.0 \leq v_1 < 0.5 \\ D: & 0.5 \leq v_1 < 1.0 \end{cases}.$$

The structural analysis method used in the optimization process requires an effective and flexible approach capable of potentially handling complex loadings, nonlinear material responses (i.e., buckling), and additional physical effects. Therefore, each branched structure generated is converted via an Abaqus FEA suite [96] pre-processor into a *frame* structure consisting of 2-D beam elements based on the topology associated with that structure as discussed in Section 2.1.2. Several additional considerations govern the generation of these 2-D structures, such as boundary constraints and the trimming of free-ended segments, can be found in Appendix A-B. The section assignment associated with each segment (i.e., material assignment and cross-sectional size and shape) is defined by the predetermined interpretation of the characters in α. During analysis, an uncommon but efficient parallelization approach is taken whereby the ability of Abaqus to analyze assemblies of multiple parts is leveraged and all members of a generation are analyzed simultaneously in a single "assembly" rather than individually. After analysis is completed, an Abaqus post-processor evaluates the multiple objective functions (e.g., weight, stiffness, structural actuation characteristics, etc.). Any designs in violation of a constraint (e.g., the local stress exceeding an acceptable maximum) are penalized via assignment of artificially poor values for each objective function.

The full multiobjective optimization framework is shown in Figure 2.3, where the data flow illustrates how an entire population of potential designs in each generation is arrayed, passed into the FEA solver as a single set, solved simultaneously, and post-processed in the same manner. Note that while this work considers only mechanically driven designs, it is a goal to extend this framework to multi-physical design studies, such as fluid-structural interactions as has been addressed in other works [97, 70].

2.2 Design Optimization Examples

To demonstrate the ability of the L-System genetic design approach (cf., Figure 2.3) in determining effective configurations for muscular-skeletal multifunctional structures, three design problems of increasing complexity are presented. The first problem addresses only structural (i.e., skeletal) functionality in the form of a cantilevered frame configured to carry a static transverse

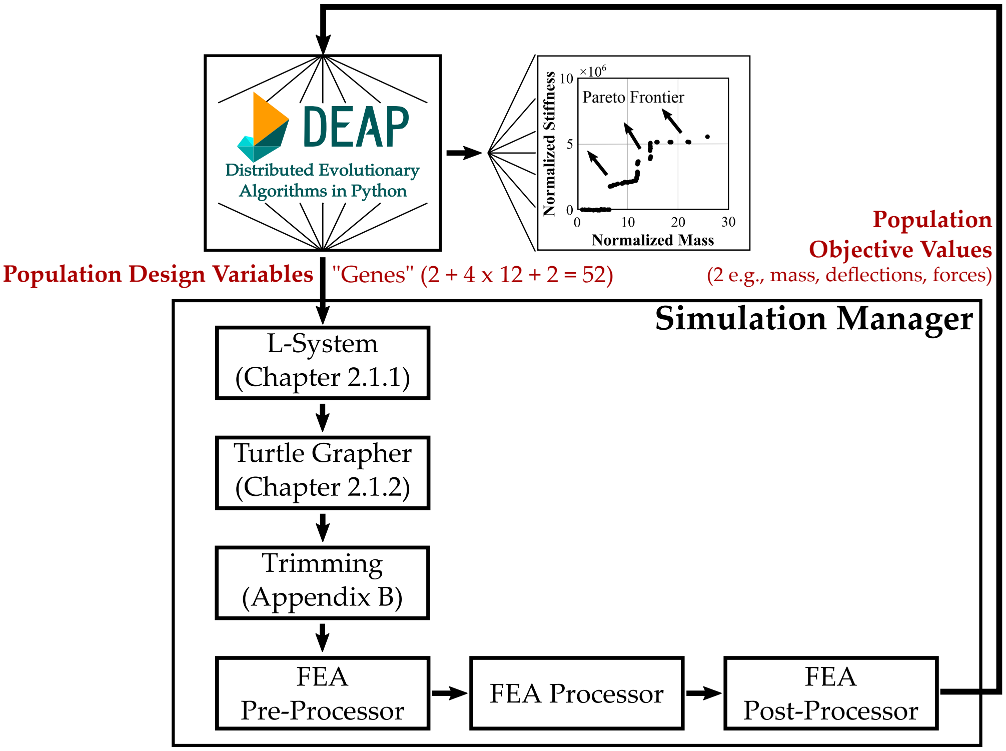

DEAP
Distributed Evolutionary Algorithms in Python
Normalized Stiffness
×10⁶
10
5
0
Pareto Frontier
0
10
20
30
Normalized Mass
Population Objective Values
(2 e.g., mass, deflections, forces)
Population Design Variables
"Genes" (2 + 4 x 12 + 2 = 52)
Simulation Manager
L-System (Chapter 2.1.1)
Turtle Grapher (Chapter 2.1.2)
Trimming (Appendix B)
FEA Pre-Processor
FEA Processor
FEA Post-Processor

Figure 2.3: Flowchart indicating the overall genetic programming topology optimization framework for multifunctional structures (low mass to high stiffness represent example objectives). Note that the entire current population chosen by the genetic algorithm is assessed simultaneously using analysis parallelization.

load. The second study considers both structural and compliant regions (i.e., skeletal and cartilage-like functionalities) for the design of an inverting compliant mechanism [98] of the same kind used in demonstrating alternative topology optimization schemes [2]. The third, final, and least intuitive design study also utilizes structural and compliant regions to create a mechanical motion rectifier capable of converting a bidirectional input into a unidirectional output [99]. For each problem the structural design approach is summarized, the optimization problem is quantified, and results are presented in the form of Pareto frontiers and associated example design configurations.

Experimentally measured material properties for the skeletal and compliant regions considered in the design analyses analyses are shown in Table 2.1. These materials are associated with the Stratasys Objet 500 “PolyJet” 3-D printer, which is employed in this work for the experimental validation of predicted optimized design performance (cf., Section 2.3.2). VeroWhite is a stiff photopolymer associated with skeletal-like behavior, while TangoBlack is a rubber-like photopolymer that represents analogously the flexibility of cartilage. The properties shown in Table 2.1 are based on material characterization experiments conducted at the Facility for Innovative Research in Structures Technology (FIRST) laboratory at the Air Force Research Laboratory.

For each problem, structures are created based on the L-System representation of branched topologies as outlined in Section 2.1. In particular, these branched structures originate from a predetermined position (or positions) in the 2-D structure and spread throughout predetermined bounded domains. It is important to note that for each design problem, the determination of domain boundaries, branch roots (i.e., locations where the L-System string originates from), and any initial structure is critical for successful design space exploration. In particular, it is necessary to ensure that structural analysis can be completed and meaningful structural responses can be calculated, regardless of the quality of the design. In displacement-driven FEA, this generally depends on the existence of a load path between points on which displacement and/or force boundary conditions are applied, regardless of the predetermined structural domain or branch roots. This condition can be generally guaranteed if a default initial structure is generated around the design boundary and further if branch roots are predetermined to exist at points of boundary condition application. Previous work using L-Systems to define cellular topologies [74] likewise satisfied these conditions for all designs by assuming structural domains were bounded by load-bearing structural members. This work makes a similar assumption regarding initial designs, herein referred to as “null” designs as they represent the full extent of the structure in cases where the L-System corresponds to no

Table 2.1: Experimentally measured properties associated with the skeletal- and cartilage-like materials in subsequent analyses.

Material	Density (kg/m^3)	Modulus (MPa)	Poisson’s Ratio
VeroWhite	1175	1500	0.40
TangoBlack	1125	0.34	0.45

generated internal structure. Segments that would result in the violation of these boundaries are restricted so as to intersect the boundaries but not pass through them. The turtle interpreter is then only permitted to construct new segments if such segments are interior to the boundaries. More details on the boundary constraint algorithm can be found in Appendix A. For each problem, the initiation point(s) (or "roots") of the branched structures and predetermined structural boundaries are clearly illustrated.

2.2.1 Cantilevered Frame

The first example problem considers a simple cantilevered frame composed of beam elements. Figure 2.4 illustrates the initial structure, domain boundary, branch root location, and boundary conditions associated with this problem. Note that a $y - z$ symmetry condition is enforced along the x-axis, allowing for the consideration of only half of the structure. The initial (null) structure consists of compliant beams[5] that are roughly five orders of magnitude more compliant than generated internal segments (cf., Table 2.1), all with an assumed cross-section of $0.2\,\text{mm} \times 20\,\text{mm}$; this initial structure also serves as the domain boundary for each design considered. Additional boundaries are placed along the y-axis of the structure and along the axis of symmetry to enclose the domain. As dictated by the boundary, the frame has a maximum length of 300 mm and maximum total height of 350 mm. A branch root is placed 50 mm above the axis of symmetry at the minimum x position, and each line segment generated corresponds to a stiff beam element with a cross-section of 2 mm × 20 mm. A displacement boundary condition of $u_x = u_y = 0$ is enforced at the branch roots, and a concentrated force of $f_y = 0.05\,\text{N}$ is applied to the tip of the frame. Note that the placements of boundary conditions and/or branch root locations are arbitrary and can be adjusted depending on the problem. Furthermore, one could potentially add these locations to the design problem by introducing additional design variables using a method that will be discussed in Section 3.5.1.

Details regarding the multiobjective design optimization of a structural frame are shown in

[5] Note that these compliant regions are irrelevant as load-bearing members if any internally connected segments are generated. The absence of connected internal structure results in a low-mass, low-stiffness design that serves as the baseline for this design study.

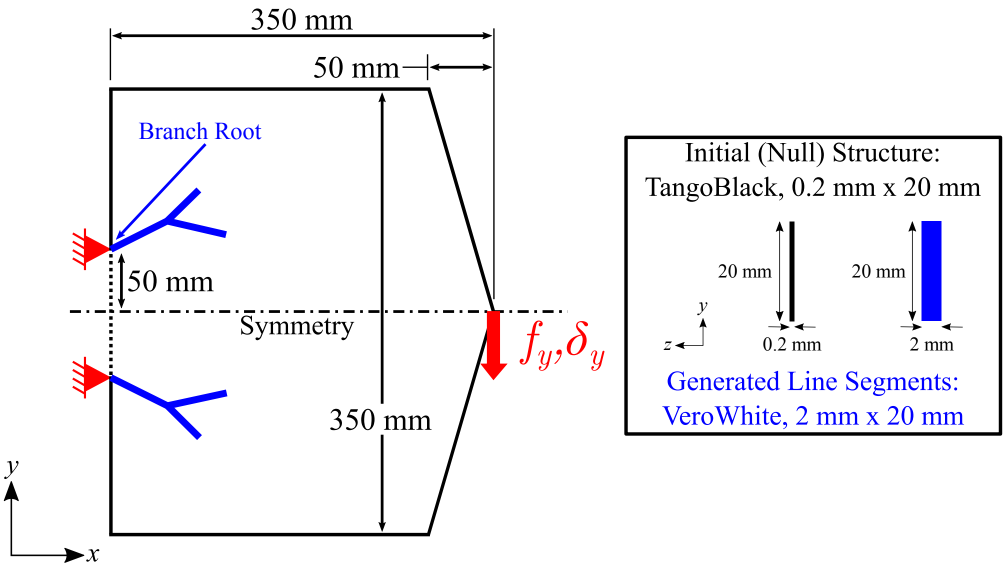


Figure 2.4: Initial (null) structure, domain boundary, branch root location, and boundary conditions associated with the frame design study.

Table 2.2. The objective of reduced mass favors designs with a sparse topology, while the objective of increased stiffness tends to create dense topologies. As discussed in Section 2.1.3, the L-System implementation employed herein requires 52 independent design variables. The segment length parameter x is bounded between 30-150 mm, and the segment turn angle is bounded between 18-45 degrees. Note that each letter character in the alphabet is assigned the same material and non-variable cross-section and thus only defines the associated segment length parameter. Considering the small load and assuming small deflections for meaningful designs, a fully linear FEA with an average element length of approximately 30 mm is employed and no structural constraints are considered. Due to the speed of such analysis as well as the efficient parallelization approach in which all members of a given population are analyzed simultaneously (cf., Figure 2.3), $100 \times 1,000 = 100,000$ designs can be analyzed in only a few hours.

The Pareto frontier associated with the multiobjective optimization described in Table 2.2 is shown in Figure 2.5, along with three example Pareto-optimal topologies found by the optimizer. The mass and stiffness of each design is normalized by the derived properties of the "null" design (i.e., made up of only the initial structure), which is the lightest and least stiff design possible. The

Table 2.2: Specifications for the multiobjective topology optimization problem of a light and stiff structural frame using a geometry-based interpretation of L-System encodings.

Design Problem Statement	
Minimize (Maximize):	normalized mass (normalized stiffness)
by varying:	2 axiom characters,
	4 rule assignments (12 genes each),
	segment length parameter x, segment turn angle θ
subject to:	no constraints
Alphabet Definitions (Section & Length)	
A	VeroWhite, 2 mm×20 mm, Fwd x
B	VeroWhite, 2 mm×20 mm, Fwd $x/2$
C	VeroWhite, 2 mm×20 mm, Fwd $x/3$
D	VeroWhite, 2 mm×20 mm, Fwd $x/5$
NSGA-II Parameters [68]	
100 members for 1,000 generations,	
P_{cross} = 0.9, η_{cross} = 20,	
P_{mut} = 1/52, η_{mut} = 20	

stiffest and heaviest design features a dense distribution of branches. For further assessment, we consider the utopia point in the objective space [6]. A particular topology of interest (herein referred to as the fabricated design) would then be the design on the frontier closest to the utopia point (normalized mass of 20.98, normalized stiffness of 5.72E6). This design will be further analyzed in Section 2.3.

Also depicted in Figure 2.5 is the frontier associated with a series of Michell trusses (as defined in [100]), which have been shown mathematically to be optimal in the sense of maximizing stiffness per unit mass [101]. An illustration of a Michell truss geometry is shown in Figure 2.6. These structures consist of beam elements with the same material properties and cross-sections as described above. It is important to acknowledge that despite analyzing 100,000 topologies, the L-System framework is unable to match the performance of the Michell truss; in fact, at heavier masses L-Systems designs are out-performed by a factor of two. This includes the design nearest to the utopia point, which has several topological features in common with a Michell truss.

[6] defined as the point that maximizes or minimizes all objective functions (normalized mass of 1.0, normalized stiffness of 8.01E6)

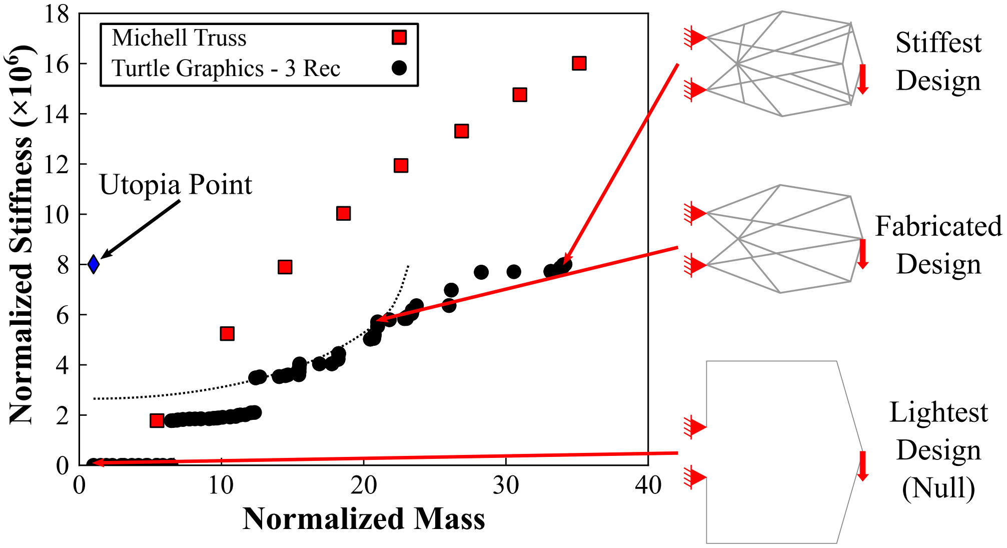


Figure 2.5: Pareto frontier of the structural frame design problem after 1,000 generations compared with Michell truss (i.e., known optimal) solutions, along with the utopia point for the L-System optimization frontier. Dashed line represents points of constant distance from this utopia point.

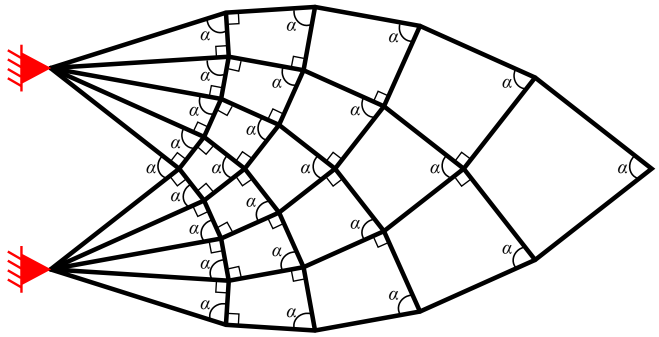


Figure 2.6: Illustration of Michell truss geometry, which is comprised of members that intersect each other at right angles as indicated in the figure. Here it is assumed that the members are equally spaced such that select angles are equivalent and equal to α as defined in [100].

Figure 2.7 compares the results of the L-System optimization with frame structures obtained from MATLAB implementations of both the SIMP [18] and level set [102] topology optimization methods (cf., Chapter 1). These implementations consider a design problem identical to that shown in Figure 2.4 and material properties consistent with those given in Table 2.1. Comparison of

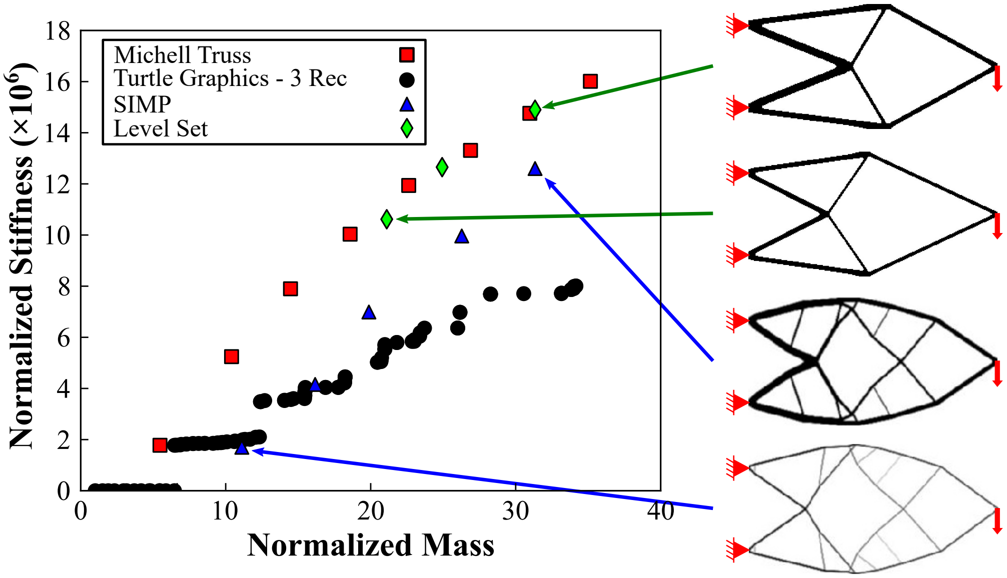


Figure 2.7: Comparison of L-System optimization and Michell truss Pareto frontiers to those generated using both SIMP and level set topology optimization methodologies, along with example topologies generated using the SIMP and level set methods.

these two methods to the proposed L-System method yields several interesting results. As shown by the example topologies in Figure 2.7, for different volume fractions both the SIMP and level set methods converge to a common topology, with higher volume fractions achieved by simply thickening select members of the structure. The thickening of these members consistently leads to an improvement in structural performance, indicating that the Pareto frontier associated with the L-System optimization could be improved if structural members were allowed to have varying cross-sections. Also note that at lower volume fractions and therefore more comparable member sizes, topologies generated using the L-System framework compare well with those obtained using the SIMP method. Topologies obtained using the level set method perform markedly better and seem to converge to the two-bar Michell truss solution. However, this level set implementation fails to converge at lower volume fractions, indicating that it is not capable of representing long slender structural members with the current mesh refinement.

2.2.1.1 Limitations of Geometry-Based Interpretation

The results in Figure 2.5 call into question the ability of an L-System description to capture the performance and indeed, the topology, of a Michell truss. It will be shown that this limitation is offset by the highly successful multi-objective design studies of Sections 2.2.2 and 2.2.3, but it is worth addressing here in the context of simple linear elastic (e.g., static structural) problems. To begin, we consider the "three-bar" Michell truss shown in Figure 2.8 (left) in more detail. Accounting for symmetry, it was found by iterative exploration that the topology of this structure can be obtained using the following L-System instructions (cf., Section 2.1):

```
[A - - - B - A][- A + A - - - A][- - A + A + A A],
```

resulting in the form in Figure 2.8 (right). Note that these represent the full L-System instructions (i.e., after recursion) based on the same alphabet and associated functionalities as given in Table 2.2 as well as line segment trimming (cf., Appendix B). Given appropriate segment length and turn angle parameters, these instructions generate a structure with a slightly different geometry but identical topology to that of a Michell truss. This L-System Michell truss structure corresponds to a normalized mass of 15.41 and a normalized stiffness of 7.36E5, meaning that it is unable to match the performance of both the Michell truss and L-System optimization.

To better understand why the L-System is generally outperformed, consider the stress field comparisons between the Michell truss and both the L-System Michell truss (Figure 2.9a) and L-System optimized frame (Figure 2.9b). The stress field on the top surface of each beam element in each structure is associated with a tip load of $f = 0.05\,\mathrm{N}$ and deformation is scaled by a factor of 5,000 to exaggerate the behavior of each structure. As expected from a mathematically ideal solution, the Mises stress of the Michell truss is well-distributed, purely axial, and thus relatively low. However, there are noticeable stress concentrations in the L-System Michell truss, particularly near points of bi-linear segment junctions with angle changes that are not supported by another line segment. This causes segments to bend, leading to larger deformations and stress concentrations.

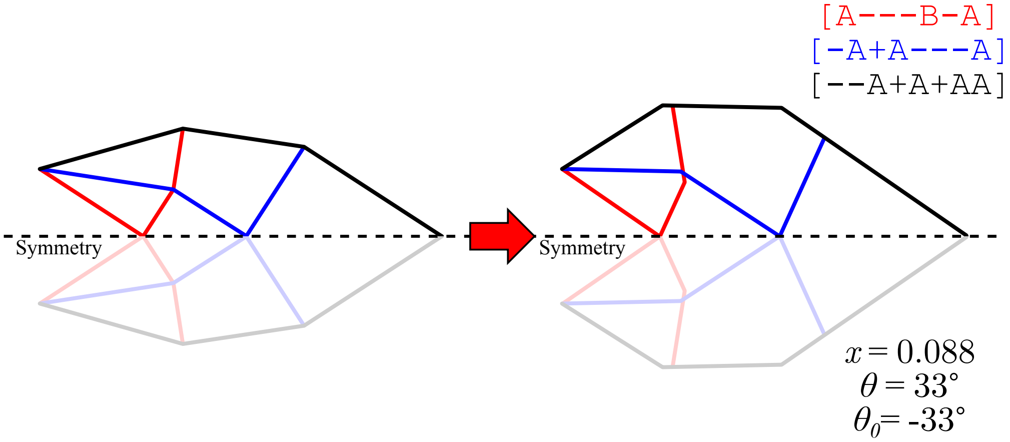
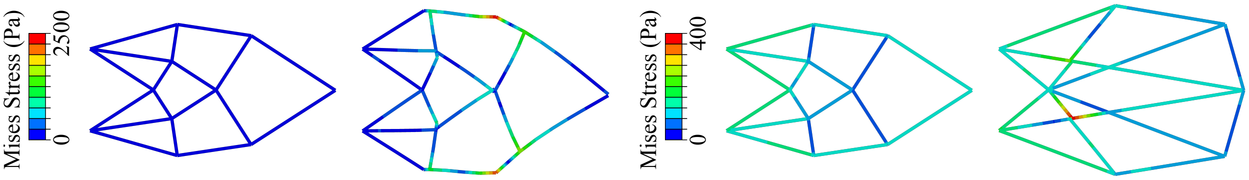


Figure 2.8: Illustration of a three-bar Michell truss geometry and associated L-System instructions that create a topologically identical structure.

Compare this with the L-System optimized frame, which has a more comparable stress distribution to the Michell truss but still features stress concentrations and has a lower associated stiffness. This result confirms a correlation between stiffness criterion and distributed stress criterion in structural optimization as shown by Li et al [103]. We also conclude that this L-System framework is not guaranteed to produce trusses (i.e., structures with non-bending members) for the given problem, and that topological similarity to a known ideal solution in no way guarantees good performance.

The inability of this L-System formulation in general to match the performance of a traditional



(a) L-System Michell truss ($x = 0.088$, $\theta = 33°$, $\theta_0 = -33°$).

(b) L-System optimized frame.

Figure 2.9: Comparison of Mises stress field and deformation for a Michell truss and both the L-System Michell truss and L-System optimized frame associated with a tip load of $f = 0.05\,\mathrm{N}$ (deformation scale factor of 5,000; line segments thickened to better illustrate stress fields).

Michell truss can be attributed to the restrictions placed upon possible segment length parameters and segment turn angles. Notice in Table 2.2 that there is a single variable for both segment length parameter x and segment turn angle θ, and that the potential segment length parameters assigned to each character in the alphabet are simply a function of x. Thus, there are a finite number of discrete lengths a given branch can have and a finite number of angles a given segment can turn through. Figure 2.9 illustrates the importance of having line segments intersect at segment turns for overall structural stiffness; even with 5×10^{24} design possibilities, it does not appear possible to capture that same behavior given a finite number of segment lengths and turn angles. However, the current formulation does develop non-intuitive designs that are effective, even when compared to a mathematically rigorous benchmark. This will be exemplified in Sections 2.2.2 and 2.2.3.

2.2.2 Compliant Mechanism: Tensile Inverter

The second design problem is more applicable to the goals of this work and considers a compliant mechanism, specifically a *tensile inverter*, composed of materials that exhibit either a stiff or flexible response corresponding to the assignment of bone-like or cartilage-like functionalities, respectively, in muscular-skeletal structures. The initial structure, domain boundary, branch root location, and boundary conditions associated with this mechanical problem are illustrated in Figure 2.10. The initial structure consists of both compliant and stiff beam segments with an assumed rectangular cross-section of 2 mm $\times$ 10 mm; beams A and D in this initial structure also serve as part of the domain boundary. Beams B and C are necessary to ensure the existence of a load path but their bent form and soft material make them negligible in meaningful designs; as such, these four segments are not considered during post-processing.

As in Section 2.2.1, a symmetry condition is enforced along the $y = 0$ plane; this plane and a second plane offset 100 mm above also serve as boundaries and enclose the domain within which branched structures can be grown. Note that, unlike the frame design of Section 2.2.1, *two* branched structures are considered simultaneously (before accounting for symmetry). The two branch roots each start from one of the two boundary conditions on the left side of the structure. This *multi-branch* approach is implemented by splitting the two-character axiom into two discrete

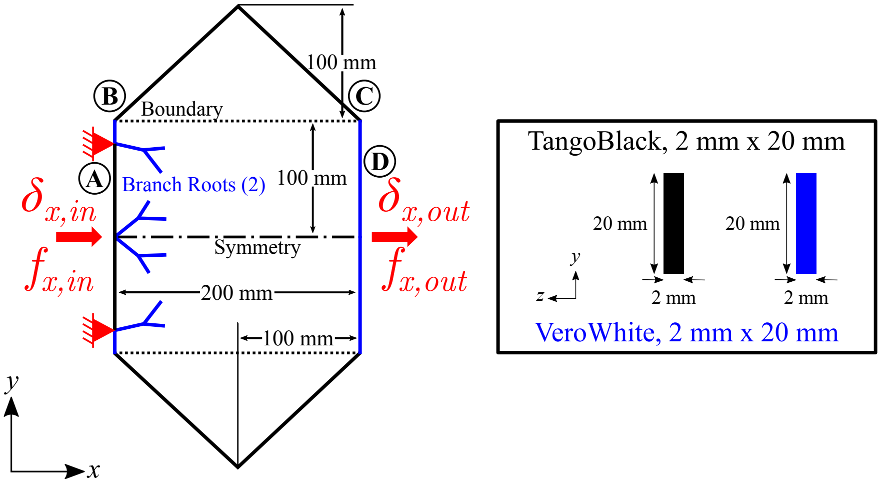


Figure 2.10: Initial structure, domain boundary, branch root location, and boundary conditions associated with the tensile inverter design study.

strings, which are each recursively and independently grown as discussed in Section 2.1.1 and then interpreted as separate, though often intersecting, branched structures. A displacement boundary condition of $u_x = \delta_{x,in}$ is applied to the lower branch root, while a displacement boundary condition of $u_x = u_y = 0$ is applied to the upper branch root.

The goal of a well-designed tensile inverter is to maximize both the positive output displacement in the x-direction and positive output force in x-direction for given associated negative inputs. Thus, two independent structural analyses are considered in the assessment of each design. In the first analysis a displacement is applied to the input node ($\delta_{x,in} = -5\,\text{mm}$) while the resulting output displacement $\delta_{x,out}$ is measured when no load is present, providing an indication of *free displacement inversion*. In the second analysis, the output node is forced back to its reference position ($\delta_{x,out} = 0\,\text{mm}$) and the reaction forces at both the input and output points ($f_{x,in}$ and $f_{x,out}$) are recorded, providing an indication of the *blocked force inversion*. The analyses of both steps consider the full geometric nonlinearity of the problem (e.g., internal buckling, large rotations, etc.). Given the description of loading conditions associated with Figure 2.10, the free displacement inversion ratio and blocked force inversion ratio are defined respectively as

$$r_\delta = \left.\frac{\delta_{x,out}}{\delta_{x,in}}\right|_{f_{x,out}=0} \quad \text{and} \quad r_f = \left.\frac{-f_{x,out}}{f_{x,in}}\right|_{\delta_{x,out}=0}. \tag{2.1}$$

Engineering goals for inverter design might then include the absolute minimization of both quantities (i.e., the generation of large negative values for r_δ and r_f), among other objectives such as mass minimization.

It is well known from other topological optimizations of compliant mechanisms that large displacement inversion ratios are achieved through large internal rotations/translations of stiff regions [2], where all internal motions can result only from highly localized material deformation (i.e., localized bending). In this study, the alternate assignment of VeroWhite sections to some segments and TangoBlack sections to others is shown in the following to allow for most deformations to be isolated to the more compliant TangoBlack regions. However, to ensure the structural feasibility of potential tensile inverter designs, a stress constraint in VeroWhite members of σ^{max}_{Mises}=33.8 MPa is imposed via the penalty method (Section 2.1.3).

Details regarding the multiobjective design optimization problem associated with the tensile inverter mechanisms are specified in Table 2.3, where segment length parameter x is bounded from 10-50 mm and segment turn angle θ is bounded from 10-40 degrees. Note that an *a priori* understanding of the need to localize deformation dictates that, on average, compliant members should be shorter than stiff ones; this is reflected in the assignment of alphabet characters to section functionalities. As before, the FEA model consists of elements with an average length of approximately 30 mm. Due to the nonlinear behavior of the mechanism, and in particular due to the effects of local buckling, implicit dynamic analysis was required and loading increments during analysis were often small. This substantially slowed the analysis process such that the total number of designs considered was necessarily reduced to $60 \times 400 = 24,000$ based on previous work [85].

The results of the multiobjective design optimization are shown in Figure 2.11. Regarding

Table 2.3: Specification of the multiobjective topology optimization problem for tensile compliant inverter mechanism.

Design Problem Statement	
Maximize:	r_δ, r_f
by varying:	2 axiom characters, 4 rule assignments (12 genes each), segment length parameter x, segment turn angle θ
subject to:	$\sigma_{Mises}^{max} \leq 33.8\,MPa$
Variable Character Definitions (Section & Length)	
`A,B`	VeroWhite, 2 mm×10 mm, Fwd x
`C,D`	TangoBlack, 2 mm×10 mm, Fwd $x/2$
NSGA-II Parameters [68]	
	60 members for 400 generations, P_{cross} = 0.9, η_{cross} = 20, P_{mut} = 1/52, η_{mut} = 20

the Pareto frontier associated with the displacement/force inversion maximization problem, an approximate ideal solution that nominally bounds the problem can easily be obtained. Assuming that the response of the mechanism is linear and is used to displace a load of constant magnitude, the ratio of input to output work could be calculated as

$$\frac{W_{out}}{W_{in}} \approx \frac{f_{x,out}\delta_{x,out}}{f_{x,in}\delta_{x,in}}. \tag{2.2}$$

Considering the ideal case of $W_{out}/W_{in} = 1$ and substituting into Equation 2.1 leads to the simple ideal frontier associated with energy conservation

$$r_\delta = \frac{1}{r_f}. \tag{2.3}$$

As seen in Figure 2.11, the Pareto frontier obtained using this topology optimization approach conforms to this ideal solution curve but does not reach it as all compliant mechanism designs require the conversion of some amount of the external work applied into internally stored strain energy. Thus, the L-System framework is adept at creating topologies that are suitable for tensile

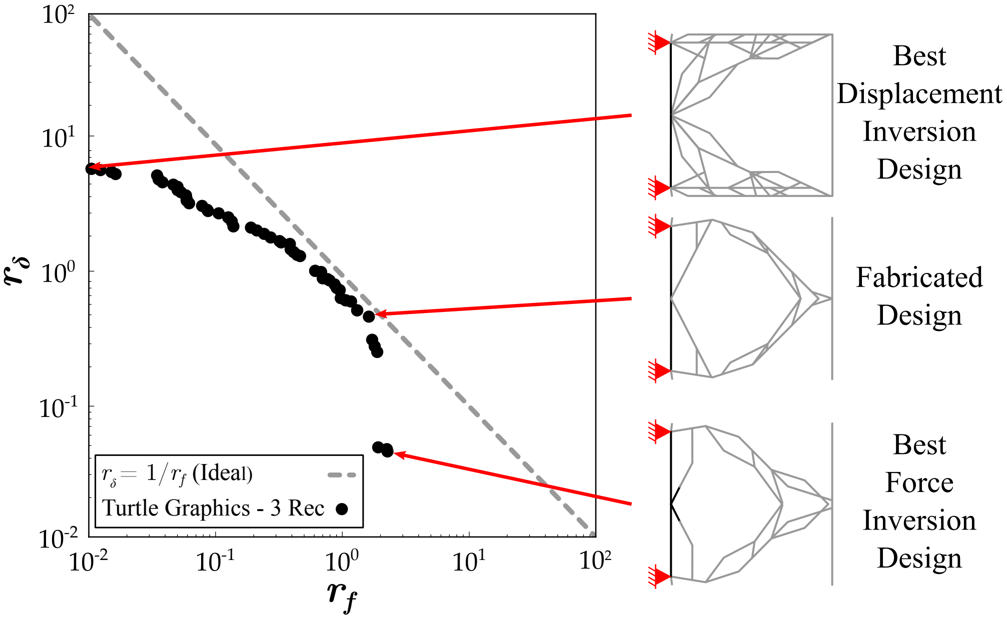
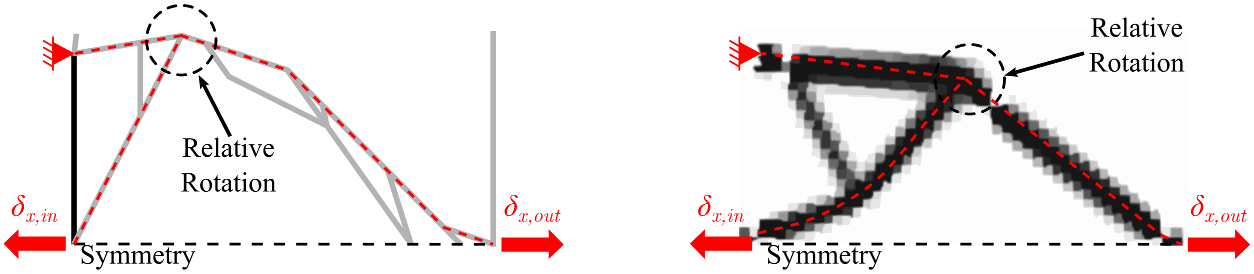


Figure 2.11: Pareto frontier of the tensile inverter design problem after 400 generations compared with an approximate analytically-derived frontier. Null structure removed for clarity.

inverters and does so in a manner that populates a full Pareto frontier. For further analysis, the single design closest to the ideal frontier is selected as shown in Figure 2.11 (r_δ of 0.484, r_f of 1.620).

Further investigation into this selected design yields several non-intuitive findings that serve to highlight the strength of this approach. First, despite allowing for the possibility of a dual-material structure (which will factor heavily in the following subsection on mechanical motion rectification), the only TangoBlack in the selected design is located in the predefined initial structure. Thus, the response of the compliant mechanism is predicated solely on buckling or bending behavior initiated in the VeroWhite branches, which is clearly in service of the force inversion objective. Second, despite noticeable differences in the topology between the selected inverter and those found in the literature [2, 85, 98, 12], the underlying geometry associated with the load path remains the same. As an example, Figure 2.12 compares the fabricated tensile inverter design generated by the L-System genetic optimization framework to a tensile inverter created using a SIMP-based method [104]. While the topologies of these mechanisms are obviously different,



(a) One of many tensile inverter designs generated using the L-System genetic optimization framework.

(b) Single tensile inverter generated using traditional subtractive topology optimization (SIMP) [104].

Figure 2.12: Comparison of geometry and topology of L-System-generated and traditional topology optimization-generated tensile inverters. While the overall topologies of the mechanisms differ, the underlying geometries (denoted by the dashed red lines) associated with the load paths are equivalent.

both share a similar core geometry (denoted by a dashed red line). As an input displacement is applied the two stiff members rotate, pushing the third stiff member to right and inducing the desired deformation at the output. Thus, despite having a far different methodology, the L-System framework is able to arrive at the same geometrical solution as traditional topology optimization methods, but without the large number of design variables and local stress concentrations associated with the SIMP topology optimization method. Furthermore, rather than simply one design, the framework provides a wide range of diverse preliminary options from throughout the Pareto frontier from which a designer may draw inspiration.

2.2.3 Compliant Mechanism: Kinematic Rectifier

The final design problem considers another compliant mechanism called a *kinematic rectifier* (or mechanical motion rectifier [99]); an input displacement drives an output displacement such that the direction of output displacement is independent of the direction of the input displacement. The initial structure, domain boundary, branch root location, and boundary conditions associated with this problem (Figure 2.13) are the same as the those of the tensile inverter mechanism (cf., Section 2.2.2), with the exception of the output point. To ensure physically meaningful designs capable of producing motion under load, a spring with a constant $k = 1$ N/mm is modeled as attached

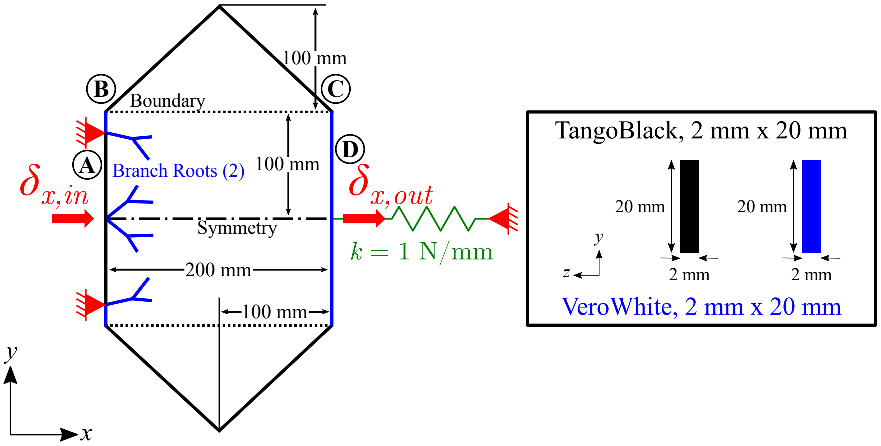


Figure 2.13: Initial structure, domain boundary, branch root location, and boundary conditions associated with the kinematic rectifier design study.

to the output point. This spring constant is based on an *a priori* knowledge of the expected output force and desired output displacement of the mechanism as inspired by the results of Section 2.2.2.

As before, two independent structural analyses are performed. In the first analysis a displacement is applied to the input node ($\delta_{x,in} = 5\,\text{mm}$) while the resulting output displacement $\delta_{x,out}$ is measured. In the second analysis an input displacement of the same magnitude is applied in the opposite direction ($\delta_{x,in} = -5\,\text{mm}$), and the resulting output displacement is again measured. Both steps account for potential geometric nonlinear behavior. Given the description of loading conditions associated with Figure 2.13 and the goal of maximizing $\delta_{x,out}$ to the left (i.e., negative) regardless of the input, the displacement inversion ratios are defined as

$$r_{push} = \frac{-\delta_{x,out}}{\delta_{x,in}} \quad \text{and} \quad r_{pull} = \frac{\delta_{x,out}}{\delta_{x,in}}, \tag{2.4}$$

with the objective being to minimize both quantities. The full design optimization problem is summarized in Table 2.4, where all other design variables, alphabet definitions, and genetic optimization parameters are the same as those outlined in Table 2.3. Once again, the FEA model consists of elements with an average length of approximately 30 mm.

The results of the multiobjective design optimization problem for the kinematic rectifier are

Table 2.4: Specification of the multiobjective topology optimization problem for kinematic rectifier mechanism.

Design Problem Statement	
Minimize:	r_{push}, r_{pull}
by varying:	2 axiom characters, 4 rule assignments (12 genes each), segment length parameter x, segment turn angle θ
subject to:	$\sigma_{Mises}^{max} \leq 33.8\,MPa$
Alphabet Definitions (Section & Length)	
`A,B`	VeroWhite, 2 mm×10 mm, Fwd x
`C,D`	TangoBlack, 2 mm×10 mm, Fwd $x/2$
NSGA-II Parameters [68]	
60 members for 400 generations, P_{cross} = 0.9, η_{cross} = 20, P_{mut} = 1/52, η_{mut} = 20	

shown in Figure 2.14. Unlike the previous two design problems, the Pareto frontier for the kinematic rectifier contains points that have negative objective values. To clarify, designs found in quadrant I are true kinematic rectifiers and perform as desired (invert motion when "pushed", transmit motion when "pulled"). Designs found in quadrant II are kinematic transmitters (the displacement of the output point is always in the same direction as the input point). Designs found in quadrant IV are kinematic inverters (the direction of the output displacement is always the opposite of the direction of input displacement as defined in Section 2.2.2).

Although a similar mechanism has been proposed by Li et al., where oscillatory vibration experienced by a shock absorber is converted into unidirectional rotation for the purposes of energy harvesting [99], there is no ideal solution with which the Pareto frontier can be compared. To choose a single-point design for further analysis, we consider a combination of designs closest to the utopia point ($r_{push} = 2.22$, $r_{pull} = 1.94$) and mechanism work efficiency. The work efficiency of the mechanism quantifies the fraction of external work applied to the input node that is

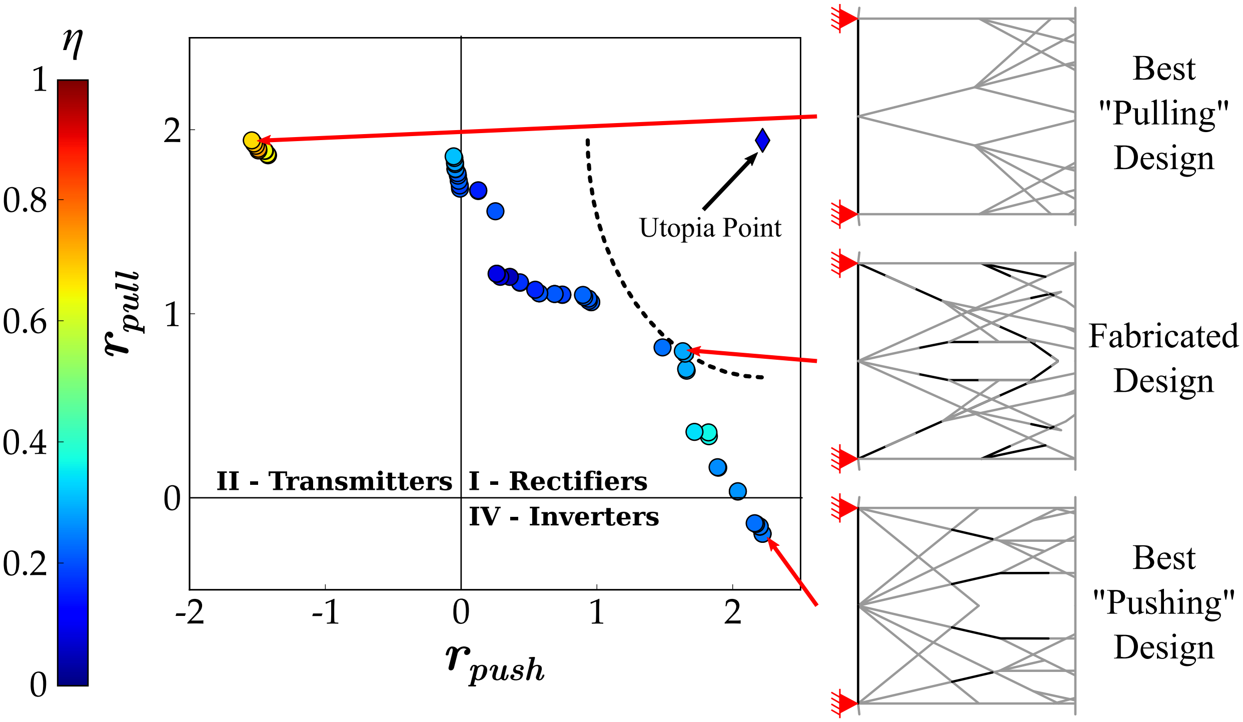


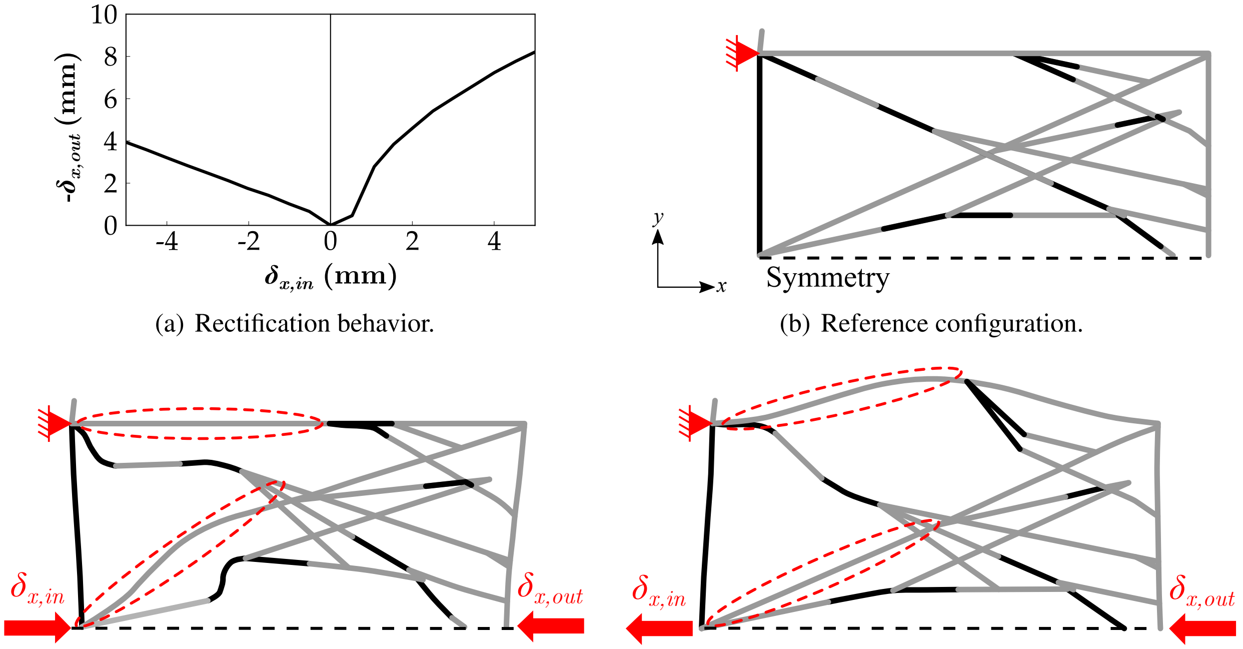
Figure 2.14: Pareto frontier of the kinematic rectifier design problem after 400 generations with associated mechanism efficiencies, along with the utopia point associated with the frontier. Dashed line shows points of constant distance from the utopia point Designs are shown without initial (null) structure.

subsequently stored in the output spring and is defined as

$$\eta = \frac{W_{spring}}{W_{input}} = \frac{\frac{1}{2}k\delta_{x,out}^2}{\int_0^{\delta_{x,in}} f_{x,in}\mathrm{d}x}, \tag{2.5}$$

where $f_{x,in}$ is the reaction force at the input point. The work efficiency of each design is denoted by its color in Figure 2.14. With the exception of a small subset of designs that function as simple kinematic transmitters (e.g., sets of axial rods), the work efficiencies of the Pareto frontier are less than $\eta = 0.4$ due to the storage of substantial strain energy within the structures, especially during buckling of segments. Since the designs that function as kinematic transmitters feature little/no buckling, their efficiencies approach $\eta_{max} = 0.744$. The design closest to the utopia point ($r_{push} = 1.63$, $r_{pull} = 0.80$) also has a higher efficiency ($\eta = 0.29$) than surrounding designs and is therefore selected as the fabricated design.

The undeformed and deformed configurations of the fabricated kinematic rectifier are shown in Figure 2.15. Note that the response of the mechanism to a given input is governed primarily by two individual branches; this design truly represents a multifunctional structure, as the functions of these two branches change depending on the direction of input. When undergoing a “pushing” input, the upper segment remains stiff in tension and causes the lower segment to buckle in compression. In this case the branch buckles upward, generating the desired displacement at the output point. When undergoing a “pulling” input, the lower branch remains rigid in tension, forcing the upper branch to buckle in compression. As before, the branch buckles upward and allows the output point to simply be pulled to the left.



(a) Rectification behavior.

(b) Reference configuration.

(c) Configuration during “pushing”.

(d) Configuration during “pulling”.

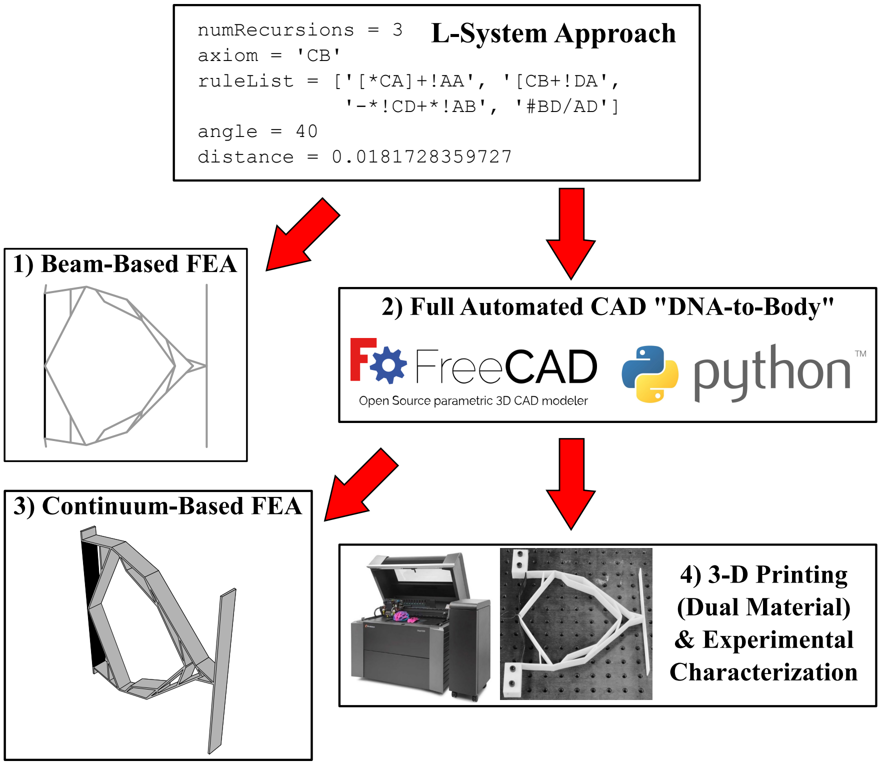
Figure 2.15: a) Illustration of mechanical motion rectification of the fabricated rectifier, and comparison of fabricated mechanism b) undeformed and c-d) deformed configurations (deformations are unscaled). Notice that the functionalities of two specific (circled) branches govern the rectification response of the mechanism.

2.3 Validated Analysis of Designs

As previously discussed, the L-System genetic optimization framework assesses the mechanical response of line segment-based topologies using computationally efficient beam element formulations. While effective for evaluating the responses of a large number of designs in a relatively short amount of time, these models neglect a number of critical aspects associated with a more rigorous modeling approach. Overlap of material at segment branches/intersections, precise calculation of inter-segment connections (and especially moment arms), and consideration of internal (self) contact are all neglected. Additionally, in the case of linear beam analysis (cf., Section 2.2.1), nonlinear responses such as buckling are also neglected. Thus, while the optimization framework has shown to produce non-intuitive designs capable of satisfying multiple objectives, the physical accuracy of the underlying predictions on which these designs are based should be suspect.

To truly demonstrate the feasibility of the proposed approach as a means of topology optimization, results from multiple structural models should be compared to both high-fidelity FEA models considering more realistic continuum responses, self-contact, buckling, and other mechanical aspects as well as experimentally characterized prototypes for the purpose of validation. While the L-System framework generates a 2-D topology consisting of a series of lines connected at vertices, an approach for generating 3-D volumetric structures for high fidelity analysis is illustrated in Figure 2.16. This method is enabled in large part by an open-source parametric modeling tool known as FreeCAD [105], which is based on the Python programming language, allowing for easy coupling with the developed L-System genetic optimization framework and the automated generation of 3-D volumetric models corresponding to the fabricated designs discussed in Section 2.2.

Validation of both frame- and continuum-based mechanical analyses must consider the characterization and quantified response of representative physical models. To that end, the method illustrated in Figure 2.16 allows for the exporting of 3-D solid models for both high-fidelity continuum-based FEA which considers more realistic continuum responses and additive manufacturing purposes. As mentioned in Section 2.2, this approach is based on structures fabricated on a Stratasys Objet 500 3-D printer and composed of TangoBlack and VeroWhite material.



```
numRecursions = 3
axiom = 'CB'
ruleList = ['[*CA]+!AA', '[CB+!DA',
            '-*!CD+*!AB', '#BD/AD']
angle = 40
distance = 0.0181728359727
```

L-System Approach
1) Beam-Based FEA
2) Full Automated CAD "DNA-to-Body"
FreeCAD
Open Source parametric 3D CAD modeler
python
3) Continuum-Based FEA
4) 3-D Printing (Dual Material) & Experimental Characterization

Figure 2.16: Graphical illustration of the multi-fidelity analysis, fabrication, and characterization approach. L-System formalized designs are directly analyzed using frame-based models, converted to 3-D volumetric bodies via scripted CAD [105], analyzed using continuum-based FEA, fabricated by 3-D printing, and experimentally characterized.

2.3.1 Continuum-Based Finite Element Analysis

As before, FEA models are analyzed using the Abaqus finite element suite, but here consider 3-D second-order continuum elements with reduced integration (Abaqus type C3D20R) rather than reduced-order beam elements. The results of the continuum-based analysis associated with each design are compared to frame-based modeling results, and conclusions are drawn about the performance of the frame-based modeling utilized by the optimization framework.

2.3.1.1 Cantilevered Frame

The 3-D volumetric model of the selected frame design detailed in Section 2.2.1 is shown in Figure 2.17. A fully-fixed boundary condition (i.e., $u_x = u_y = u_z = 0$) is applied to each

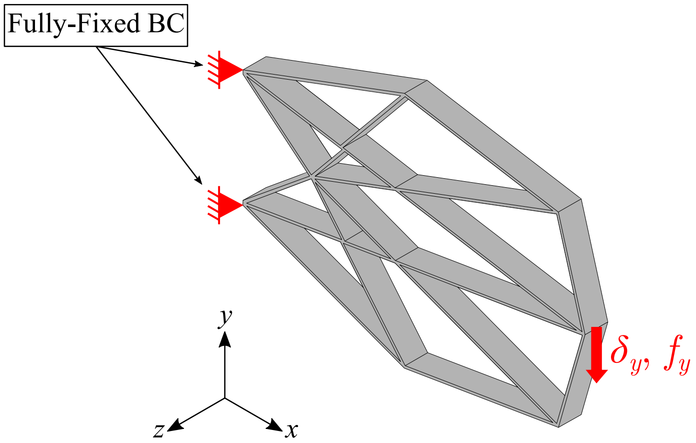


Figure 2.17: 3-D FEA model and associated boundary conditions for the continuum-based characterization of the fabricated frame design.

branch root, and a concentrated downward force of $f_y = 25\,\mathrm{N}$ is placed on the tip of the frame. Unlike the optimization analysis previously discussed, this continuum-based analysis considers nonlinear geometric effects, allowing for accurate results by accounting for buckling and other large deformations. The frame structure is made up of 8,964 elements as determined by a mesh density study detailed in Appendix C.

Figure 2.18 shows the predicted tip displacement of the frame as a function of applied load for both frame-based and continuum-based FEA. At lower applied loads both analyses predict approximately the same structural stiffness, though in the detailed analysis higher applied loads elicit a decreased stiffness in the structure. This can be attributed to noticeable buckling in the part during loading, which is not captured in the linear frame-based approach and indicates a potential problem with the frame-based optimization approach as implemented.

2.3.1.2 Compliant Mechanism: Tensile Inverter

Figure 2.19a depicts the 3-D volumetric model of the dual-material tensile inverter design considered in Section 2.2.2. Like its 2-D counterpart, only half of the inverter is modeled, with symmetry constraints applied to the negative y-surfaces of the part. Fully-fixed boundary conditions are applied to the left side to prevent rigid body translation and rotation. The free displace-

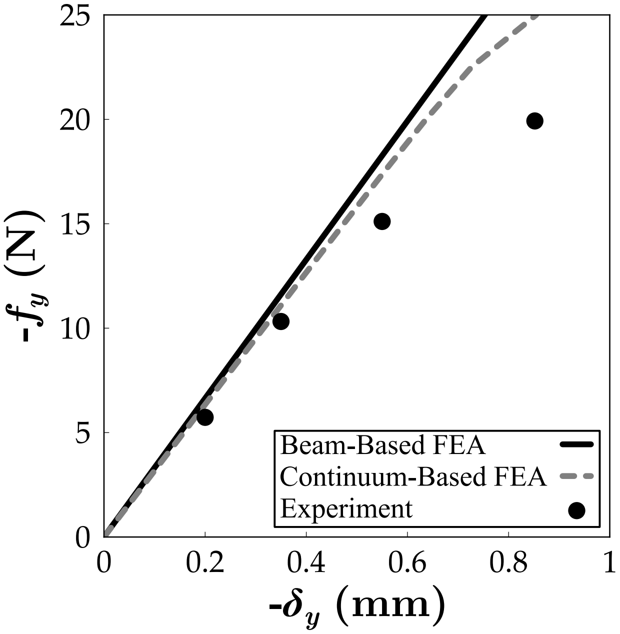
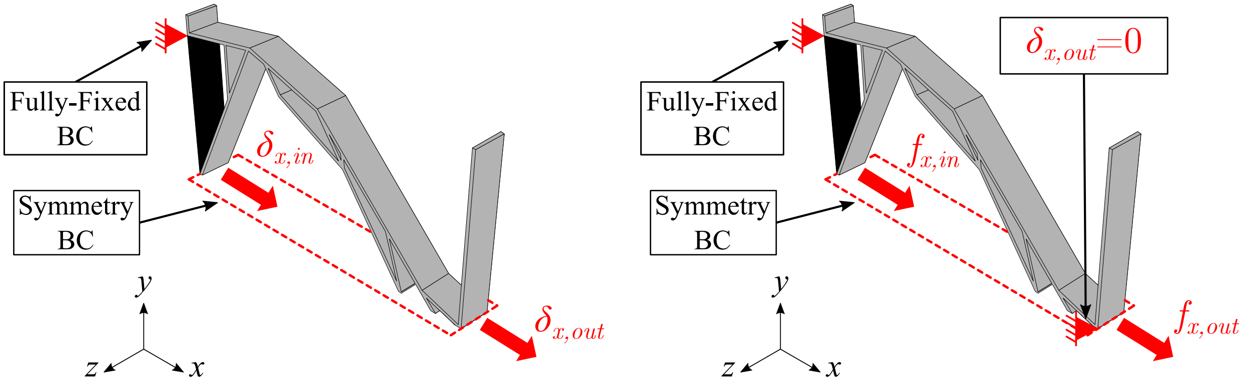


Figure 2.18: Comparison of numerical simulations and experimental results for the behavior of the frame. Note that the experiment shows good agreement with the simulations during linear response, but exhibits more buckling behavior than was predicted for larger applied forces.

ment inversion test and blocked force test are identical to those described in Section 2.2.2. The continuum-based analysis also allows for nonlinear geometric effects and employs 13,062 elements (mesh density study found in Appendix C).

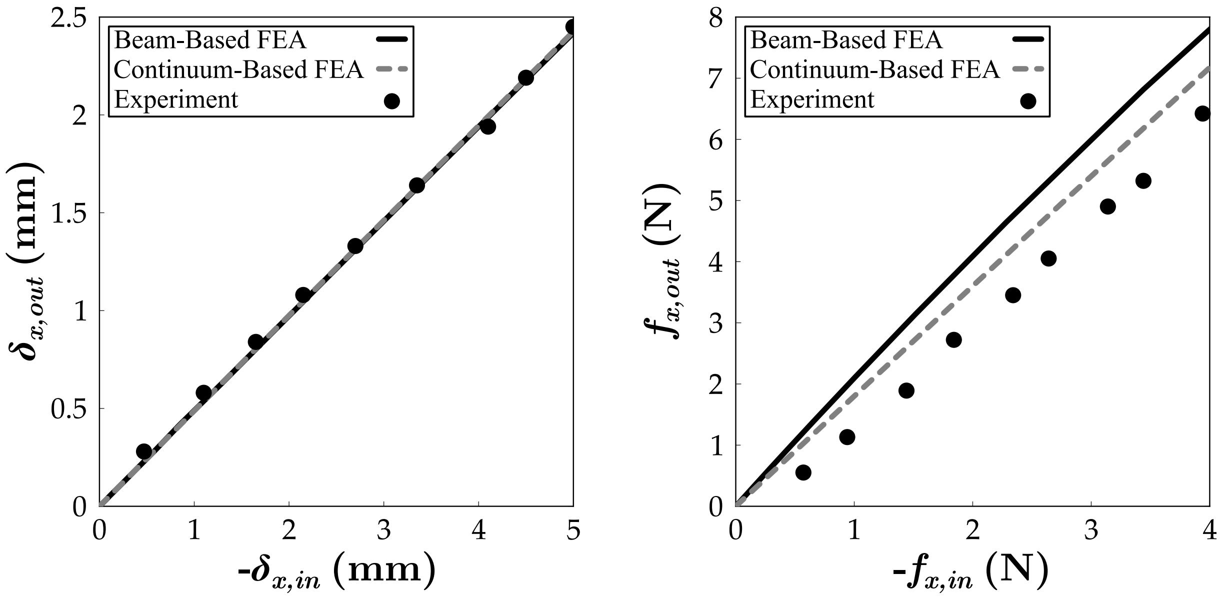
A comparison of the inversion behavior for the frame-based and continuum-based analyses is shown in Figure 2.20. It is interesting to note that the predicted displacement inversion behaviors in both simulations is essentially identical (Figure 2.20a). However, Figure 2.20b shows that the predicted force inversion behavior in the continuum-based analysis is slightly lower than that of the frame-based analysis. These results are consistent with the differences in inversion ratios observed by Hartl et al. [85] and are attributed to the differences in element formulations as well as the mesh densities associated with each model.



(a) Boundary conditions associated with displacement inversion testing.

(b) Boundary conditions associated with force inversion testing.

Figure 2.19: 3-D FEA model and associated boundary conditions for the continuum-based characterization of the selected tensile inverter design.



(a) Displacement inversion behavior.

(b) Force inversion behavior.

Figure 2.20: Comparison of numerical simulations and experimental results for the behavior of the tensile inverter. For both test cases the experimental data shows good agreement with computational predictions.

2.3.1.3 Compliant Mechanism: Kinematic Rectifier

The 3-D volumetric model of the fabricated dual-material rectifier design (cf., Section 2.2.3) is illustrated in Figure 2.21. As in the previous section, only half of the component is considered, with symmetry boundary conditions placed on the negative y-surfaces of the rectifier. Both test cases for the continuum-based model are identical to those discussed in Section 2.2.3 and illustrated in Figure 2.21. The continuum-based analysis considers both nonlinear geometric effects and contact between segments, and the model is constructed using 27,230 elements (cf., Appendix C).

The predicted response of the rectifier for both the optimization and detailed analyses are shown in Figure 2.22. Notice that for the "pushing" displacement test the frame-based and continuum-based models predict a nearly identical mechanism response (Figure 2.22a). When considering a "pulling" input displacement (Figure 2.22b), the frame-based FEA predicts a slightly higher output displacement than that of the continuum-based FEA; this again can be attributed to differences in element formulations and mesh densities associated with the two models. Nevertheless, in both cases the computational analyses are in good agreement.

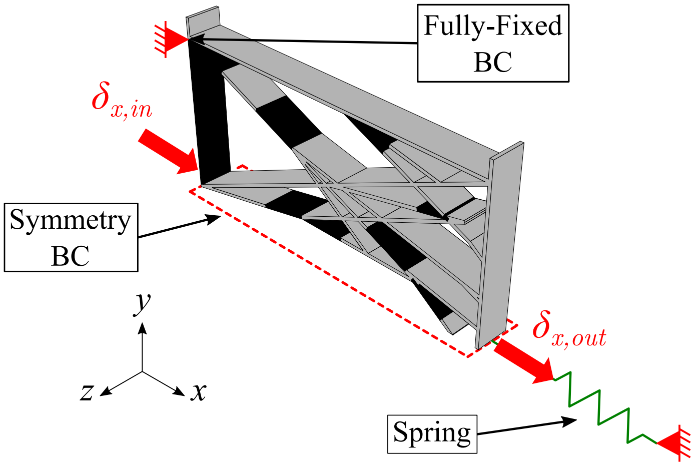
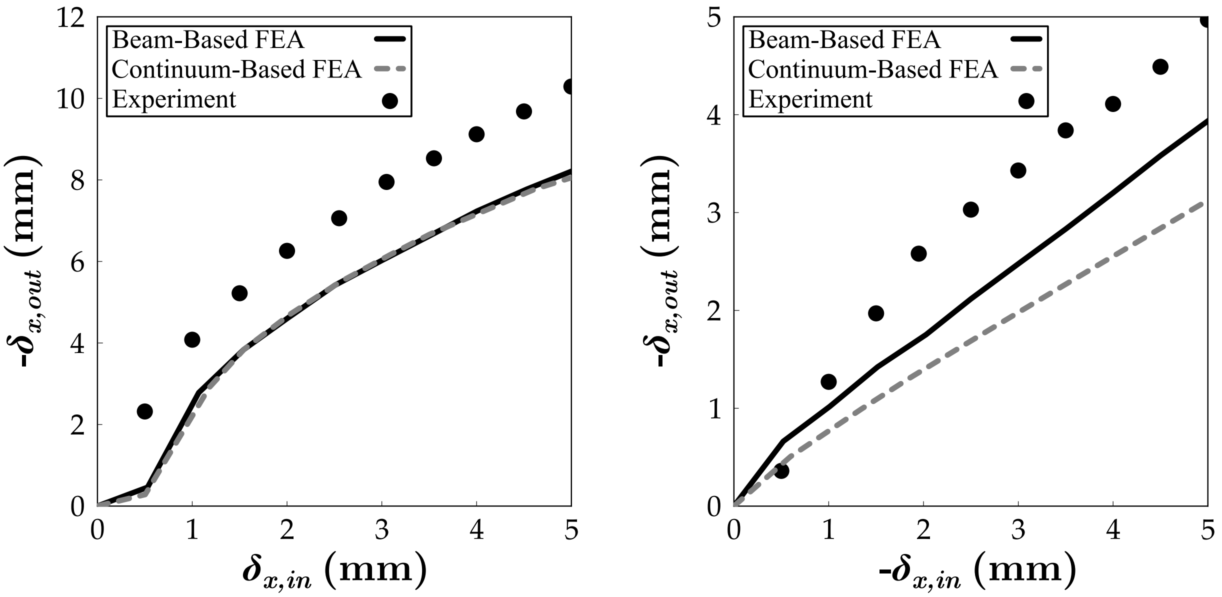


Figure 2.21: 3-D FEA model and associated boundary conditions for the continuum-based characterization of the fabricated kinematic rectifier design.



(a) "Pushing" displacement behavior (inversion).

(b) "Pulling" displacement behavior (transmission).

Figure 2.22: Comparison of experimental and computational results for the behavior of the kinematic rectifier. While both computational predictions are in good agreement, experimental results overpredict the output response of the mechanism due to repeated buckling of structural components that occurred during testing.

2.3.2 Experimental Characterization of Structures

Having compared frame- and continuum-based structural analyses of the three design problems introduced in Section 2.2, the experimental characterization of representative physical models is now considered. These models are manufactured using a Stratasys Objet 500 3-D printer, which cures deposited material using ultraviolet (UV) light and is capable of printing two materials simultaneously, thus precluding the need for additional bonding between material interfaces. The mechanical responses of these physical models are compared to the frame- and continuum-based analyses associated with each test case. To quantify the agreement between experimental and simulation results, the root-mean-square (RMS) error is calculated as

$$\epsilon = \sqrt{\frac{\sum_{i=1}^{n}(x_{i,exp} - x_{i,frame})^2}{n}}, \tag{2.6}$$

where $x_{i,exp}$ and $x_{i,frame}$ are the values of interest resulting from experimental characterization and frame-based FEA, respectively, and n is the number of experimental data points gathered. Using the numerical values calculated for each test case as shown in Table 2.5, conclusions are drawn about the ability of the optimization framework to develop realistic multifunctional mechanisms.

It should be noted that the error values provided in Table 2.5 assume that the 3-D printed materials used to manufacture each prototype have properties consistent with those provided in Table 2.1. However, the properties of these materials have shown to be sensitive to variations in the parameters of the printing process. For example, studies performed by Barclift and Williams demonstrated that changing printing parameters such as in-plane build orientation, out-of-plane build orientation, and part spacing resulted in measured VeroWhite elastic modulus values ranging from 1176 to 2495 MPa [106]. Clearly, a material property variability of this magnitude would have a noticeable effect on the predicted behavior of each structure during the optimization process. While outside the scope of this work, one could account for these variabilities by optimizing based on a range of potential structural responses rather than a single deterministic response. This process would be relatively straightforward for a single-material structural problem which assumes a linear structural response (i.e., the cantilevered frame problem in Section 2.2.1), but becomes more complex when considering multiple materials, each with their own inherent variable properties, and nonlinear structural responses.

Table 2.5: Overview of errors between experimental data and frame-based FEA predictions for each structural test case. All structures are 300 mm in length.

Structure	Test	RMS Error	% of Max Magnitude
Frame	Transverse Force Test	0.14 mm	15.9
Tensile Inverter	Displacement Inversion Test	0.03 mm	1.4
	Force Inversion Test	1.19 N	17.9
Kinematic Rectifier	“Pushing” Displacement Test	1.68 mm	16.3
	“Pulling” Displacement Test	0.77 mm	15.5

2.3.2.1 Cantilevered Frame

An illustration of the experimental setup for frame characterization is shown in Figure 2.23. One end of a high stiffness string was attached to the tip of the frame, while the other was attached to a Shimpo FG-7000 force gauge with a resolution of 0.01 N. This force gauge was abutted to a micrometer, which, when adjusted, allows for the smooth application of loading. To provide accurate displacement measurements, a Keyence IL-600 laser sensor with a resolution of 0.01 mm was placed next to the micrometer. A small piece of reflective tape was placed on the end of the frame to allow the laser sensor to measure displacement parallel to the direction of applied force. Given the overall length of the frame (300 mm) and the expected tip deflection during loading (≈ 1 mm), this method accounted for the translational motion of the reflective tape while neglecting its rotation.

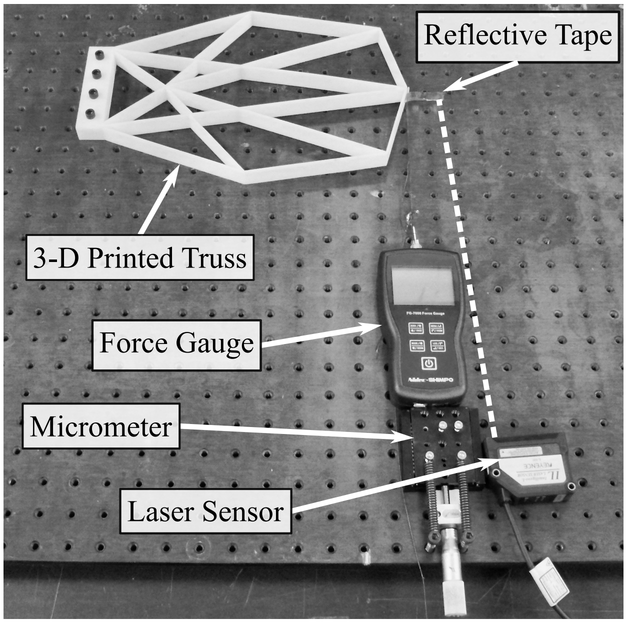


Figure 2.23: Experimental setup for testing of the 3-D printed frame.

Figure 2.18 shows a comparison between the numerical predictions presented in Section 2.3.1.1

and experimental results. Where good agreement is observed during linear response, noticeable buckling behavior is exhibited by the 3-D printed frame at applied loads above 10 N. Using Equation 2.6, the total RMS error for the frame is calculated to be $\epsilon_{frame} = 0.14\,\text{mm}$. This is approximately 15.9% of the maximum displacement observed during the experiment. Thus, while both simulated results for the frame are unable to accurately capture the experimentally observed buckling behavior, they are still in relatively good agreement with the response of the physical model.

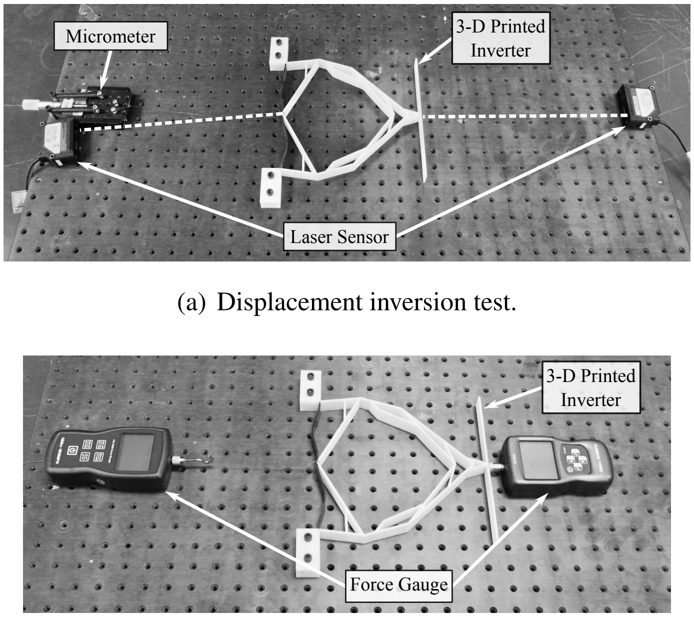
2.3.2.2 Compliant Mechanism: Tensile Inverter

Illustrations of the experimental setups for the 3-D printed tensile inverter are shown in Figure 2.24. To test the displacement inversion behavior (Figure 2.24a), a high stiffness string connected a micrometer to the inverter. One Keyence laser sensor was aimed at the input to the inverter, while a second laser sensor was aimed at its output. To test the force inversion behavior (Figure 2.24b), the micrometer was replaced by a Shimpo force gauge, while a second force gauge constrained the displacement at the output.

Comparisons of experimental results to the computational predictions discussed in Section 2.3.1.2 are shown in Figure 2.20. Note that the experimental data associated with displacement inversion is nearly identical to both computational simulations, with an RMS error of $\epsilon_\delta = 0.03\,\text{mm}$ (1.4% of the maximum output displacement). Experimental testing results in a slightly higher average force inversion ratio ($r_f \approx 1.38$) than is predicted in the frame-based simulation ($r_f = 1.29$), corresponding to an RMS error of $\epsilon_f = 1.19\,\text{N}$ (17.9% of the maximum blocking force). Therefore, once again computational predictions are able to capture the general response of the physical model.

2.3.2.3 Compliant Mechanism: Kinematic Rectifier

Figure 2.25 depicts the experimental setups for the 3-D printed kinematic rectifier. Due to manufacturing complications, the mechanism was printed with a thickness of 20 mm (compared to the 10 mm considered in the computational analyses). To test the "pushing" (inversion) response



(a) Displacement inversion test.

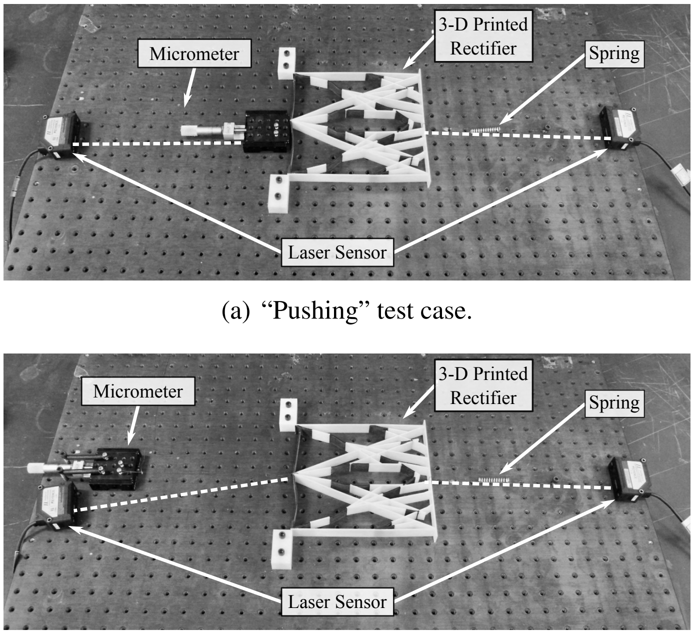
(b) Force inversion test.

Figure 2.24: Illustrations of experimental setups for the 3-D printed tensile inverter.

of the rectifier (Figure 2.25a), a micrometer was abutted to the input of the mechanism. As the micrometer was adjusted, it imparted an input displacement on the rectifier, which was measured using a laser sensor aimed directly at the back of the of the micrometer; a second laser sensor tracked the displacement at the output. To test the "pulling" response of the rectifier (Figure 2.25b), a high stiffness string connected the micrometer to the rectifier. One laser sensor was aimed at the input of the rectifier, while another was aimed at the output. In both test cases, a spring with an experimentally characterized stiffness of 3.9 N/mm[7] was connected to the rectifier.

Figure 2.26 shows a comparison of the distributed deformation at an input displacement $\delta_{x,in} = 5\,\text{mm}$ in both the continuum-based FEA model and physical prototype. Notice that in both models the segment buckling behavior and overall deformation of the structure are the same, further

[7] Due to the assumed symmetry condition in both computational models, the spring used to experimentally characterize the full mechanism should have double the spring stiffness of that used in the analysis (i.e., 2 N/mm). Due to manufacturing complications, the physical prototype was printed with an out of plane thickness double that of the computational models. To provide a direct comparison between computational predictions and experimental characterization regarding motion, the experimental spring constant was again doubled to ≈4 N/mm.



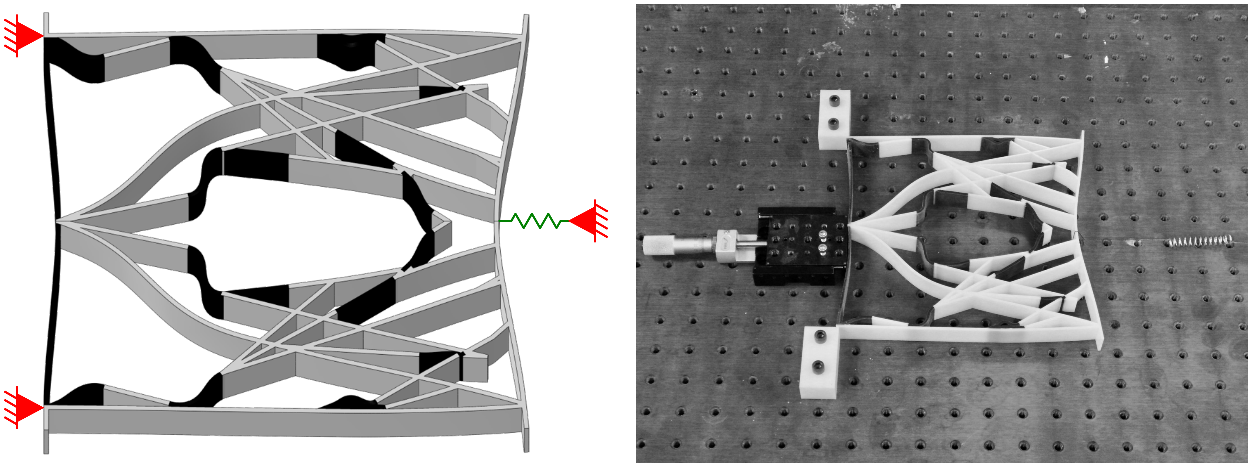
(a) “Pushing” test case.

(b) “Pulling” test case.

Figure 2.25: Experimental setups for the 3-D printed kinematic rectifier.

validating the ability of the optimization framework to find realistic solutions to multifunctional structural problems. A comparison between experimental measurements and computational predictions for both rectifier test cases are shown in Figure 2.22.

In both test cases higher magnitudes of output displacement are seen during the experiment than in simulated results; this is believed to be caused by the permanent warping of several structural components that evolved over loading cycles due to large deformations; such permanent deformation was not considered in the FEA models. The RMS error for the “pushing” test case (Figure 2.22a) is calculated as $\epsilon_{rect,push} = 1.68\,\text{mm}$ (16.3% of the maximum output displacement); however, the RMS error for the “pulling” test case (Figure 2.22b) is calculated as $\epsilon_{rect,pull} = 0.77\,\text{mm}$ (15.5% of the maximum output displacement), which is expected given that this test case represents a simpler response overall (simple axial tensile load transfer between input and output). On the whole, despite a slight overprediction, experimental results once again agree well with predictions from both computational models.



(a) continuum-based FEA model.

(b) 3-D printed model.

Figure 2.26: Comparison of deformed rectifier structure corresponding to continuum-based FEA and physical models during “pushing” (inversion) for an input displacement of $\delta_{in} = 5\,\text{mm}$. The “pulling” deformation is much simpler in nature.

3. TWO-DIMENSIONAL L-SYSTEM TOPOLOGY OPTIMIZATION USING GRAPH-BASED INTERPRETATION*

Although the standard L-System with geometric interpretations discussed in Chapter 2 make it possible to generate a variety of interesting structures, from abstract fractal patterns to plant-like branching structures, their modeling power is actually quite limited. One problem can be traced to the definition of all structure as a function of some distance parameter x (cf., Section 2.2.1.1), meaning that there are only a finite number of possible lengths a segment in the structure can have, This same issue is also found in orientation-changes, as each angle change is simply a function of some angle parameter θ. Thus, while the L-System formulation presented in Chapter 2 is able to create non-intuitive self-similar structures that can satisfy a given multiobjective design problem, the constraints placed on distance and angle definitions limit the potential design space and can lead to L-System-generated topologies that perform well below known optimal designs.

Furthermore, the turtle graphics algorithm was actually first developed to introduce computer programming to those unfamiliar with the subject [75] and is not particularly well-suited to structural topology development. The creation of meaningful structures requires the existence of a load path between points on which displacement and/or force boundary conditions are applied. As discussed in Section 2.1.2, the algorithm constructs straight line segments in between two points in 2-D space based on interpreted L-System instructions. Therefore, generating viable structural topologies with the turtle graphics algorithm generally relies on line segments emanating from applied boundary conditions to intersect; while certainly not impossible, any intersections that do occur are not explicitly defined by the algorithm and occur by happenstance. Additionally, when using the turtle graphics algorithm lines with different material assignments are allowed to be created on top of one another; in this case, the last material assigned to that area is what is represented in the final topology. Thus material reassignment and the formation of load paths, which form

*Portions reprinted with permission from “L-System-Generated Mechanism Topology Optimization Using Graph-Based Interpretation” by Bielefeldt, B. R., Akleman, E., Reich, G. W., Beran, P. S., and Hartl, D. J., 2019. *Journal of Mechanisms & Robotics*, 11(2), 1-10, Copyright 2019 by ASME.

the very essence of topological description, occur coincidentally when using the turtle graphics algorithm.

This chapter describes the development of a graph-based L-System interpreter known as Spatial Interpretation for the Development of Reconfigurable Structures (SPIDRS), which is written such that its interpretation of parameterized L-System commands leads to topological changes that are deliberate and natural, and is organized as follows: Section 3.1 provides an introduction to several important principles of graph theory; Section 3.2 discussed the graph-based interpretation algorithm, including the parameterized L-System formulation it interprets, the data structure that forms the foundation for the algorithm, and the graphical operations that initiate topological modifications to the graph; Section 3.3 then describes the optimization framework and how graphical information produced by SPIDRS is converted into a structural topology; Section 3.4 demonstrates the proposed graph-based interpreter using two benchmark problems, where Pareto optimal designs are compared to both known optimal/ideal solutions and those generated using the geometry-based interpretation method proposed in Chapter 2 where possible; Section 3.5 explores several extensions enabled by the SPIDRS algorithm, such as the optimization of nodal locations on the graph and the ability to consider large, solid structural domains similar to density-based topology optimization methods; and finally, having considered the results of the previous section, Section 3.6 proposes modifications to the SPIDRS algorithm and topology optimization framework and demonstrates the improved performance of the proposed method using several benchmark problems.

3.1 Graph Theory: Preliminary Definitions

In mathematics, *graph theory* is the study of graphs, or structures used to model pairwise relations between objects. In general, a graph G is defined as an ordered triple $(N(G), E(G), \psi_G)$ consisting of a nonempty set $N(G)$ of *nodes*, a set $E(G)$ of *edges* that is disjoint from $N(G)$, and an *incidence function* ψ_G that associates each edge of G with an unordered pair of (not necessarily distinct) nodes of G. If e is an edge and u and v are nodes such that $\psi_G(e) = uv$, then e is said to join u and v and these nodes are referred to as the ends of e [107, 108, 109]. Consider, for example, the graph illustrated in Figure 3.1a which can be written as

$$G = (N(G), E(G), \psi_G)$$

where

$$N(G) = \{n_1, n_2, n_3, n_4, n_5\},$$
$$E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\},$$

and where ψ_G is defined by

$$\psi_G(e_1) = n_1n_2, \quad \psi_G(e_2) = n_2n_3, \quad \psi_G(e_3) = n_3n_3, \quad \psi_G(e_4) = n_3n_4,$$
$$\psi_G(e_5) = n_2n_4, \quad \psi_G(e_6) = n_4n_5, \quad \psi_G(e_7) = n_2n_5, \quad \psi_G(e_8) = n_2n_5.$$

In a similar vein, the graph H depicted in Figure 3.1b can be written as

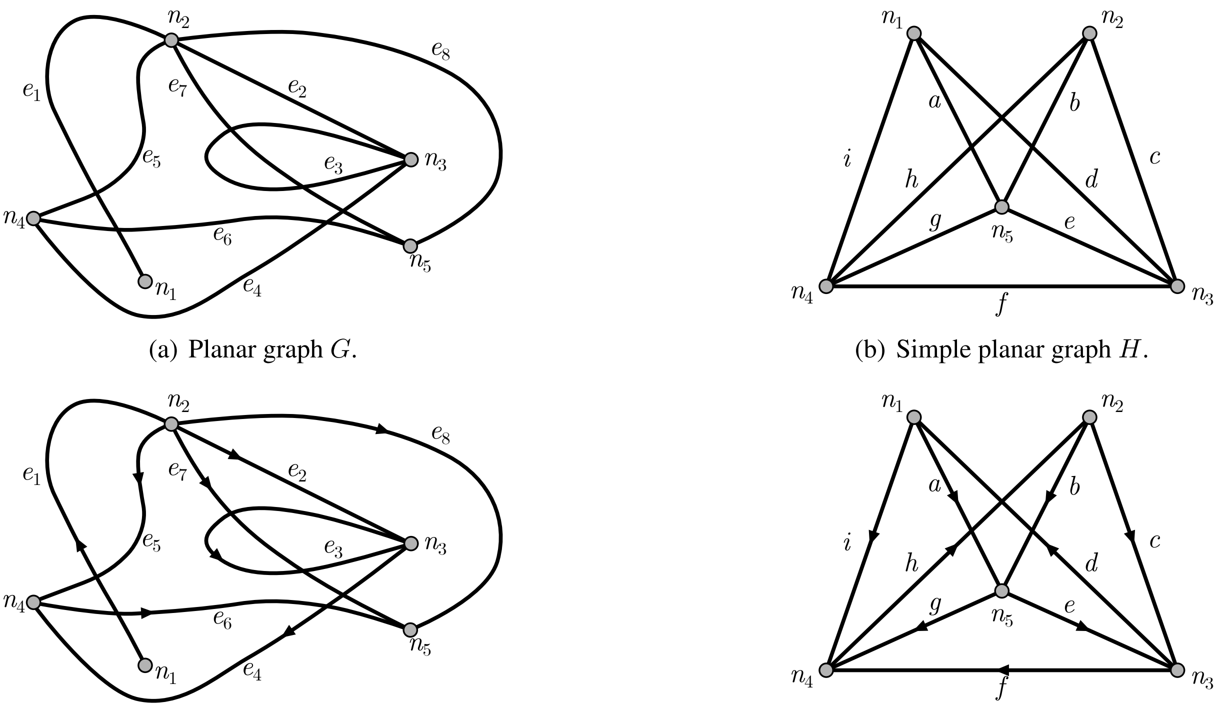
$$H = (N(H), E(H), \psi_H)$$

where

$$N(H) = \{n_1, n_2, n_3, n_4, n_5\},$$
$$E(H) = \{a, b, c, d, e, f, g, h, i\},$$

and where ψ_H is defined by

$$\psi_H(a) = n_1n_5, \quad \psi_H(b) = n_2n_5, \quad \psi_H(c) = n_2n_3, \quad \psi_H(d) = n_1n_3, \quad \psi_H(e) = n_3n_5$$
$$\psi_H(f) = n_3n_4, \quad \psi_H(g) = n_4n_5, \quad \psi_H(h) = n_2n_4, \quad \psi_H(i) = n_1n_4.$$



(a) Planar graph G.

(b) Simple planar graph H.

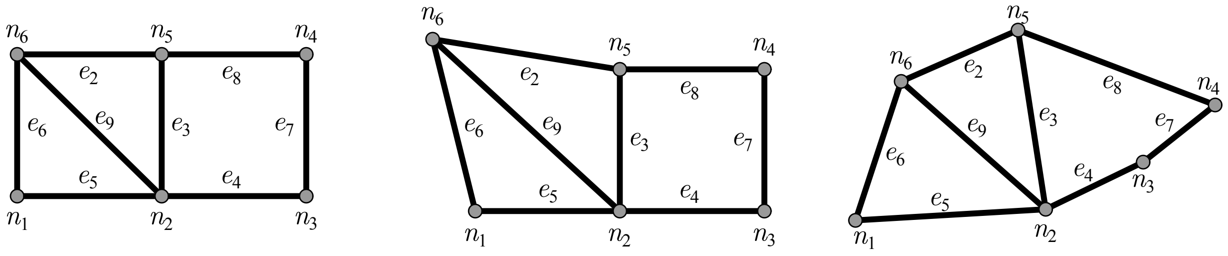
(c) Directed planar graph G'. Note that in this case $\psi_G = \psi_{G'}$.

(d) Directed simple planar graph H'. Note that in this case $\psi_H \neq \psi_{H'}$.

Figure 3.1: Diagrams of example graphs (adapted from [108]).

Graphs are so named because they can be represented graphically, allowing one to understand their properties. In Figures 3.1a-b, each node is represented by a point and each edge by a line connecting the points which represent their ends as defined by ψ_G or ψ_H, respectively. It should be noted that there is no unique way of drawing a graph, as the relative positions of points representing nodes and lines representing edges have no spatial significance. For example, the diagrams shown in Figure 3.2 are all valid representations of graph I, as each represents the same sets of nodes and edges and the same incidence function. This concept is notable when considering the effect of the geometry of the design domain on the performance of potential structural configurations and will be revisited in Section 3.5.1.

One can further characterize a graph based on its properties [108]. Graphs that have a diagram whose edges intersect only at their end nodes are referred to as *planar graphs*, as such graphs can be represented in the plane in a simple manner. Though not immediately clear, both of the graphs



(a)

(b)

(c)

Figure 3.2: Three diagrams that all present the graphical information of graph I.

in Figures 3.1a-b are planar, as one can use the aforementioned property of graphs to redraw their diagrams such that their edges only intersect one another at end nodes. A graph is *simple* if it has no loops (i.e., an edge that joins a node to itself) and no two edges join the same pair of nodes. Graph G in Figure 3.1a is not simple, as edge e_3 forms a loop and edges e_7 and e_8 both join nodes n_2 and n_5. Graph H in Figure 3.1b, however, is a simple graph. Finally, a *directed graph* is a graph in which each edge has an assigned orientation. Directed graphs (herein referred to as G' and similar) are defined identically to general graphs with the exception of the incidence function, which no longer associates each edge of G' with an unordered pair. Instead, given $\psi_{G'}(e) = uv$, u is said to be the *tail* of e and v is said to be the *head* of e. Figure 3.1c depicts a directed graph G' using G (cf., Figure 3.1a) as its *underlying graph*. Note that in this case ψ_G correctly labels the tails and heads of each edge in $E(G')$ such that $\psi_G = \psi_{G'}$. A directed graph H' based on underlying graph H (cf., Figure 3.1b) is shown in Figure 3.1d. Here, $\psi_H \neq \psi_{H'}$ because the edges in the set $E(H')$ have tails and heads that are not captured within ψ_H, specifically $\psi_H(e)$, $\psi_H(g)$, and $\psi_H(h)$. Based on the notion that edges have an orientation, one can define the concept of a *directed walk* in a directed graph G', which is a finite non-null sequence $W_{G'} = (n_0, e_1, n_1, \ldots, e_k, n_k)$ whose terms are alternatively nodes and edges such that, for $i = 1, 2, \ldots, k$, the edge e_i has head n_i and tail n_{i-1}. Note that this sequence is not necessarily unique. This sequence can be further simplified to include only the sequence of nodes along the walk. For example, graphs G' and H' have directed walks of $W_{G'} = (n_1, n_2, n_3, n_4, n_5)$ and $W_{H'} = (n_1, n_5, n_3, n_1, n_4, n_2, n_5, n_4)$, respectively.

3.2 Spatial Interpretation for the Development of Reconfigurable Structures: A Graph-Based Interpretation Algorithm

3.2.1 The Parameterized L-System Approach

To increase the modeling capability of L-Systems, Lindenmayer proposed the addition of numerical parameters to both variable and constant L-System characters to allow for varying distance and angle definitions [110]. To construct a parameterized L-System, the formulation discussed in Section 2.1.1 is modified to include a set of *formal parameters* $\mathbf{\Sigma}$ associated with certain characters of the alphabet that allow for greater flexibility when interpreting the L-System encoding. In the parameterized L-System formulation presented herein, the set $\boldsymbol{\alpha}$ is modified to include an additional variable character such that $\boldsymbol{\alpha} = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}\}$ (motivation to be later described), while the sets $\boldsymbol{\beta} = \{\mathtt{-}, \mathtt{+}\}$ and $\boldsymbol{\gamma} = \{\mathtt{[}, \mathtt{]}\}$ remain unchanged from Section 2.1.1. Characters in the sets $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined such that $\alpha_i = \alpha_i(\sigma_{\alpha_1}, \sigma_{\alpha_2})$ and $\beta_i = \beta_i(\sigma_{\beta_1})$, where σ_{α_1}, σ_{α_2}, and σ_{β_1} are real numbers associated with the set of formal parameters $\mathbf{\Sigma}$ and take on different meanings depending on the characters in $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ that they are associated with. As in Section 2.1.1, an axiom of two letter characters ($\boldsymbol{\omega}^0 = \omega_1\omega_2, \omega_i \in \boldsymbol{\alpha}$) is used to initiate the recursive creation of topological instructions.

Considering the set of production rules $\boldsymbol{P}$, this parametric L-System formulation considers five production rules, each consisting of 10 characters. These are written as[1]

$$\begin{aligned} P_i : \alpha_i(\sigma_{\alpha_1}, \sigma_{\alpha_2}) \rightarrow & \lambda_1^i \lambda_2^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_2 \lambda_3^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_3 \lambda_4^i(\sigma_{\beta_1})_4 \lambda_5^i \\ & \lambda_6^i \lambda_7^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_7 \lambda_8^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_8 \lambda_9^i(\sigma_{\beta_1})_9 \lambda_{10}^i, \end{aligned}$$

[1] The operator $(\sigma_{\alpha_1}, \sigma_{\alpha_2})_j$ here is meant to signify that the associated parameters belong to λ_j^i and are independent of any other parameters (i.e., $(\sigma_{\alpha_1}, \sigma_{\alpha_2})_2 \neq (\sigma_{\alpha_1}, \sigma_{\alpha_2})_3$, etc.).

$i = 1, \ldots, 5$ for all $\alpha_i \in \boldsymbol{\alpha}$, where $\lambda_1^i \lambda_2^i \ldots \lambda_{10}^i$ is a string of 10 characters such that $\lambda_j^i \in \Lambda_j$ and

$$\Lambda_1 = \{\, \texttt{[}\,,\_ \}\,,\;\; \Lambda_2 = \boldsymbol{\alpha},\;\; \Lambda_3 = \boldsymbol{\alpha},\;\; \Lambda_4 = \boldsymbol{\beta},$$
$$\Lambda_6 = \{\, \texttt{[}\,,\_ \}\,,\;\; \Lambda_7 = \boldsymbol{\alpha},\;\; \Lambda_8 = \boldsymbol{\alpha},\;\; \Lambda_9 = \boldsymbol{\beta}.$$

As before, assignments for λ_5^i and λ_{10}^i are made such that the "begin branch" indicator "[" in λ_1^i and λ_6^i, if assigned, is closed. Given the definition of each production rule and the need for each character in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to have two parameters and one parameter, respectively, each production rule requires 10 individual parameters; thus, only 18 independent variables are needed to define each production rule, for a total of 92 independent variables *which remain completely independent of structural complexity or configuration*. These variables are determined via the genetic optimization framework that will be discussed in Section 3.3.

The recursive generation of final strings using the parameterized L-System approach is illustrated in Figure 3.3. While not shown, every variable character in each production rule has two associated parameters. The numbers shown in each production rule represent a character from Λ_4 or Λ_6 as well as an associated parameter, and are used here to show that these characters/parameters are carried through each iteration.

3.2.2 Half-Edge Data Structure

Concepts from graph theory are implemented and manipulated within the SPIDRS algorithm using a *half-edge data structure* (also known as a *doubly-connected edge list*), which was first proposed by Muller and Preparata [111] and represents a directed simple planar graph G' in such a way that it is easy to traverse and manipulate [112]. The data structure contains a record of each node, edge, and face of the graph, which are designated as follows:

- Nodes are defined in manner consistent with Section 3.1.
- Edges are defined based on the idea that each edge can be represented using two half-edges that are oriented in opposite directions and assumed to belong to two distinct faces. Thus,

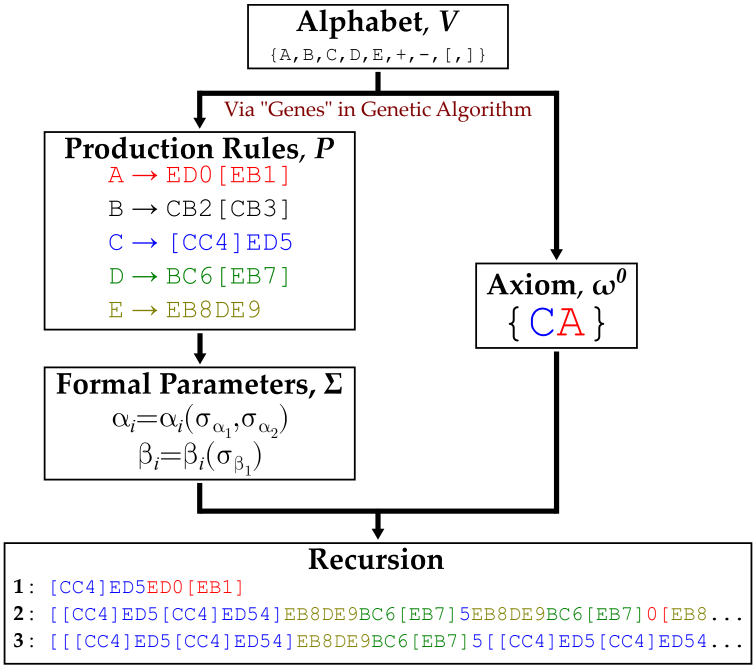


Figure 3.3: Example of recursively generated final strings using a parameterized L-System approach consisting of an axiom and production rules constructed with a specific alphabet. Note that A, B, C, D, and E represent variable characters. Numbers illustrate parameterized angle changes, and are typeset as simple integers here to show that these commands are carried through each iteration.

for an edge e that joins nodes u and v, there exists two half-edges e^+ and e^- such that $\psi_{G'}(e^+) = uv$ and $\psi_{G'}(e^-) = vu$. Herein, for simplicity half-edges will be denoted as e_{uv}, where u and v are the tail and head of the half-edge, respectively. The orientation of a given half-edge is assigned to be counterclockwise relative to the face it is contained in. Because the half-edge data structure assumes a planar graph, edges can only intersect one another at defined nodes.

- Faces are defined as directed walks $W_{G'} = (n_0, n_1, \ldots, n_{k-1}, n_k)$ (cf., Section 3.1) where $n_0 = n_k$. In other words, a face is an enclosed set of half-edges oriented such that one can start at a given node, traverse around a portion of the graph based on the orientation of adjacent half-edges, and return to that node. Herein, n_k will be omitted from the definition of a face and it will be assumed that once an observer reaches n_{k-1} they will return to the "beginning" of the face n_0.

In addition, the half-edge data structure can also contain a record of additional information assigned to specified nodes, edges, and faces (known as *attribute information*) [112].

To better introduce this concept, consider the directed simple planar graph I' shown in Figure 3.4. The sets of nodes and edges that make up I' is defined as

$$N(I') = \{1, 2, 3, 4, 5, 6\},$$
$$E(I') = \{e_{12}, e_{23}, e_{34}, e_{41}, e_{25}, e_{56}, e_{63}, e_{32}\}.$$

In this example, e_{23} and e_{32} are defined as *twin* half-edges because they make up a single edge between nodes 2 and 3. Note that in this example (as well as in Section 3.4), half-edges on the exterior of the graph are neglected (i.e., there is no half-edge e_{14} defined in I'); this will be revisited in Section 3.6.2. The set of faces that make up I' are intuitively written as

$$F(I') = \{f_1, f_2\},$$

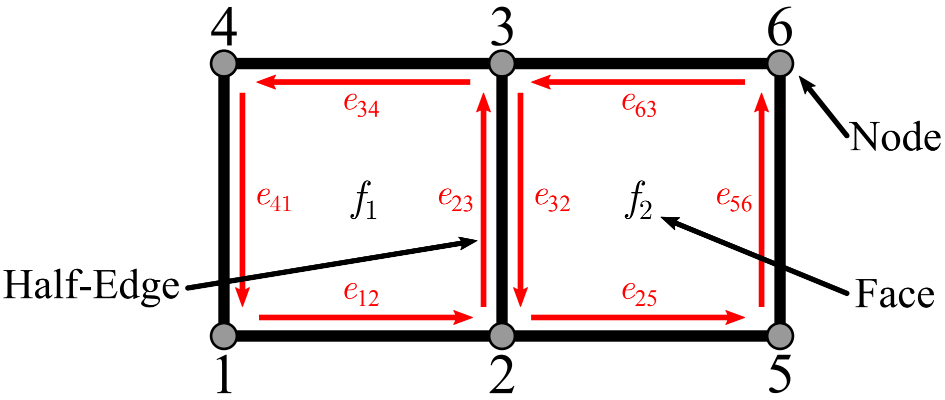


Figure 3.4: Example of a graph I' represented using a half-edge data structure.

where

$$f_1 = [1,2,3,4] \quad \text{and} \quad f_2 = [2,5,6,3]\,.$$

Both f_1 and f_2 are defined as portions of the graph enclosed by a set of half-edges such that as an observer walks along the set of half-edges, the face that these edges bound always lies to the left (assuming half-edge orientations are counterclockwise). Note that the set of edges $E(I')$ is contained within $F(I')$ such that $[1,2]$ represents e_{12}, $[2,3]$ represents e_{23}, etc. The existence of twin half-edges can allow an agent to easily traverse a graph. For example, in graph I' an agent can move freely between f_1 and f_2 by simply switching between the twin half-edges e_{23} and e_{32}.

As this work is concerned with creating a graph for the purpose of structural optimization (cf., Section 3.3), the attribute information contained within the data structure must include a set of material assignments associated with each edge. For the graph I' depicted in Figure 3.4, this set is defined as

$$M(I') = \{[S_1, S_2, S_3, S_4]\,[S_5, S_6, S_7, S_2]\}\,,$$

where $S_1, \ldots, S_7$ represent some material or section functionality assignment. Note that the material associated with the twin half-edges e_{23} and e_{32} remains consistent. Also included within the attribute information are the spatial coordinates associated with each node, which are required when converting the graph into a structural topology (cf., Section 3.3).

Half-edge data structures are used in computational geometry to describe geometric polygonal subdivisions because of the ease with which one can manipulate the topological information (which edges bound a given face, which faces are adjacent, etc.) associated with a planar graph. For instance, returning to the example graph I' and assuming a subdivision of face f_1 between nodes 2 and 4 results in the operation illustrated in Figure 3.5. Graph I' is now defined by the face and

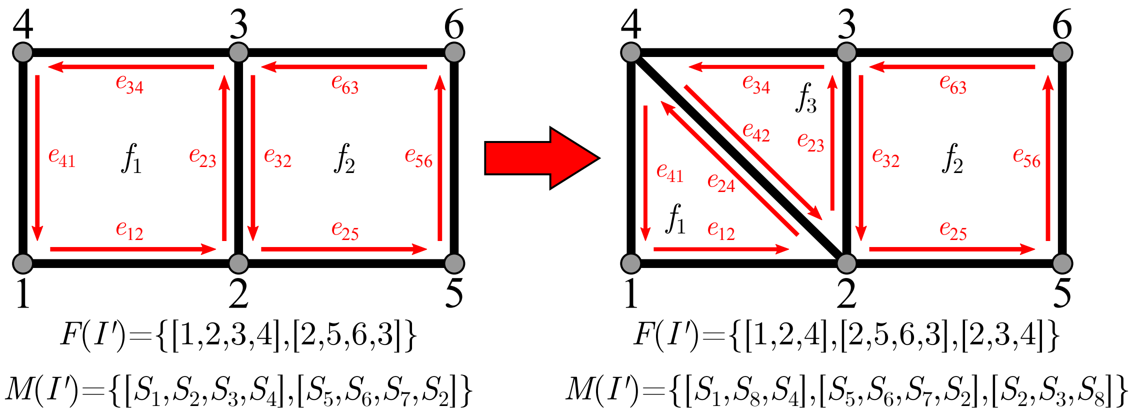


Figure 3.5: Example of how the half-edge data structure treats topological modifications of the graph I' in Figure 3.4. Note that subdividing the face f_1 results in the creation of a new face, f_3.

material sets

$$F(I') = \{[1,2,4]\ [2,5,6,3]\ [2,3,4]\} \quad \text{and} \quad M(I') = \{[S_1,S_8,S_4]\ [S_5,S_6,S_7,S_2]\ [S_2,S_3,S_8]\}\,.$$

Notice that this subdivision results in the creation of a new face f_3 and a new set of twin half-edges, e_{24} and e_{42}. The material set is likewise updated to reflect this change with the material associated with the newly-created half-edges written to remain consistent. Also note that this subdividing behavior and associated modification to the set of faces is only valid if the domain being subdivided is convex, as a half-edge data structure requires a planar graph in which edges only intersect at nodes; if the face f_1 was not convex, there would be a chance that the newly created edge between nodes 2 and 4 would intersect another edge at a location on the graph without a node. Thus, inducing topological changes within a graph is a direct result of simple modifications to the set of faces that define that graph. One can easily see the benefits of such a tool when considering a long, complex string of characters generated by the parameterized L-System approach introduced in Section 3.2.1, as the operation associated with a given character can be efficiently incorporated into a potentially complex graph towards to creation of a final structural topology.

3.2.3 Graphical Operations

Once the encoded parameterized L-System instructions are generated, the SPIDRS algorithm both interprets the instructions and performs graphical operations towards the creation of structure (Figure 3.6). Table 3.1 provides an overview of these SPIDRS graphical operations and their associated parameterized L-System character, along with the number of parameters required by each operation. Recall from Section 3.2.1 that each variable character in $\boldsymbol{\alpha}$ has two parameters and each constant character in $\boldsymbol{\beta}$ has one parameter. If the algorithm encounters an operation in which one of the parameters is not used (i.e., $\mathtt{A}(\sigma_{\alpha_1}, \sigma_{\alpha_2})$ where $\mathtt{A}$ requires only one parameter), the second parameter is simply ignored. The following sections provide a brief overview of each graphical operation as well as an example showing what each operation looks like in the context of the half-edge data structure.

3.2.3.1 Move-Integer Operation

The Move-Integer command is represented by the L-System encoding by $\mathtt{A}(\sigma_{\alpha_1})$ and moves the agent by $\lfloor N \times \sigma_{\alpha_1} \rfloor$[2] node(s) in the current face, where N is the total number of nodes in the current face. In Figure 3.7, assume that the agent begins at node 1 and receives a command

[2] The floor function $\lfloor x \rfloor$ is a function that takes a real number x and returns the greatest integer less than or equal to x.

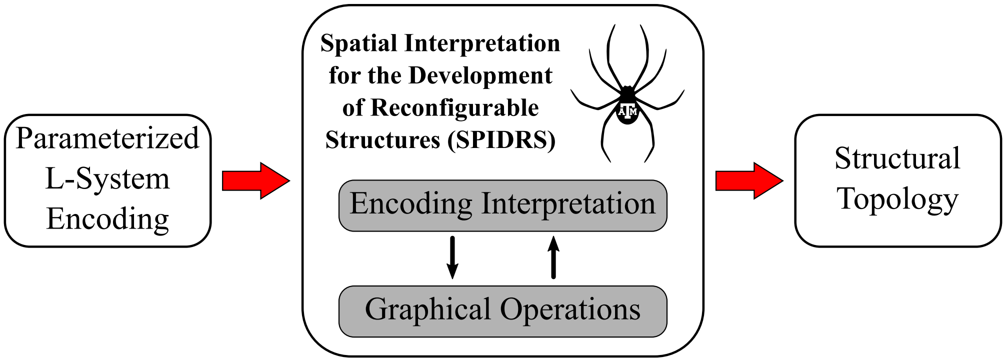

Parameterized L-System Encoding
Spatial Interpretation for the Development of Reconfigurable Structures (SPIDRS)
Encoding Interpretation
Graphical Operations
Structural Topology

Figure 3.6: Illustration of how the SPIDRS framework converts a parameterized L-System encoding into a structural topology. SPIDRS interprets the parameterized L-System instructions and performs graphical operations that modify the topology of the structure.

Table 3.1: Overview of SPIDRS graphical operations and their associated parameterized L-System character.

L-System Character	SPIDRS Operation	No. of Parameters
`A`	Move-Integer	1
`B`	Move-Real	1
`C`	Create Edge-Integer	2
`D`	Create Edge-Real	2
`E`	Change Material	1
+	Move Faces - CW	1
−	Move Faces - CCW	1
[	Save Location	0
]	Return to Location	0

`A`(0.62). Since $N = 5$, $\lfloor N \times \sigma_{\alpha_1} \rfloor = \lfloor 5 \times 0.62 \rfloor = 3$, causing the agent to move three nodes to node 4. Note that this command results only in movement of the agent and does not modify the topology of the graph (i.e., F and M remain unchanged).

3.2.3.2 Move-Real Operation

The Move-Real command is represented by `B`(σ_{α_1}) and moves the agent first by $\lfloor N \times \sigma_{\alpha_1} \rfloor$ node(s) in the current face, and then to a newly created node $N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor$ in between the current node and next node. This command is illustrated in Figure 3.8. Assuming again

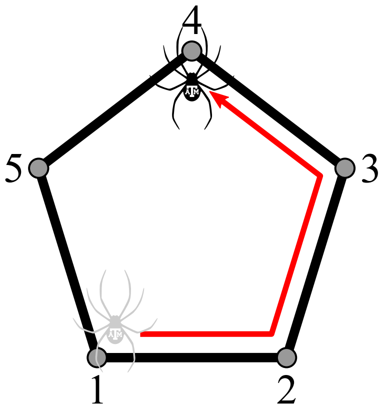


Figure 3.7: Example of a Move-Integer operation `A`(0.62).

that the agent begins at node 1 and now receives a command $\texttt{B}(0.3)$, the agent will first move by $\lfloor N \times \sigma_{\alpha_1} \rfloor = \lfloor 5 \times 0.3 \rfloor = 1$ node. Next, the agent will move $N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor = 1.5 - 1 = 0.5$ of the way in between nodes 3 and 4 and create node 6. The topology of the structure is accordingly modified as follows:

$$F = \{[1,2,3,4,5]\} \rightarrow F = \{[1,2,3,6,4,5]\}\,,$$
$$M = \{[S_1,S_2,S_3,S_4,S_5]\} \rightarrow M = \{[S_1,S_2,S_3,S_6,S_4,S_5]\}\,.$$

Note that since a new node is added to F, a new material assignment must be added to M such that the material along that edge remains consistent (i.e., $S_6 = S_3$). Additionally, since a half-edge has been subdivided (in this example, e_{34} becomes e_{36}, e_{64}), the SPIDRS framework also subdivides the twin of that half-edge if applicable (e.g., if e_{43} exists in the graph it will become e_{46}, e_{63}). This command is restricted such that $0.1 \leq N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor \leq 0.9$ in an attempt to distribute nodes throughout the structure and avoid the formation of smaller features; commands that result in $0.1 > N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor$ or $N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor > 0.9$ are set to 0.1 and 0.9, respectively.

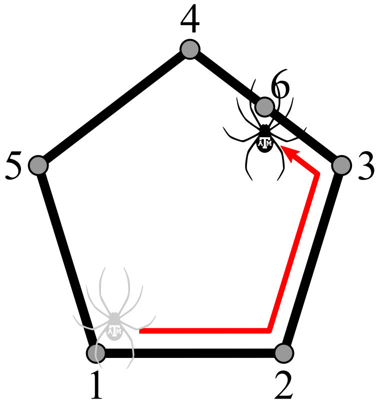


Figure 3.8: Example of a Move-Real operation $\texttt{B}(0.5)$.

3.2.3.3 Create Edge-Integer Operation

The Create Edge-Integer command is represented by $\texttt{C}(\sigma_{\alpha_1}, \sigma_{\alpha_2})$ and causes the agent to move by $\lfloor N \times \sigma_{\alpha_1} \rfloor$ node(s) in the current face before constructing an edge back to its original location, thus subdividing the face. The parameter σ_{α_2} determines what material will be assigned to the newly created edge. Figure 3.9 shows an example of the agent starting at node 1 and receiving the command $\texttt{C}(0.62, 0.75)$, where $\sigma_{\alpha_2} = 0.75$ corresponds to the material S_6 (in a manner consistent with that discussed in Section 3.3). As in Section 3.2.3.1, the agent moves three nodes to node 4 before constructing a new edge back to its starting location (node 1). The topology of the structure is then modified as follows:

$$F = \{[1,2,3,4,5]\} \rightarrow F = \{[1,2,3,4][1,4,5]\}\,,$$
$$M = \{[S_1,S_2,S_3,S_4,S_5]\} \rightarrow M = \{[S_1,S_2,S_3,S_6][S_6,S_4,S_5]\}\,.$$

After subdividing the graph, the agent's location is defined to be along its original half-edge (i.e., since the agent originated along the half-edge e_{12}, after executing this command it will remain in the face that contains that half-edge). This command is ignored if moving by $\lfloor N \times \sigma_{\alpha_1} \rfloor$ node(s) results in the agent reaching a node that is collinear with its original node, as this can result in overlapping faces and material assignments. Collinearity is tracked using a graph-based method based on additions and/or modifications to the initial set of edges.

3.2.3.4 Create Edge-Real Operation

The Create Edge-Real command is represented by $\texttt{D}(\sigma_{\alpha_1}, \sigma_{\alpha_2})$ and results in the agent first moving by $\lfloor N \times \sigma_{\alpha_1} \rfloor$ node(s) in the current face and then to a newly created node $N \times \sigma_{\alpha_1} - \lfloor N \times \sigma_{\alpha_1} \rfloor$ in between the current node and next node, before finally constructing an edge back to its original location. The parameter σ_{α_2} again determines what material will be assigned to this new edge. As illustrated in Figure 3.10, assume that the agent begins at node 1 and receives a command of $\texttt{D}(0.3, 0.47)$, where $\sigma_{\alpha_2} = 0.47$ corresponds to the material S_7. As in Section 3.2.3.2,

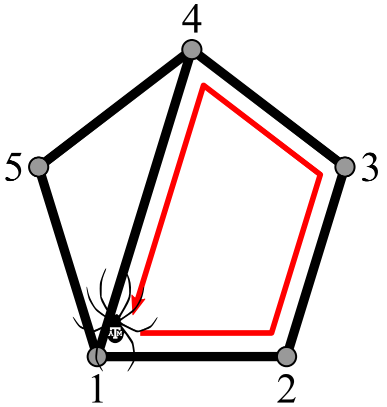


Figure 3.9: Example of a Create Edge-Integer operation $\mathtt{C}(0.62, 0.75)$.

the agent moves one node to node 3 and then creates node 6 halfway in between nodes 3 and 4. The agent then creates a new edge from node 6 back to its starting location (node 1). Thus, the topology of the structure is modified per:

$$F = \{[1, 2, 3, 4, 5]\} \rightarrow F = \{[1, 2, 3, 6][1, 6, 4, 5]\}\,,$$
$$M = \{[S_1, S_2, S_3, S_4, S_5]\} \rightarrow M = \{[S_1, S_2, S_3, S_7][S_7, S_6, S_4, S_5]\}\,.$$

As before, the location of the agent after edge creation is defined to be along its original half-edge. Additionally, because a half-edge has been subdivided, the SPIDRS framework also subdivides the twin of that half-edge if applicable as described in Section 3.2.3.2. Restrictions are placed both on the ability to create nodes within a certain distance of existing nodes (cf., Section 3.2.3.2) and the creation of edges between collinear nodes (cf., Section 3.2.3.3).

3.2.3.5 Change Material Operation

The Change Material command is represented by $\mathtt{E}(\sigma_{\alpha_1})$ and changes the material associated with the agent's current half-edge to the material specified by the parameter σ_{α_1} before moving the agent by one node as shown in Figure 3.11. In this example, the agent originates at node 1 and half-edge e_{12} with assigned material S_1 (designated by black lines). When it receives the command

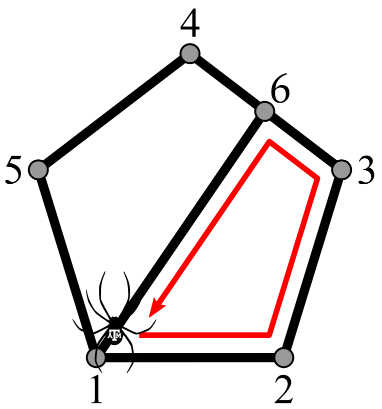


Figure 3.10: Example of a Create Edge-Real operation $\mathtt{D}(0.5, 0.47)$.

$\mathtt{E}(0.21)$, where $\sigma_{\alpha_1} = 0.21$ corresponds to the material H_1 (designated by grey lines), the agent changes that material of half-edge e_{12} to H_1 before moving to node 2. While this command does not change the set F, the set M is modified simply per:

$$M = \{[S_1, S_2, S_3, S_4, S_5]\} \rightarrow M = \{[H_1, S_2, S_3, S_4, S_5]\}.$$

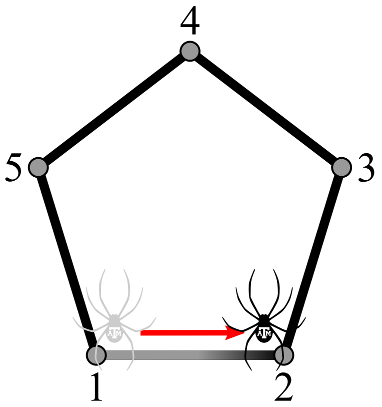
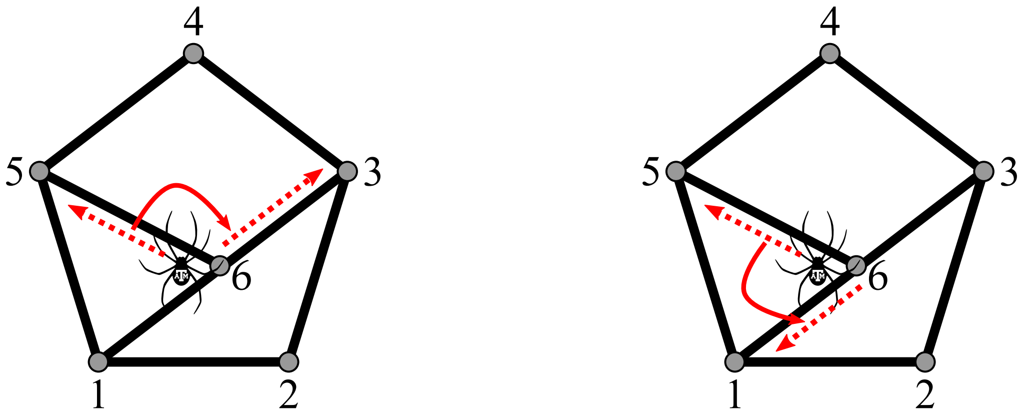


Figure 3.11: Example of a Change Material operation $\mathtt{E}(0.21)$.

3.2.3.6 Turning Operations

Turning operators are represented by $+(\sigma_{\beta_1})$ and $-(\sigma_{\beta_1})$ for clockwise and counterclockwise turns, respectively, and are used to navigate between faces in the graph. These commands are based on the number of faces M associated with the agent's current node. The agent will then move $\lfloor M \times \sigma_{\beta_1} \rfloor$ faces in the direction specified by the command. Examples are shown in Figure 3.12, where the agent begins at node 4 along the half-edge e_{45}. In Figure 3.12a, the agent receives the command $+(0.38)$. Given that node 4 helps define four distinct faces, the agent will move $\lfloor M \times \sigma_{\beta_1} \rfloor = \lfloor 4 \times 0.38 \rfloor = 1$ face in the clockwise direction. Figure 3.12b depicts the agent receiving the command $-(0.38)$; as before, the agent will move 1 face, but this time in the counterclockwise direction. After each turning operation, the agent orients itself in the new face along the half-edge that contains its original location and remains consistent with its counterclockwise movement around the face (e.g., e_{49} in Figure 3.12a, e_{43} in Figure 3.12b). If the agent's current node is only associated with one face, this command is ignored.



(a) +(0.38).

(b) −(0.38).

Figure 3.12: Examples of (a) clockwise and (b) counterclockwise turning operations.

3.2.3.7 *Bracket Operations*

Bracket operators are represented by `[` and `]` for the beginning and ending, respectively, of a particular string of agent commands. An open bracket saves the current location of the agent, which then executes any commands that follow; a closed bracket returns the agent to the most recently saved location (without constructing any additional edges on the graph). An example of this operation is shown in Figure 3.13. Assume that the agent receives the command `[` is at node 1 along the half-edge e_{12}. This location is saved by the agent, which then completes a series of graphical operations as defined by the parameterized L-System until it receives the command `]` when located at node 4 along the half-edge e_{45}. At this point, the agent will return to the most recently saved location and orient itself along the same half-edge. Note that brackets can be nested. Because it is possible that the graph-based definitions of saved locations can be modified by graphical operators (e.g., creating a node 7 in between nodes 1 and 2 would mean that the half-edge e_{12} no longer exists), after each operation the algorithm updates saved locations to reflect the current state of the graph.

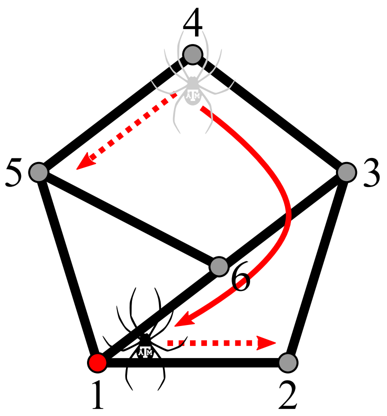


Figure 3.13: Example of a bracket operation `[...]`.

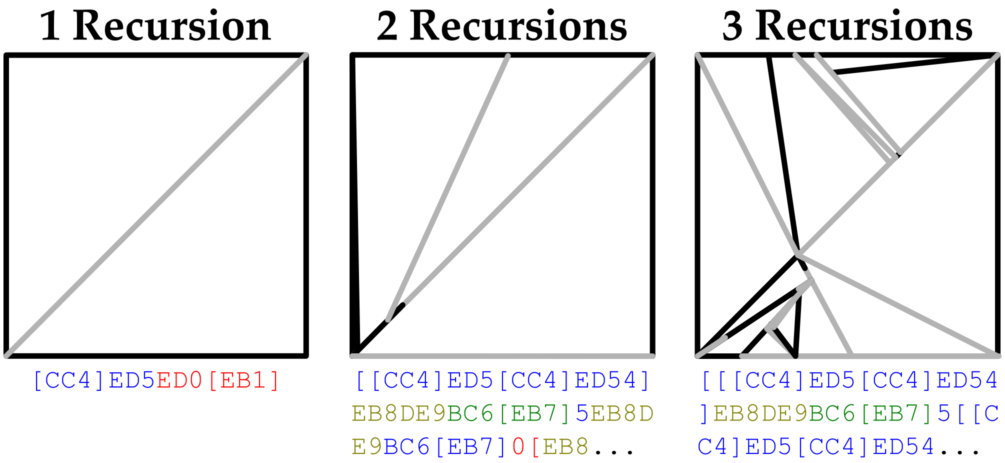


Figure 3.14: Example of structural topology generation using a SPIDRS interpretation of the parameterized L-System example in Figure 3.3.

3.2.4 SPIDRS-Interpreted Topologies

The SPIDRS algorithm takes a parameterized L-System encoding, performs a graphical operation associated with each character in the encoding, manipulates the topological information of the graph based on that graphical operation using the half-edge data structure, and returns a final topology. Example topologies generated by the SPIDRS algorithm using the parameterized L-System example in Figure 3.3 are shown in Figure 3.14. The colors associated with the edges represent material assignments and serve to highlight the amount of design freedom offered by the SPIDRS algorithm, particularly the Change Material operator.

3.3 Structural Analysis and Design Framework

To generate optimal mechanically-driven multifunctional structures, the parameterized L-System and SPIDRS interpreter, which consider 92 design variables for multi-material problems, are coupled with an FEA package and genetic algorithm in the same manner outlined in Section 2.1.3. As before, the 42 discrete variables associated with ω_i and λ_j^i introduced in Section 3.2.1 are mapped evenly onto a real number line in the semi-open interval $[0, 1)$. The remaining 50 variables associated with the continuous parameters in the set Σ are likewise mapped to the same interval via consideration of their lower and upper bounds. The structural analysis framework continues to use the Abaqus FEA suite, which takes the final topology of the graph and converts each edge of the graph into a 2-D beam element. The section assignment associated with each edge is defined by the material set contained within the data structure of the graph. The efficient parallelization approach, post-processor, and constraint violation discussed in Section 2.1.3 are also utilized here.

The subdividing behavior of SPIDRS precludes the need for the boundary constraints and trimming algorithm required by the geometry-based L-System interpretation approach (cf., Appendix A-B). An initial graph that the agent can traverse but never cross will define the boundary of the design domain, and the edge creation operations in the SPIDRS algorithm are written such that free-ended edges cannot be formed. However, these are replaced by a constraint which ensures minimal material overlap when converting the graph to a structure consisting of 2-D frame elements. This constraint is expanded upon in Appendix D. The full multiobjective optimization process when considering a graph-based interpretation of L-System encodings is shown in Figure 3.15, which illustrates the changes made to the design framework from that shown in Figure 2.3.

3.4 Design Optimization Examples

To demonstrate the ability of the parameterized L-System and SPIDRS interpreter to generate effective structural configurations given multiple objective functions, two distinct design problems are presented. The first problem revisits the single-material cantilevered frame problem introduced in Section 2.2.1, while the second problem considers a dual-material inverting compliant mechanism problem similar to that discussed in Section 2.2.2. For both problems, the structural design approach is summarized, the optimization problem is quantified, and results are presented in the form of Pareto frontiers and associated example topological configurations. As in Section 2.2, these design problems consider materials based on those associated with the Stratasys Objet 500 “PolyJet” 3-D printer, the experimentally measured properties of which can be found in Table 2.1.

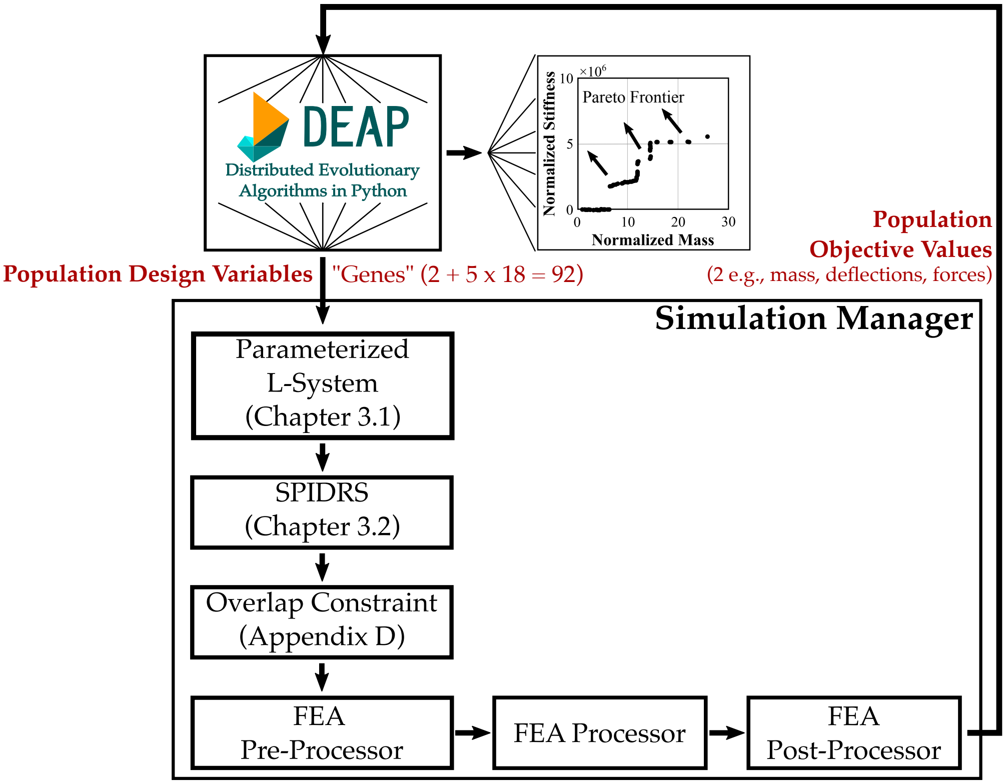

DEAP
Distributed Evolutionary Algorithms in Python
Normalized Stiffness
10
5
0
×10⁶
Pareto Frontier
0
10
20
30
Normalized Mass
Population Objective Values
(2 e.g., mass, deflections, forces)
Population Design Variables
"Genes" (2 + 5 x 18 = 92)
Simulation Manager
Parameterized L-System (Chapter 3.1)
SPIDRS (Chapter 3.2)
Overlap Constraint (Appendix D)
FEA Pre-Processor
FEA Processor
FEA Post-Processor

Figure 3.15: Flowchart indicating the overall genetic programming topology optimization framework for multifunctional structures when considering a graph-based interpretation of L-System encodings.

3.4.1 Cantilevered Frame

The first design example considers a simple cantilevered frame design with boundary conditions and initial graph information illustrated in Figure 3.16. The initial structure and subsequently generated structural members consist of stiff beam elements with an assumed rectangular cross-section of 2 mm × 20 mm (thicker dimension out of plane). As dictated by the initial structure, the frame has a maximum length of 300 mm and a maximum total height of 350 mm. Geometric symmetry is assumed about the x-z plane such that only half the topology is generated. A displacement boundary condition of $u_x = u_y = 0$ is enforced ±50 mm above/below the axis of symmetry at the minimum x position, and a concentrated force of $f_y = -0.05$ N is applied to the tip of the frame. The initial state of the graph is defined by $F = \{[1, 2, 3, 4, 5]\}$; note that because

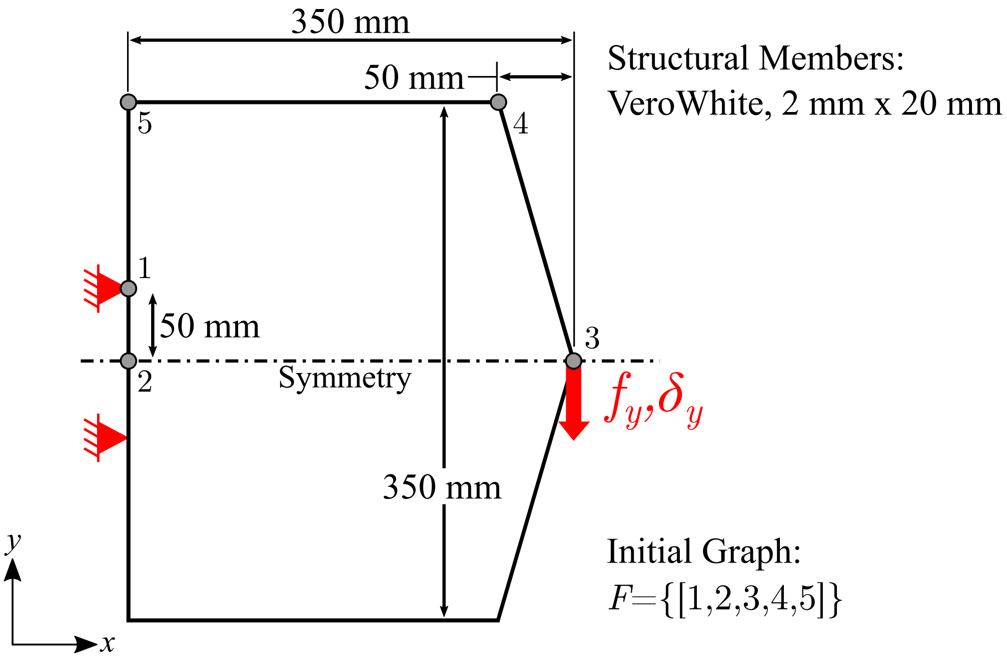


Figure 3.16: Initial graph and boundary conditions associated with the cantilevered frame design study.

the problem only considers a single material, the set of material assignments for each edge is not recorded. Also, note that edges oriented along the axis of symmetry (i.e., e_{23} and all associated subdivisions thereof) are removed from the final graph topology.

Details regarding the multiobjective optimization of the structural frame are shown in Table 3.2. The objective of reduced mass favors designs with a sparse topology, while the objective of increased stiffness tends toward dense topologies. Given the single-material definition of the problem, the change material graphical operation of the SPIDRS algorithm (cf., Section 3.2.3.5) is not considered; thus, here the parameterized L-System framework requires 74 independent design variables. As in Section 2.2.1, a fully linear FEA model assuming plane strain conditions (i.e., there is no strain in the z direction) and an average element length of 30 mm is considered. The continued use of the efficient parallelization approach discussed in Section 2.1.3 means that $100 \times 1000 = 100,000$ designs can be analyzed relatively quickly.

The Pareto frontier associated with the multiobjective optimization described in Table 3.2 is

Table 3.2: Specifications for the multiobjective topology optimization problem of a light and stiff structural frame using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Minimize (Maximize):	normalized mass (normalized stiffness)
by varying:	2 axiom characters, 4 rule assignments (18 genes each),
subject to:	no constraints
NSGA-II Parameters [68]	
100 members for 1,000 generations, P_{cross} = 0.9, η_{cross} = 20, P_{mut} = 1/52, η_{mut} = 20	

shown in Figure 3.17 along with three example topologies generated by the SPIDRS algorithm. The mass and stiffness of each topology are normalized by the derived properties of the "null" design (i.e., made up of only the initial graph), which is the lightest and least stiff design possible (mass of 0.057 kg and stiffness of 5.98 Pa). The stiffest and heaviest topology features a comparatively dense distribution of structural members. It is also interesting to observe the design on the frontier closest to the utopia point, which has a normalized mass of 1.32 and a normalized stiffness of 4.95×10^3.

Also shown in Figure 3.17 is the frontier associated with a series of Michell trusses (cf., Figure 2.6) as well as the frontier generated when interpreting L-System encodings using the turtle graphics algorithm. The designs in each frontier are represented as a series of beam elements with the same material properties and constant rectangular cross-section as described earlier. Despite analyzing 100,000 topologies, the parameterized L-System framework interpreted by the SPIDRS algorithm is unable to match the linear performance of a Michell truss. However, the SPIDRS algorithm does show marked improvement on topologies generated using the turtle graphics algorithm and does in fact generate designs that are topologically similar or even identical to a Michell truss as illustrated in Figure 3.17. Additionally, it is important to note that the motivation for the use of SPIDRS is the complex task of designing nonlinear *reconfigurable* structures for which analytically-derived solutions do not exist.

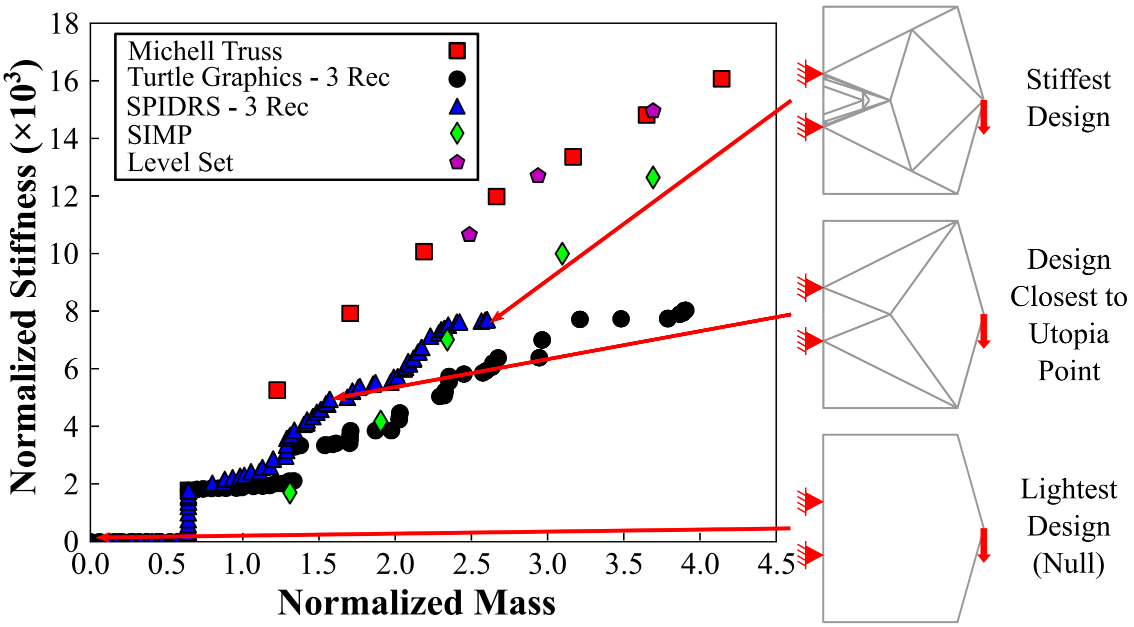
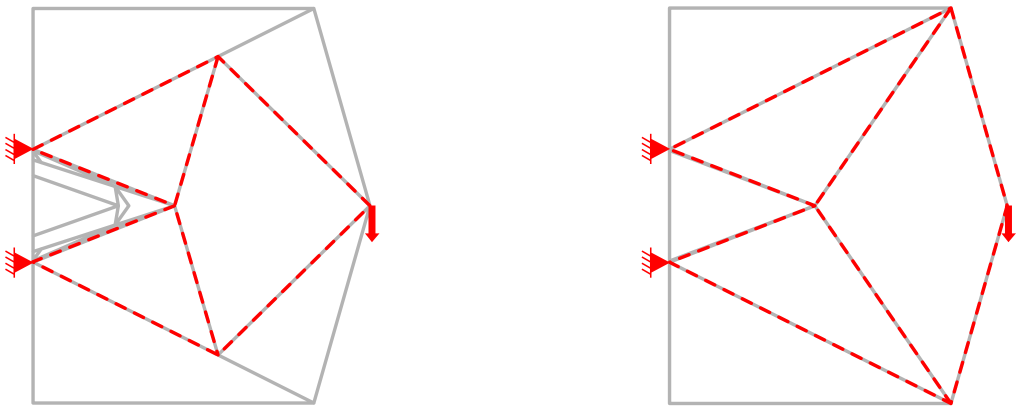


Figure 3.17: Pareto frontier of the cantilevered frame design problem generated using SPIDRS-interpreted L-System encodings, along with comparisons to the frontiers associated with a Michell truss, geometry-based L-System interpretation, SIMP implementation, and level set implementation. Note that the mass and stiffness of the initial graph (cf., Figure 3.16) has been added to these frontiers.

Figure 3.17 also compares the frontier generated using the SPIDRS algorithm with frame structures generated using the SIMP and level set implementations discussed in Section 2.2.1 and shown in Figure 2.7. These implementations consider a design problem identical to that shown in Figure 3.16 and also do not consider displacement in the z-direction. In both cases, the design domain is discretized into unit elements (i.e., each element measures 1 mm×1 mm). The generation of these Pareto frontiers is accomplished by simply increasing the permissible volume fraction. Comparison of these two methods with the proposed parameterized L-System/SPIDRS framework yields similar conclusions to those stated in Section 2.2.1: i) the Pareto frontier generated by SPIDRS could be improved if structural members were allowed to have varying cross-section like the SIMP and level set methods (this will be addressed in Chapter 5), ii) topologies generated using the SPIDRS algorithm consistently demonstrate improved performance at lower normalized



(a) Stiffest SPIDRS-generated topology.

(b) SPIDRS-generated topology closest to the utopia point.

Figure 3.18: Several structures generated using the SPIDRS algorithm are topologically similar to a two-bar Michell truss as indicated by the dashed line.

mass values over those generated using the SIMP method, and iii) topologies obtained using the level set method fail to converge at lower volume fractions, indicating that this method may not be capable of representing long, slender structural members.

It is also interesting to compare the computation time needed for all three topology optimization methods. The five designs generated using the SIMP method required an average of 190 iterations and an average computation time of 7.99 minutes, while the three generated using the level set method required an average of 103 iterations and an average computation time of 1.88 minutes. For comparison, a single objective genetic algorithm utilizing the SPIDRS algorithm with an initial population of 100 individuals required 53 generations and a total computation time of 37 minutes. However, one of the benefits of the proposed framework is its ability to easily explore multiobjective problems and generate a frontier of potential solutions. The SPIDRS-generated Pareto frontier consisting of 100 topologies shown in Figure 3.17 required 15.22 hours of computation time. Generating a similar frontier with 100 distinct topologies would require 13.32 and 3.183 hours for the SIMP and level set methods, respectively, based on the results mentioned earlier. Based on the resulting topologies shown in Figures 2.7 and Figure 3.17, one could also argue that utilizing the SPIDRS algorithm would result in a greater amount of topological diversity, which is crucial when considering complex multiobjective optimization problems where the main goal is the exploration of the design space. All analyses were performed on a computer with 132 GB of RAM and up to 12 total cores available.

3.4.2 Compliant Mechanism: Tensile Inverter

The second design problem illustrated in Figure 3.19 considers a tensile inverter mechanism similar to that discussed in Section 2.2.2. The initial graph is defined as $F = \{[1, 2, 3, 4]\}$ and $M = \{[T, V, T, T]\}$, where T indicates the compliant TangoBlack material and V indicates the stiff VeroWhite material (cf., Table 2.1). All material assignments assume a rectangular cross-section of 2 mm × 10 mm. As in Section 2.2.2, a y-z symmetry condition is enforced along the x-axis, allowing for consideration of only half of the structural topology. A displacement boundary condition of $u_x = \delta_{x,in}$ is applied to node 1, while another displacement boundary condition of $u_x = u_y = 0$ is placed on node 4 of the graph. Unlike the previous tensile inverter design problem, here a spring with a constant $k = 1$ N/mm is modeled as being attached to node 2 to ensure that the framework finds physically meaningful designs capable of producing motion under load.

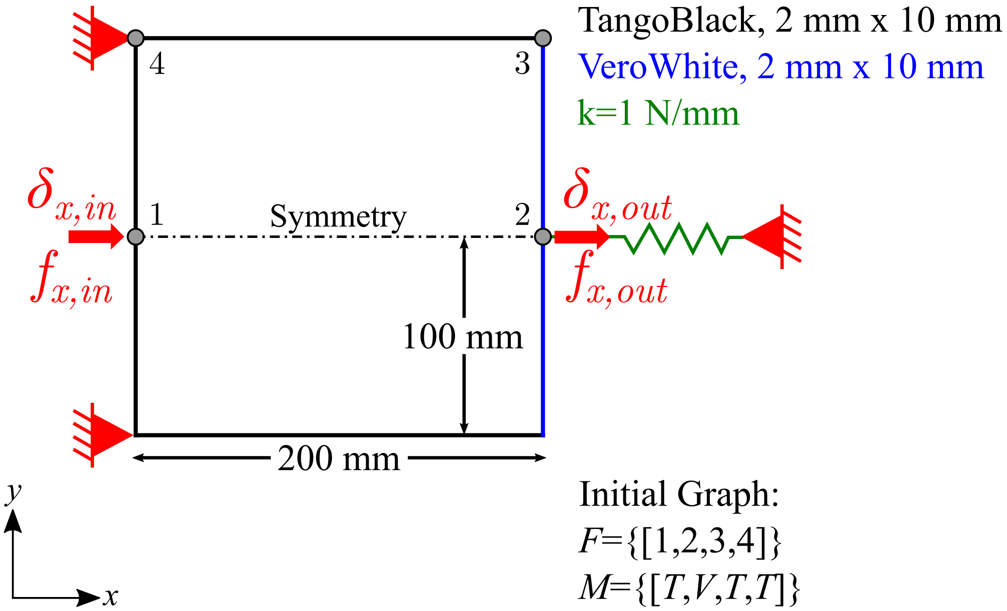


Figure 3.19: Initial graph and boundary conditions associated with the tensile inverter design study.

This spring constant is chosen based on experiences with the kinematic rectifier design problem discussed in Section 2.2.3. As in Section 3.4.1, edges oriented along the axis of symmetry (i.e., e_{12} and all associated subdivisions thereof) are removed from the final graph topology.

A summary of the multiobjective design optimization problem is shown in Table 3.3. This design study uses every SPIDRS graphical operation discussed in Section 3.2.3 and thus requires a total of 92 independent design variables. To determine a mechanism topology that maximizes both positive output displacement and force, two distinct loading steps are considered. First, a displacement of $\delta_{x,in} = -5\,\mathrm{mm}$ is applied to node 1, while the resulting output displacement $\delta_{x,out}$ at node 2 is measured when only the spring load is present. This step provides an indication of displacement inversion, which here is defined as

$$r_\delta = \left.\frac{\delta_{x,out}}{\delta_{x,in}}\right|_{f_{x,out}=k\delta_{x,out}} . \tag{3.1}$$

Next, node 2 is forced back into its reference position (i.e., $\delta_{x,out} = 0$) and the reaction forces at nodes 1 and 2 ($f_{x,in}$ and $f_{x,out}$, respectively) are measured. This step provides an indication of blocked force inversion, which is defined as

$$r_f = \left.\frac{-f_{x,out}}{f_{x,in}}\right|_{\delta_{x,out}=0} . \tag{3.2}$$

Both analysis steps consider the full geometry nonlinearity of the solution (i.e., internal buckling, large rotations, etc.). The possibility of buckling in particular requires the use of implicit dynamic analysis and thus small loading increments. As in Section 2.2.2, a stress constraint in stiff structural members of $\sigma_{Mises}^{max} = 33.8\,\mathrm{MPa}$ is imposed to ensure the structural feasibility of potential configurations.

The resulting Pareto frontier after considering 60 members for 400 generations is shown in Figure 3.20, along with several designs of interest. It is interesting to note that, as in Section 2.2.2, the design with the best displacement inversion behavior ($r_\delta = 2.69$) features predominantly VeroWhite, further indicating that the displacement response of a mechanism relies primarily upon

Table 3.3: Specifications for the multiobjective topology optimization problem of tensile inverting compliant mechanism using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Maximize:	r_{δ}, r_f
by varying:	2 axiom characters,
	5 rule assignments (18 genes each),
subject to:	$\sigma_{Mises}^{max} \leq 33.8\,\text{MPa}$
NSGA-II Parameters [68]	
60 members for 400 generations,	
P_{cross} = 0.9, η_{cross} = 20,	
P_{mut} = 1/52, η_{mut} = 20	

the buckling or bending behavior of the structural material. Comparing this topology to those generated by the turtle graphic algorithm and traditional SIMP approach (cf., Figure 2.12) reveals that, while there are noticeable differences in topology, once again the underlying geometries associated with the load path are roughly identical (Figure 3.21). Specifically, the action of the input displacement causing a rigid body rotation near the top of the structure and subsequent conversion of that rotation into a translation at node 4 is very similar in principle to the solution found using the traditional SIMP approach [104].

To further investigate the performance of the topologies lying on the Pareto frontier, consider the work efficiency of each mechanism as defined by Equation 2.5 in Figure 3.20. The low efficiency of mechanisms with high force inversion ratios (and low displacement inversion ratios) are easily explained, as these designs are unable to noticeably displace the spring. Designs with high displacement inversion ratios have slightly higher efficiencies, but these values are still approximately $\eta \approx 0.65$ due to the storage of substantial strain energy within the structures to facilitate the buckling of structural members. The design with the highest efficiency ($\eta = 0.76$) is shown in Figure 3.20 and makes use of several compliant members, allowing for less strain energy to be stored inside the structure. The deformed configuration of this high-efficiency topology is shown in Figure 3.22.

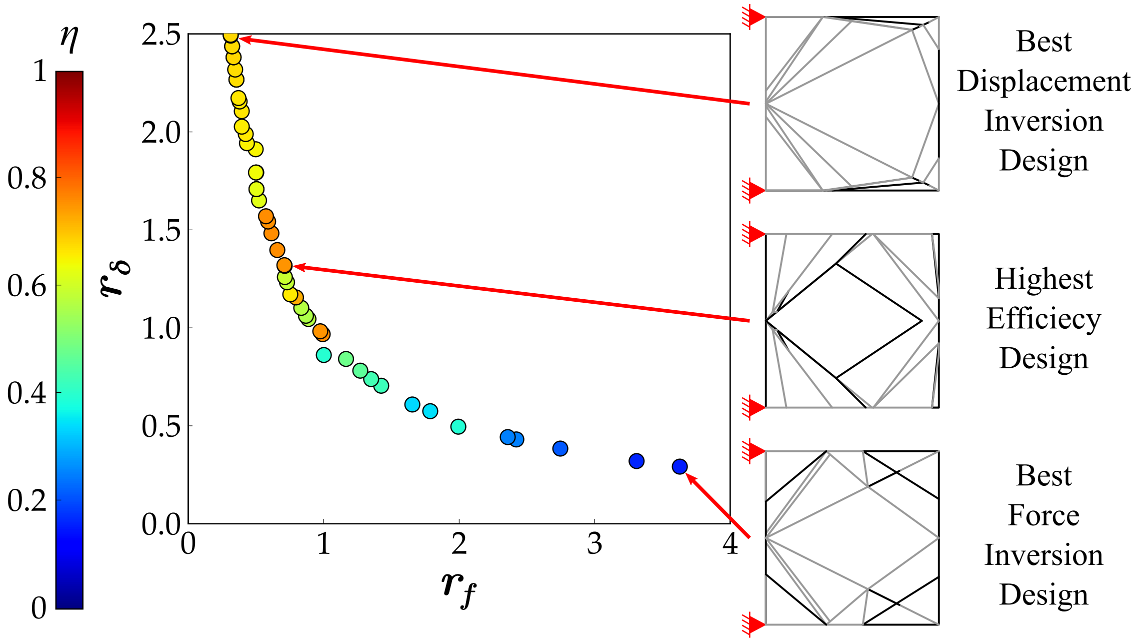
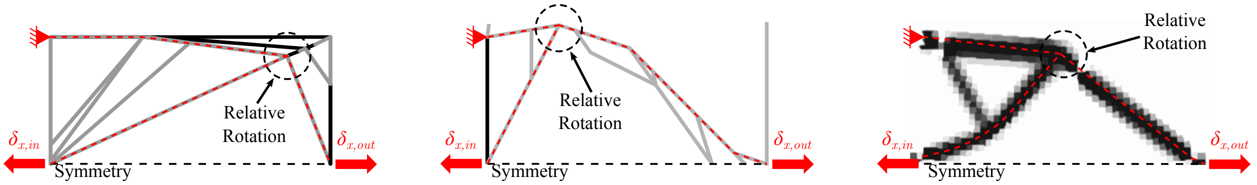


Figure 3.20: Pareto frontier of the tensile inverter design problem generated using SPIDRS-interpreted L-System encodings.



(a) One of many SPIDRS-generated tensile inverters.

(b) One of many turtle graphics-generated tensile inverters.

(c) Single tensile inverter generated using traditional subtractive topology optimization (SIMP) [104].

Figure 3.21: Comparison of geometry and topology of inverting mechanisms generated by both L-System interpretation methods as well as a traditional SIMP implementation. Once again, the underlying geometries (denoted by the dashed red lines) associated with the load paths are equivalent.

3.5 Extensions of the SPIDRS Algorithm

The graph-based approach of the SPIDRS algorithm also allows added design flexibility in the form of several extensions, which will be discussed in this section. The first extension the

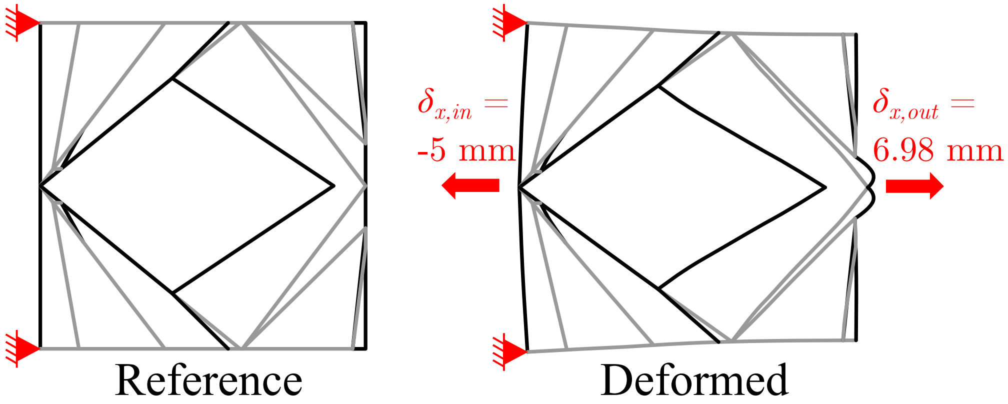


Figure 3.22: Comparison of the reference and deformed configurations of the high-efficiency tensile inverter generated by the SPIDRS framework.

be considered is the ability to include the shape of the initial in the optimization problem, which permits the algorithm to explore configurations that further optimize the performance of a given topology. The second extension involves increasing the structural dimensionality of the topology, whereby specific L-System encodings allow faces of the graph to be represented by a planar solid. This allows the SPIDRS algorithm to be applied to design problems where branched topologies alone may be unsuitable. The theory and implementation of these extensions are detailed, and their ability to generate topologies capable of achieving multiple simultaneous design goals are demonstrated using the tensile inverter design problem discussed in Section 3.4.2.

3.5.1 Shape Optimization

The development of structural topologies using L-System formulations relies upon initial structural boundaries which serve to both define the design space and ensure the existence of a load path when no meaningful structure is created. These boundaries are predetermined based on a knowledge of the design problem and expected geometry of potential optimal solutions. In the SPIDRS algorithm, these boundaries are even more important, as the initial graph serves as a starting-point for face-subdivision and can undergo changes in material assignments. Thus, the initial graph is considered a part of the overall structure and predetermining the shape of this graph limits the ability of the algorithm to fully explore the design space. Here, it will be shown that leveraging the graph-based foundation behind the SPIDRS algorithm can allow for the inclusion of *shape optimization*, or the determination of the optimum shape of a domain, inside the overall topology optimization framework.

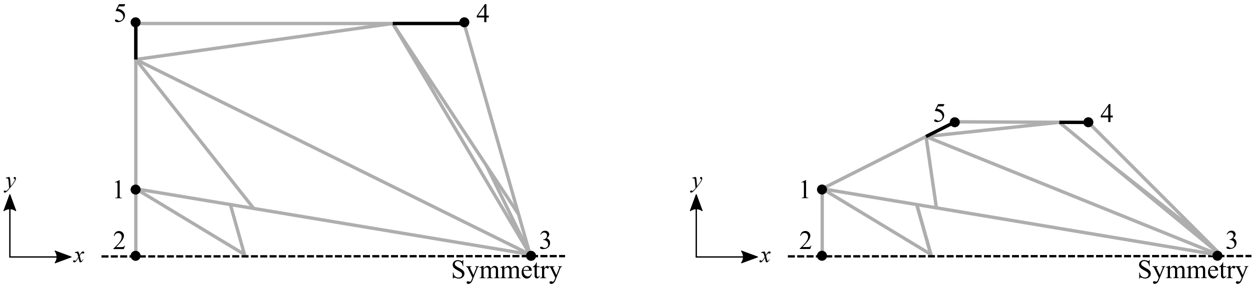
As was discussed in Section 3.1, the incidence function ψ and corresponding diagram of a graph merely represents connectivity between pairs of nodes. Nodes and edges are given no spatial significance, and thus there is no unique way of drawing a given graph [108] as illustrated in Figure 3.2. In the context of this work, spatial significance is assigned to the graph inside the SPIDRS algorithm by a set of spatial coordinates v that define the position of each node that makes up the initial graph. New nodes are defined based on their position relative to the nodes making up the initial graph and all subsequently created nodes rather than by their own individual spatial coordinates. Therefore, by changing the coordinates of the initial boundary it is possible to represent the same graph, and by extension the same structural topology, in a number of different geometric configurations, some of which may be more advantageous in satisfying certain performance objectives. This operation takes the form

$$b(v) : S(\boldsymbol{\omega}, \boldsymbol{P}, \boldsymbol{\Sigma}) \rightarrow \Omega(v, \boldsymbol{\omega}, \boldsymbol{P}, \boldsymbol{\Sigma}), \tag{3.3}$$

where b is a mapping associated with a set of initial graph nodal coordinates v, S is the graphical information generated by the parameterized L-System variables discussed in Section 3.2.1 for an arbitrary set of initial graph nodal coordinates, and Ω is the resulting structural information for the set v. The mapping b is said to be *homeomorphic* because it preserves topological equivalence between S and Ω [113].

As an example, assume a set of coordinates v_1 for an initial graph consisting of five nodes and a specified parameterized L-System encoding that creates the graphical information S. When S is coupled with the mapping $b(v_1)$, we obtain structural topology Ω_1 shown in Figure 3.23a. Now, assume a second set of coordinates v_2 for the same five nodes in the initial graph. Coupling S with a new mapping $b(v_2)$ results in the structural topology Ω_2 shown in Figure 3.23b. Notice that topological equivalency is maintained between both structural topologies, but the shape of the overall structure and positions/lengths of structural members have changed. The repositioning of these members could result in better overall performance, such as better stress distribution in frame structures, greater transmission of motion in compliant mechanisms, more favorable flow conditions in fluid problems, etc. Any modification to the initial boundary must ensure that i) the domain remains convex as required by the half-edge data structure (cf., Section 3.2.2) and ii) nodes where boundary conditions are specified remain stationary (e.g., nodes 1, 2, and 3 in Figure 3.23).

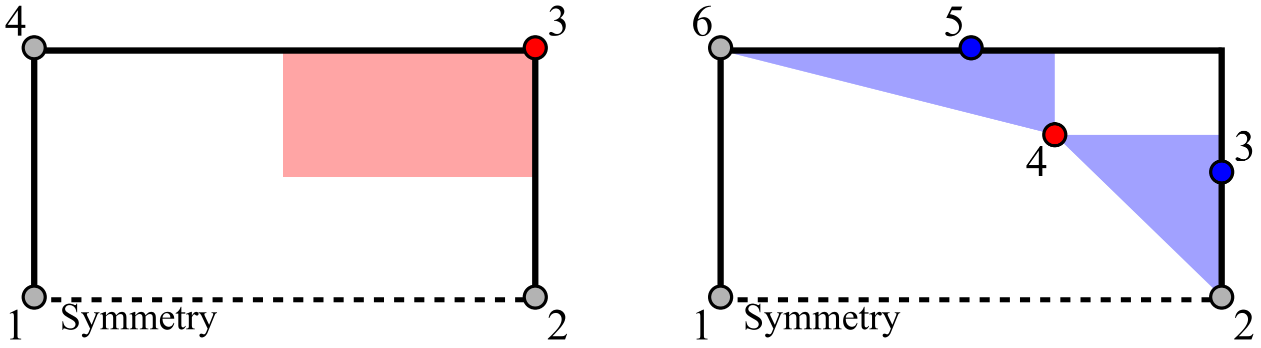
To demonstrate the ability of the SPIDRS framework to incorporate shape optimization into a topology optimization problem, two distinct cases based on the tensile inverter design problem discussed in Section 3.4.2 are presented. The first case considers the optimization of a single nodal location based on two design variables in both the x- and y-coordinates, leading to a total of 94 design variables. In the example shown in Figure 3.24a, this parameterization allows for node 3 to be placed anywhere within the shaded area. The second case considers the optimization of three nodal locations, which requires four more additional design variables for a total of 98. Here, the coordinates of node 4 are determined in the same manner illustrated in Figure 3.24a. The coordinates of nodes 3 and 5 are then calculated as a parameterization of the coordinates of nodes 2 and 4 and nodes 4 and 6, respectively. This parameterization ensures that the initial graph remains



(a) Structural topology Ω_1 associated with the set of initial graph coordinates v_1.

(b) Structural topology Ω_2 associated with the set of initial graph coordinates v_2.

Figure 3.23: Illustration of how changing the spatial definition of the initial graph results in modified structural configurations while maintaining the same topology.



(a) Single-node case.

(b) Three-node case.

Figure 3.24: Illustration of the two shape optimization cases considered.

convex, which is required by the half-edge data structure. Once the coordinates of all three nodes are calculated, an additional check is performed to ensure that the angle formed by nodes 3, 4, and 5 preserves that convexity of the domain. Non-convex initial graphs are penalized as discussed in Section 2.1.3.

The Pareto frontiers associated with both shape optimization cases after considering 60 individuals over 400 generations are shown in Figure 3.25 along with several topologies of interest. Also depicted in Figure 3.25 is a SPIDRS-generated Pareto frontier when considering a fully-fixed initial graph (cf., Section 3.4.2). Clearly, the addition of an initial graph shape optimization component to the overall topology optimization problem results in improved mechanism performance (i.e., for a given force inversion ratio r_f, topologies generated using the combined shape and topology optimization approach generally have an increased displacement inversion ratio r_δ). Additionally, both cases converge towards configurations (shown in Figure 3.26a-b) that feature the same underlying geometries associated with the load path that are generated using SIMP, turtle graphics, and SPIDRS with fixed initial graphs (cf., Figure 3.21). The deformed configurations for the best displacement inversion designs generated during both shape optimization cases are shown in Figure 3.27a-b.

The incorporation of shape optimization inside the overall topology framework also allows for the observation of trends in generated topological solutions that can better inform designers as to

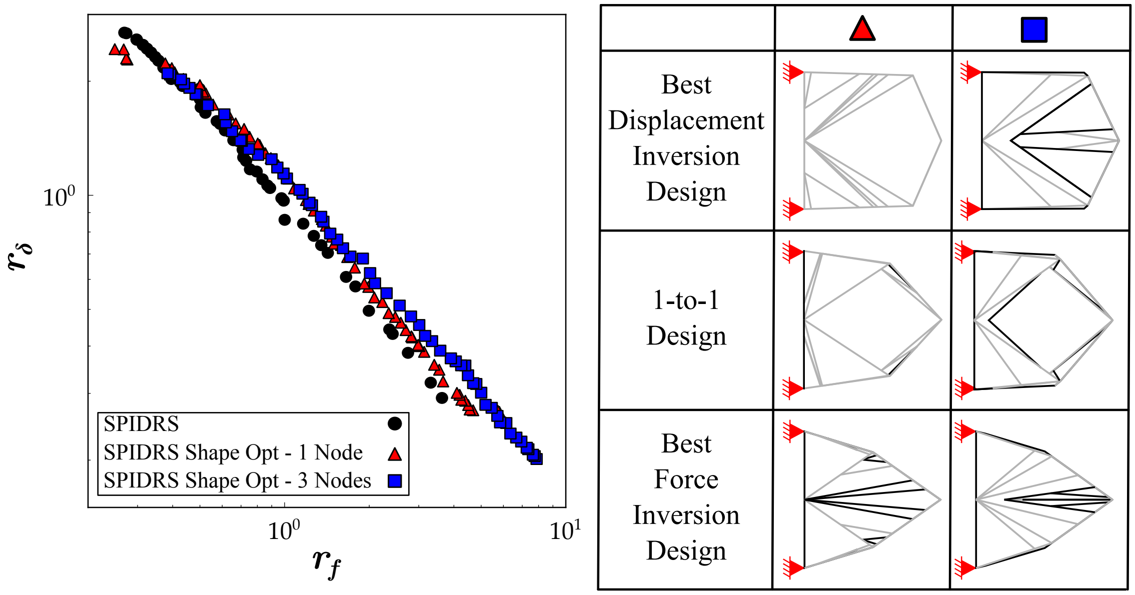
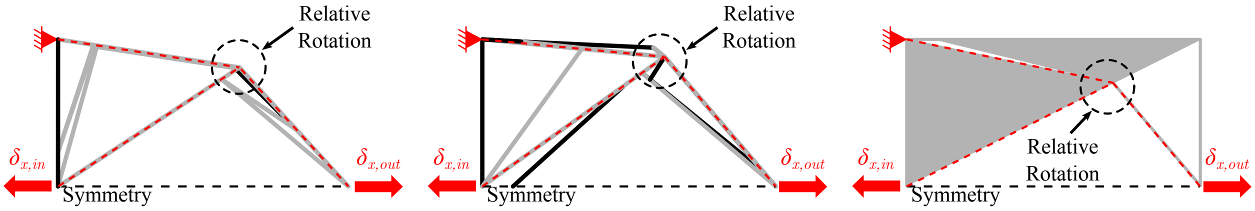


Figure 3.25: Pareto frontier and several topologies of interest for both SPIDRS shape optimization cases, along with a comparison to the SPIDRS-generated Pareto frontier using a fully-fixed initial graph (cf., Section 3.4.2).



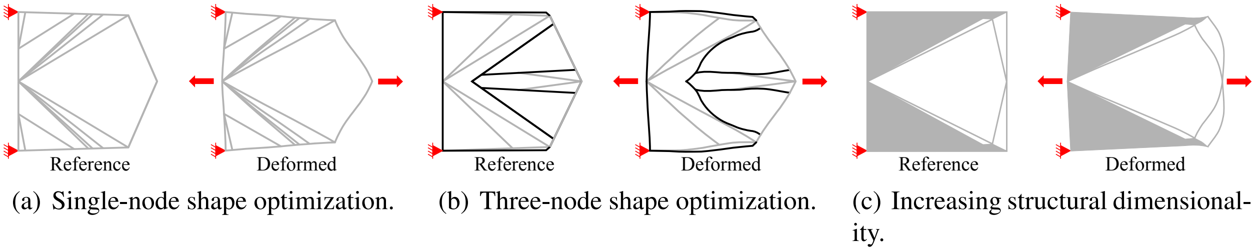
(a) Single-node shape optimization.

(b) Three-node shape optimization.

(c) Increasing structural dimensionality.

Figure 3.26: Underlying load path geometry generated by the SPIDRS algorithm for each test case. This geometry is consistent with those illustrated in Figure 3.21. Notice that after shape optimization the domain boundary has conformed to this geometry (a-b), while when using a fixed domain boundary the geometry must result from the generated topology (c).

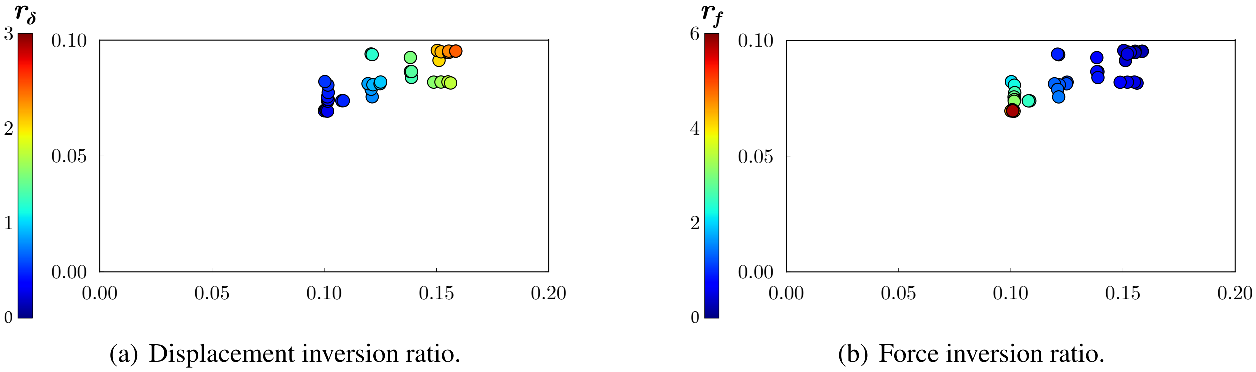
how to solve various problems. For example, Figure 3.28 shows the position of node 3 for the single-node shape optimization case along with the associated displacement and force inversion ratios. It is immediately noticeable that there is an obvious relationship between the placement of node 3 and the corresponding inversion ratios. Specifically, to obtain higher displacement inversion



(a) Single-node shape optimization. (b) Three-node shape optimization. (c) Increasing structural dimensionality.

Figure 3.27: Comparison of reference and deformed configurations of the best displacement inversion tensile inverters generated by the SPIDRS algorithm for each test case.

ratios node 3 needs to be positioned up and to the right in the design domain in order to maximize the amount of rotation about this point generated by $\delta_{x,in}$. This rotation leads to a higher magnitude of $\delta_{x,out}$ and thus a higher displacement inversion ratio. As node 3 is moved down and to the left in the design domain, the amount of rotation generated about this point is intuitively reduced, resulting in lower displacement inversion ratios. However, configurations with node 3 closer to node 1 are better equipped to facilitate the conversion of the input displacement into a higher magnitude of $f_{x,out}$ and therefore a high force inversion ratio (as opposed to designs where node 3 is closer to node 4, which convert the input displacement into a higher magnitude of $f_{y,out}$ and therefore have a low force inversion ratio). These trends are also evident when looking at the resulting topologies

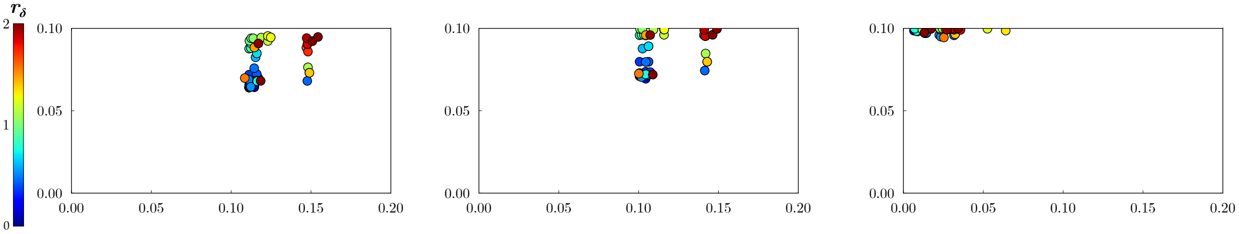


(a) Displacement inversion ratio. (b) Force inversion ratio.

Figure 3.28: Position of node 3 for the single-node shape optimization case along with associated a) displacement inversion and b) force inversion ratios.

for the SPIDRS-generated fully-fixed initial graph case illustrated in Figure 3.20.

The positions of nodes 3, 4, and 5 in the three-node optimization case along with the associated displacement inversion and force inversion ratios are shown in Figures 3.29 and 3.30, respectively. While similar trends to those seen in Figure 3.28 are also seen here, they are not as conclusive. However, it is interesting to note that, in general, the optimization results in geometries where nodes 3 and 4 are positioned extremely close to one another. Furthermore, notice that node 5 tends to be positioned close to node 6. This would seem to indicate that the optimizer is trying to turn the three-node shape optimization case into a one-node optimization case where nodes 3 and 4 and nodes 5 and 6 are being “merged” together. Results such as these could help inform designers about what is going on in the problem and help identify certain design domain geometries or

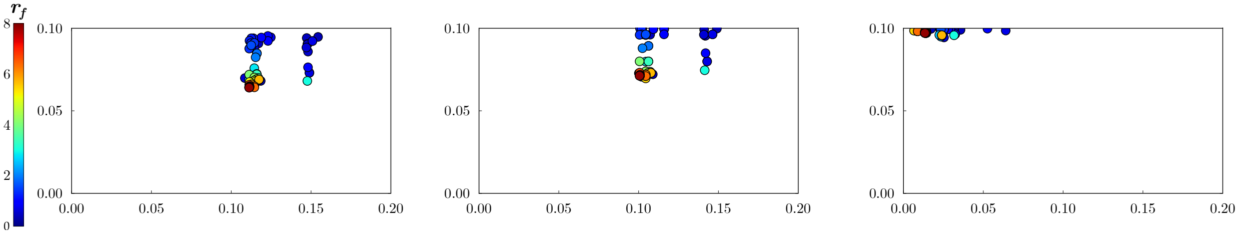


(a) Node 3.

(b) Node 4.

(c) Node 5.

Figure 3.29: Position of nodes a) 3, b) 4, and c) 5 for the three-node optimization case along with the associated displacement inversion ratios.



(a) Node 3.

(b) Node 4.

(c) Node 5.

Figure 3.30: Position of nodes a) 3, b) 4, and c) 5 for the three-node optimization case along with the associated force inversion ratios.

configurations that merit additional study.

3.5.2 Increasing Structural Dimensionality

As has already been established, the SPIDRS algorithm describes the topological evolution of a structure primarily as a series of face subdivisions inside the initial graph. Previous studies have considered these faces to be void regions of the structure, meaning that the structural topology is defined solely by the edges and associated material assignments that make up each face. However, there is no reason why the faces of the graph cannot also have an assigned material. This can be thought of as an increase of the structural dimensionality of the topology, as the topology is now defined by 2-D planar solids as well as 1-D lines. This concept requires an additional set of material assignments, this one associated with each face and denoted as M_F, to be contained within the half-edge data structure.

To facilitate material reassignments to each face, a new graphical operation is introduced. The Change Face Material command is represented by the L-System encoding $\texttt{F}(\sigma_{\alpha_1})$ and changes the material associated with the agent's current face to the material specified by the parameter σ_{α_1} before moving the agent by one node as shown in Figure 3.31. In this example, the agent originates at node 1 in face f_2 with assignment material S_1 (designated by empty space). When it receives the command $\texttt{F}(0.96)$, where $\sigma_{\alpha_1} = 0.96$ corresponds to the material H_1 (designated by grey space), the agent changes the material assigned to face f_2 before moving to node 2. While this command does not change the sets F or M, the set M_F is modified simply per:

$$M_F = \{[S_1, S_1]\} \rightarrow M_F = \{[S_1, H_1]\}$$

Note that the same rules discussed surrounding the subdivision of edges are also valid for the subdivision of faces (i.e., if face f_2 in Figure 3.31 was further subdivided the material assignment of the newly-created face would be defined such that $M_F(f_2) = M_F(f_3)$). This operation could serve as a bridge between traditional density-based topology optimization methods (e.g., SIMP) and the L-System method proposed above, combining both solid and branched structural members

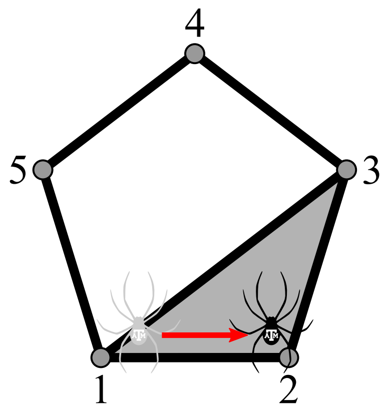


Figure 3.31: Example of a Change Face Material operation $\texttt{F}(0.96)$.

to achieve optimal performance. Such a topology could allow for the application of L-System topology optimization in multiphysical design problems where branched topologies alone may be unsuitable (i.e., designing for specific fluid flow conditions, lower resistance in an electric circuit, etc.).

The Pareto frontier associated with the tensile inverter problem discussed in Section 3.4.2 when considering the addition of the Change Face Material graphical operation for 60 individuals over 400 generations is shown in Figure 3.32, along with the SPIDRS-generated frontier associated with a fully-fixed initial graph. With the addition of this extra graphical operation (resulting in a total of 110 design variables), the SPIDRS algorithm is able to discover topologies that feature higher magnitudes of displacement inversion or force inversion than those obtained when using only slender structural members. This is due to the stiffness of the solid domains relative to the slender structural members causing these domains to behave essentially as rigid bodies. Thus, the magnitudes of an displacements and/or rotations transmitted through these bodies are higher than those of an identical body consisting of only slender members. Note that each topology shown in in Figure 3.32 also features the same underlying load path geometry previously discussed and highlighted in Figure 3.26c. The deformed configuration for the best displacement inversion design is illustrated in Figure 3.27c.

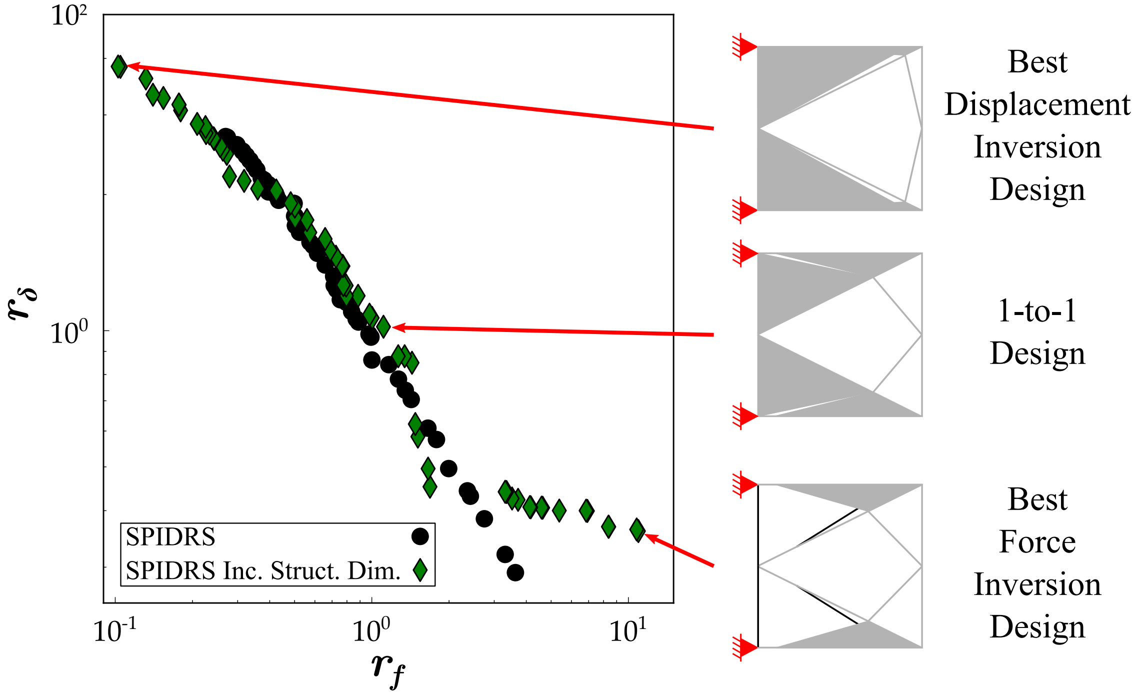


Figure 3.32: Pareto frontier associated with the addition of the Change Face Material graphical operation in the SPIDRS algorithm. The inclusion of large, solid material domains results in the generation of topologies that feature increased magnitudes and displacement or force inversion.

3.6 Considerations for Even Distributions of Structural Members

Despite its demonstrated effectiveness when considering compliant mechanism design problems (Section 3.4.2) and the added design freedom that its graph-based foundations can provide (Section 3.5), the performance of the SPIDRS algorithm when considering the cantilevered frame design problem discussed in Section 3.4.1 is noticeably lacking. As illustrated in Figure 3.17, while the Pareto frontier generated using SPIDRS outperforms the frontier associated with the turtle graphics algorithm, it is still unable to compete with the performance of a Michell truss. Topologies generated using SPIDRS do compare favorably to those generated using SIMP in regards to performance, but due to the lack of complexity of these topologies these comparisons are only valid at relatively low normalized mass values.

To enable a more thorough comparison between SPIDRS and SIMP, the cantilevered truss problem discussed in Section 3.4.1 was considered using four parameterized L-System recursions in an attempt to generate more complex topologies with higher normalized mass values. However, it was found that with four recursions Abaqus was unable convert the SPIDRS-generated graph into an FEA model. This is believed to be due to the fact that, as the example graphs in Figure 3.33 demonstrate, with four recursions the nodes and edges of the final graph are positioned too closely to one another to resolve. Furthermore, it is noticeable that topological modifications tend to be concentrated in specific portions of the graph, which results in the formation of "slivered" faces (i.e., faces with high aspect ratios) that not only may not be necessary in load-bearing structures, but also violate the material overlap check outlined in Appendix D. In general, it stands to reason that more diverse topologies (and more importantly load paths) will be discovered when topological modifications/structural members are spread more evenly throughout the graph. However, once a well-performing load path is identified, the creation of structural members should still be allowed in these portions of the graph in an attempt to further improve the performance of the topology. While theoretically one could use the shape optimization approach discussed in Section 3.5.1 to find initial graph geometries for each graph that result in a better distribution of structural members, in general this is not practical. The remainder of this section will analyze several aspects of both the SPIDRS algorithm and the topology optimization framework as a whole to determine whether slight changes can result in a methodology that favors a more even distribution of topolog-

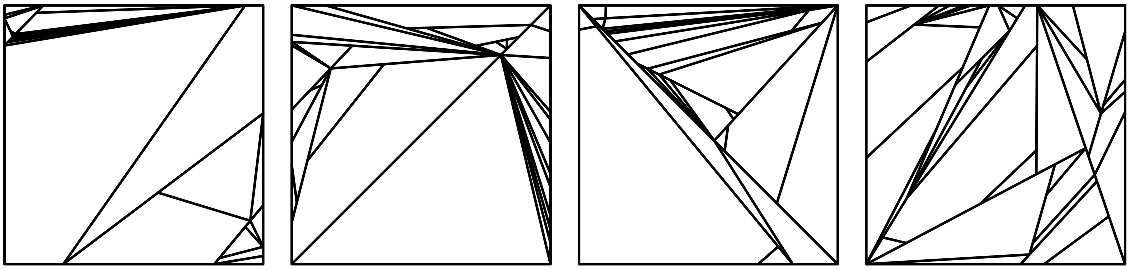


Figure 3.33: Example of final graphs when considering four parameterized L-System recursions. Notice that topological modifications are concentrated in specific portions of the graph, leading to a unequal distribution of structural members and the formation of slivered faces.

ical modifications while not completely eliminating the possibility of concentrations of structural members when such behavior is desirable.

3.6.1 Including Exterior Faces in the Graph

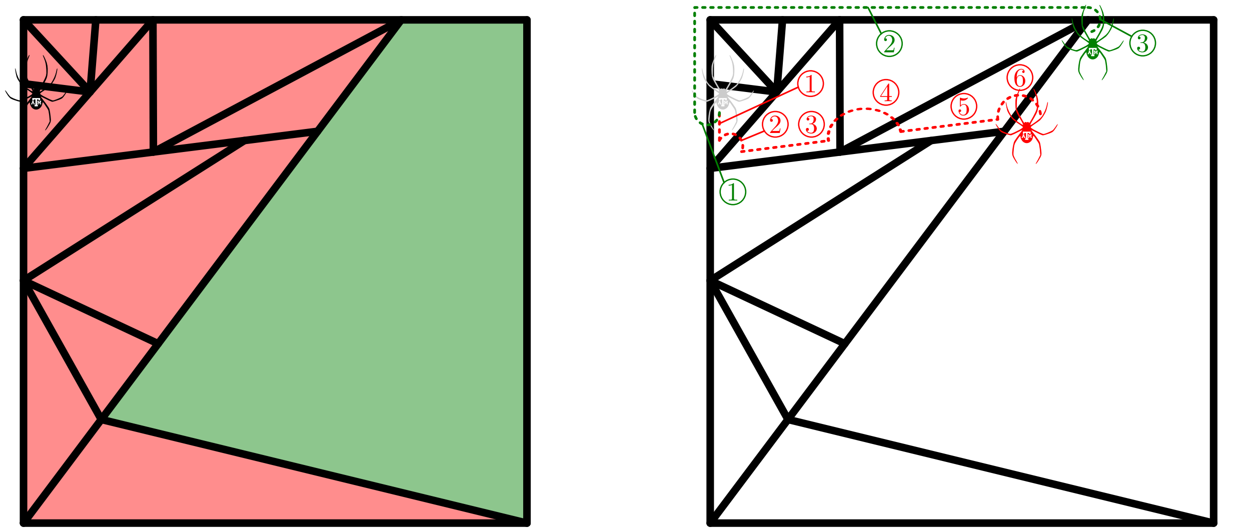
Recall from Section 3.2.2 that half-edges on the exterior of the initial graph have been neglected to this point. This decision was made both to help simplify the formatting of the half-edge data structure and based on the assumption that the turning operations introduced in Section 3.2.3.6 would be sufficient in allowing the agent to traverse the entire graph. However, this idea is now revisited based on the finding that with large numbers of recursions topological modifications tend to be concentrated in specific portions of the graph.

As previously stated, it is intuitive that a generally even distribution of structural members resulting from topological modifications is desired for a better preliminary exploration of the topological design space. For example, in the graph illustrated in Figure 3.34a, it is obvious that the portions of the graph denoted by red have been fairly well-developed and that the agent should focus additional modifications in the green portion of the graph which is relatively sparse. However, given the position of the agent relative to the rest of the graph, it would require a specific, complex sequence of graphical operations for the agent to reach this area. Specifically, as shown in Figure 3.34b, it would take a minimum of six movement operations to achieve this without taking into account additional movements and/or structure creation in intermediate faces. However, if the agent was allowed to turn onto and then traverse the exterior half-edges of the graph, it could reach the same face in a minimum of just three operations (Figure 3.34b).

Including the exterior half-edge, and by extension the exterior face, in a graph is relatively simple. Returning to the example shown in Figure 3.4, the sets of faces and materials associated with graph I' are rewritten as

$$F(I') = \{[1,4,3,6,5,2]\,[1,2,3,4]\,[2,5,6,3]\} \quad \text{and}$$

$$M(I') = \{[S_4,S_3,S_7,S_6,S_5,S_1]\,[S_1,S_2,S_3,S_4]\,[S_5,S_6,S_7,S_2]\}\,.$$

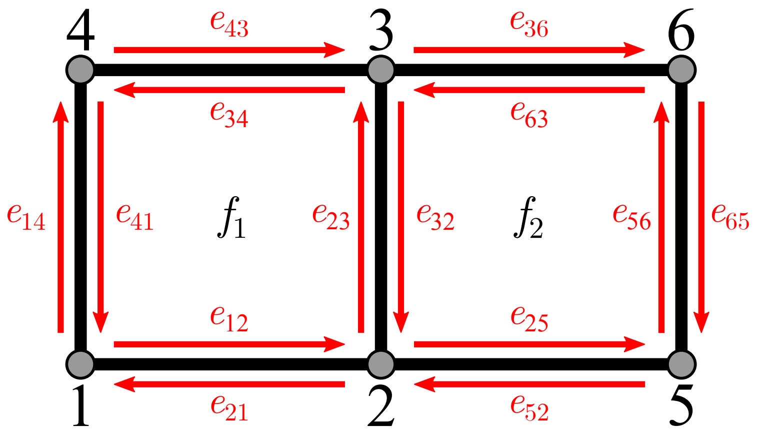


(a) Given its current state, one can say that the portions of the graph in red are fairly well-developed and that the agent should focus additional modifications on the green portion.

(b) Allowing movement along the exterior face of the graph means the agent can move to a more diserable portion of the graph in 3 graphical operations as opposed to a minimum of 6.

Figure 3.34: Example of how allowing the agent to move along the exterior face of the graph can result in a better distribution of topological modifications.

This change is illustrated in Figure 3.35. Notice that the consideration of the exterior face does require eliminating the assumption that a given half-edge is oriented counterclockwise relative to the face it belongs to. In terms of graphical information, the exterior face is treated in a manner consistent with all other faces (i.e., materials associated twin half-edges are consistent, modifications to a face due to graphical operations in neighboring faces are tracked and updated, etc.). All graphical operations that deal solely with movement around the graph (e.g., Move-Integer, Move-Real, turning and bracket operations) also remain the same. However, operations that deal with structure creation and/or material reassignment are modified to reduce computational complexity and preserve the half-edge data structure. Specifically, when operating on the exterior face the Create Edge-Integer and Create Edge-Real operations are interpreted as Move-Integer and Move-Real, respectively (i.e., $\mathtt{C}(\sigma_{\alpha_1}, \sigma_{\alpha_2}) \rightarrow \mathtt{A}(\sigma_{\alpha_1})$ and $\mathtt{D}(\sigma_{\alpha_1}, \sigma_{\alpha_2}) \rightarrow \mathtt{B}(\sigma_{\alpha_1})$). The Change Material operation is also modified such that agent may move by one node along the exterior face but does



$$F(I')=\{[1,4,3,6,5,2],[1,2,3,4],[2,5,6,3]\}$$

$$M(I')=\{[S_4,S_3,S_7,S_6,S_5,S_1],[S_1,S_2,S_3,S_4],[S_5,S_6,S_7,S_2]\}$$

Figure 3.35: Example of graph I' represented using a half-edge data structure in which half-edges on the exterior of the initial graph are considered.

not change the material associated with that half-edge.

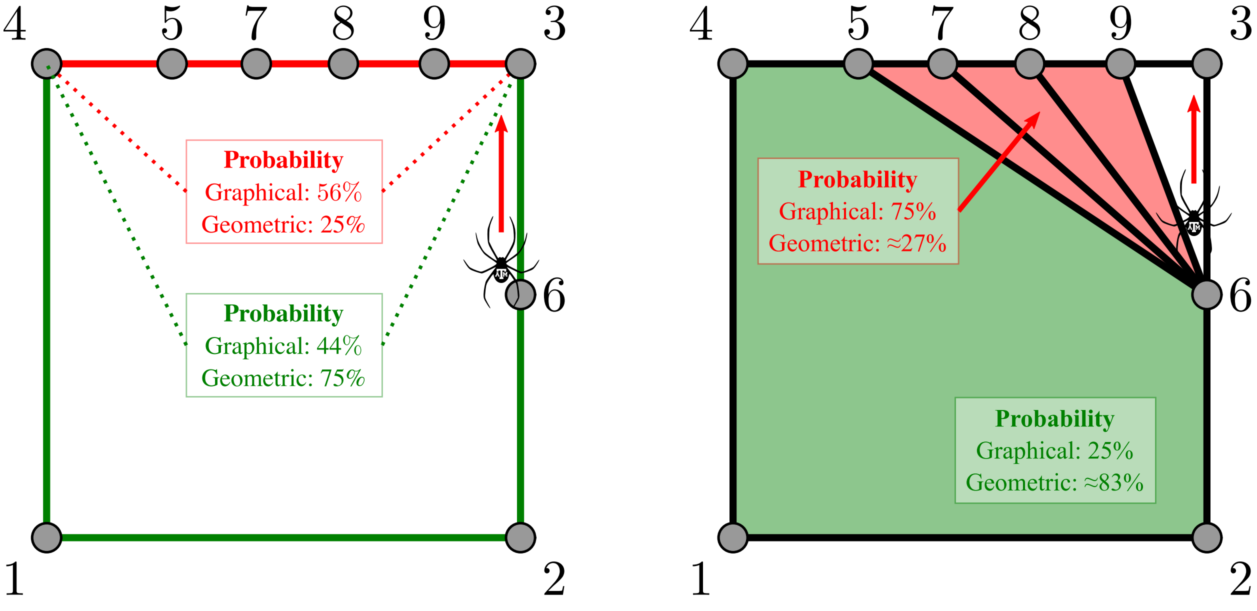
3.6.2 Graphical vs Geometric Parameterization of Graphical Operations

When developing SPIDRS, special care was taken to ensure that the algorithm remained strictly graph-based. This is due in part to experiences with the turtle graphics algorithm, whose geometry-based approach severely limited the modeling power of an L-System and made it impossible to match the performance of mathematically-proven optimal solutions (cf., Section 2.2.1.1). As such, SPIDRS graphical operations were written as a function of *only* the information associated with the graph. Movement around a face was based on the number of nodes contained in that face, and turning operations were based on the number of faces associated with the agent's current node. However, further inspection shows that this *graphical parameterization* of SPIDRS operations actually biases the agent toward concentrating topological modifications in specific portions of the graph.

To demonstrate the drawbacks of graphical parameterizations, consider the examples shown in Figure 3.36. First taking the graph in Figure 3.36a and assuming that the agent is currently at node

6 and oriented along half-edge e_{63}, it is desired that the agent move or create edges anywhere on the face except the associated half-edges between nodes 3 and 4 (i.e., e_{37}, e_{79}, e_{98}, e_{85}, or e_{54}), which are denoted by a red line. This portion of the graph features a dense distribution of nodes relative to the rest of the graph, which can lead to concentrations of structural members as the topology of the graph continues to evolve. Portions of the graph denoted by green lines, however, are sparse and underdeveloped. As discussed in Sections 3.2.3.1-3.2.3.4, the in-face movement and structural creation operations parameterize movement using the function $\lfloor N \times \sigma_{\alpha_1} \rfloor$, where N is the number of nodes in the current face. Applying this parameterization to the graph in Figure 3.36a, there is a 44% probability that the agent's operation will involve nodes 1, 2, 6, or 4 and a 56% probability of its operation involving nodes 3, 9, 8 ,7 or 5. As the graph continues to be modified, these probabilities will continue to become more and more uneven.

The turning operations defined in SPIDRS also suffer from this parameterization as illustrated



(a) In-face movement example.

(b) Out-of-face movement example.

Figure 3.36: Example of probabilities of a agent moving to specified portions of a graph using both graphical and geometric parameterization. Incorporating the geometry of the graph into the parameterization of SPIDRS operations can bias the agent away from areas that have already been modified.

in Figure 3.36b. Once again, assume the agent is currently at node 6 and oriented along half-edge e_{63}. In this graph, edges have also been created between node 6 and nodes 5, 7, 8, and 9. Here, it is intuitive that to better distribute topological modifications throughout the entire graph, the agent should move to the face defined by $[1, 2, 6, 5, 4]$ (denoted in green). The parameterization of the turning operations developed in Section 3.2.3.6 is defined as $\lfloor M \times \sigma_{\beta_1} \rfloor$, where M is the number of faces the agent could potentially turn into. Given that there are four possible faces for the agent to turn into, there is a 25% probability of the agent moving to face $[1, 2, 6, 5, 4]$ and a 75% probability of the agent moving to one of the slivered faces. As before, the more complex the graph, the more uneven these probabilities will become.

To eliminate this biasing and better distribute topological modifications throughout the graph, a *geometric parameterization* of SPIDRS operations is proposed. Graphical operations remain a function of the nodes, edges, and faces of the graph and function in the same manner as described in Section 3.2.3. However, decisions as to which node or face the agent moves to are determined based on the current geometry of the graph rather than the number of nodes or faces available. Specifically, in-face movement of the agent is defined by solving for the number of nodes N^* that the agent should move by in the combined inequality

$$\frac{\sum_{i=1}^{N^*} l_i}{L} \leq \sigma_{\alpha_1} < \frac{\sum_{i=1}^{N^*+1} l_i}{L},$$

where l_i is the length of the half-edge between nodes i and $i + 1$ and L is the total length of all half-edges in the current face. These inequalities serve to reduce (but not eliminate) the probability of the agent moving to areas of the graph that feature large concentrations of nodes, as the odds that nodes in close proximity with one another relative to the rest of the face will satisfy the compound inequality are relatively low. For the Move- and Create Edge-Integer operations (cf., Sections 3.2.3.1 and 3.2.3.3), the agent moves by N^* nodes in its current face and moves to the next operation as defined by the L-System encoding. For the Move-and Create Edge-Real operations (cf., Sections 3.2.3.2 and 3.2.3.4), the agent moves by N^* nodes in its current face, and then to a newly created node $1 - (L\sigma_{\alpha_1} / \sum_{i=1}^{N^*+1} l_i)$ of the way in between the current node and next node. In a similar manner, turning operations (cf., Section 3.2.3.6) are parameterized by solving for the number of faces M^* that the agent should move by in the combined inequality

$$\frac{\sum_{i=1}^{M^*} a_i}{A} \leq \sigma_{\alpha_1} < \frac{\sum_{i=1}^{M^*+1} a_i}{A},$$

where a_i is the area of a specified face i and A is the sum of the areas of all the faces the agent could potentially move to. This parameterization increases the probability that the agent will move into larger faces, which one can assume to be sparser and less developed.

The advantages of this parameterization can be observed by returning to the examples in Figure 3.36. Because SPIDRS in-face operations are now a function of the perimeter of the current face and the the distance between nodes, the probability of the agent's operation in Figure 3.36a involving nodes 1, 2, 6, or 4 improves from 44% to 75%. Likewise, the probability of the agent moving to the face $[1, 2, 6, 5, 4]$ in Figure 3.36 increases from 25% to ≈83% because operations now favor moving to faces with larger areas. Thus, while retaining the graph-based foundation of the SPIDRS algorithm, a change in parameterization informed by geometry can bias SPIDRS toward traversing and making topological modifications to the whole graph. However, it should be noted that this change in parameterization does not preclude SPIDRS from moving into areas with a high concentration of nodes or edges, as for some design problems this may actually be a benefit and worth exploring.

Comparisons of the two parameterizations methods are shown in Figure 3.37. Starting with a $1\,\mathrm{m} \times 1\,\mathrm{m}$ square initial graph, the same L-System encoding is interpreted using both graphical and geometric parameterization over a number of recursions. Visual inspection indicates that a geometric parameterization is more successful in achieving an even distribution of nodes and edges throughout the graph than a graphical interpretation. Furthermore, a geometric parameterization appears to reduce (though not eliminate) the formation of slivered faces. Figure 3.38 provides a more quantitative comparison by illustrating the probability density of both parameterizations cre-

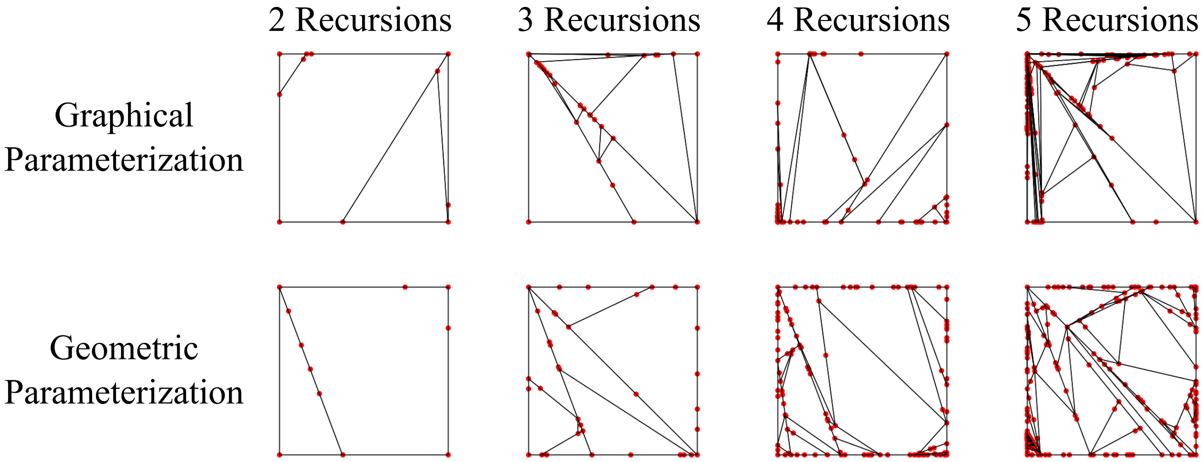
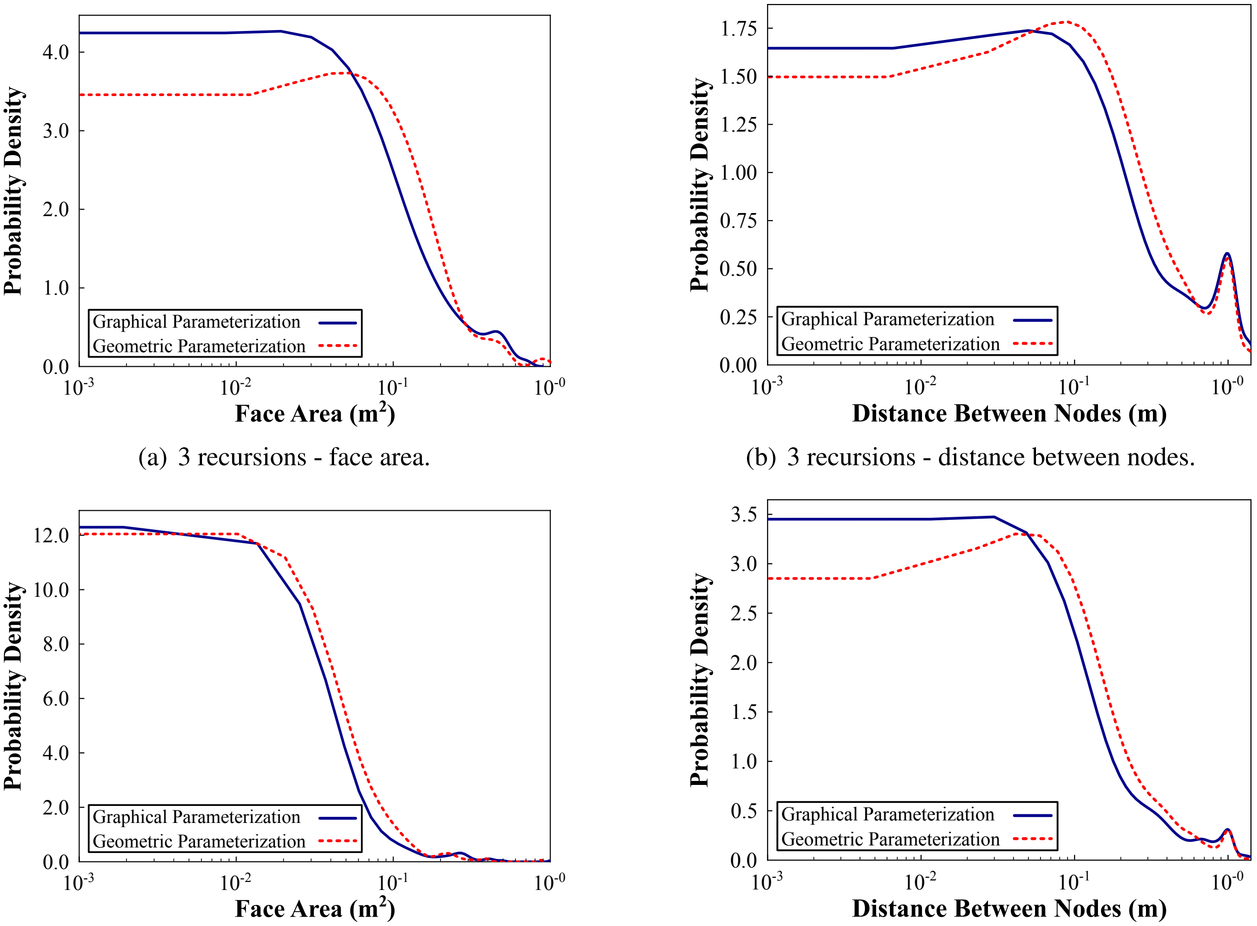


Figure 3.37: Comparison of a single L-System encoding interpreted using both graphical and geometric parameterizations over a number of recursions. The geometric parameterization approach appears to result in a more even distribution of nodes (signified by the red circles) and edges throughout the graph as well as a reduction of slivered faces.

ating faces of a certain area and nodes within a certain distance of each other for both three and four L-System recursions. These plots were made by aggregating the resulting graphs of ten distinct L-System encodings interpreted using the two parameterization methods. For three recursions, a geometric parameterization noticeably reduces the probability of faces having an area of less than $0.01\,\mathrm{m}^2$ (Figure 3.38a) and increases the probability of nodes being more than $0.1\,\mathrm{m}$ away from one another (Figure 3.38b). These results indicate that nodes and edges are being more evenly distributed throughout the graph, and that the probability of slivered faces forming is reduced. For four recursions, the probability of faces having an given area are roughly the same for both parameterization methods, although it appears there is a slightly higher probability of faces having an area of greater than $0.01\,\mathrm{m}^2$ when considering a geometric parameterization (Figure 3.38c). However, the geometric parameterization greatly reduces the probability of nodes being less than $0.01\,\mathrm{m}$ from one another (Figure 3.38d). This suggests that while the size of the faces in the graph may be similar regardless of the parameterization method, a geometric approach is more successful at distributing nodes throughout the graph. Given these results, one can conclude that utilizing



(a) 3 recursions - face area.

(b) 3 recursions - distance between nodes.

(c) 4 recursions - face area.

(d) 4 recursions - distance between nodes.

Figure 3.38: Probability density plots comparing the effect of graphical and geometric parameterization on face area and distance between nodes on 10 SPIDRS-generated graphs.

a geometric parameterization approach enables the SPIDRS algorithm to better traverse the entire graph and should result in a more even distribution of topological modifications.

3.6.3 Genetic Algorithm Parameters

In addition to the workings of the SPIDRS algorithm, the behavior of the genetic algorithm that drives the proposed topology optimization framework is also considered. It is well established that crossover and mutation operations and their associated parameters have a considerable effect on the results generated by genetic algorithms [114, 115, 116]. The structural design problems discussed in Sections 2.2 and 3.4 utilized the default crossover and mutation parameters included in the NSGA-II implementation of DEAP [95]. However, given the highly discontinuous nature of the proposed L-System framework compared to traditional optimization problems, a study is conducted to assess whether modifying various NSGA-II parameters can result in better-performing and more diverse topologies.

Here, the effects of changing four genetic algorithm parameters that inform both crossover and mutation operations are considered. P_{cross} is the crossover probability which controls the probability with which parent solutions are combined to generate offspring solutions. η_{cross} is the crossover distribution index that is inversely proportional to the amount of perturbation in a gene during crossover. High values of η_{cross} dictate that resulting offspring resemble their parents, while low values will result in a much different solution. P_{mut} is the mutation probability which determines the probability that a given gene in a genome will be mutated. η_{mut} is the mutation distribution index and has the same effect as η_{cross} on mutation perturbations. To determine the effect of these parameters on the optimization algorithm, a four-factor, three-level *design of experiments* (DOE) is considered using the parameter values shown in Table 3.4. Rather than utilizing a traditional full-factorial DOE that would result in $3^4 = 81$ distinct analyses, this study employs an L9 orthogonal array proposed by Taguchi [117] shown in Table 3.5, which reduces the number of analyses to 9.

Given the stochastic nature of genetic algorithms, it can be difficult to accurately judge whether certain genetic algorithm parameters can lead to “better” solutions. To attempt to alleviate this issue, the DOE considers a geometric image-matching problem that is detailed in Appendix E. Here, it will suffice to say that by leveraging the Change Face Material operation discussed in Section 3.5.2, SPIDRS can attempt to create topologies capable of matching simple specified shapes. This is an attractive problem for the DOE as it has a single known global optimal solution to compare to, making it easier to compare the performance of each set of parameters. For this study, a multiobjective problem is proposed with the goal of attempting to match two distinct images (shown in Figure 3.39). Additionally, the study considers 50 different optimizations for each set of genetic algorithm parameters, each considering 100 generations starting from the same seeded

Table 3.4: Parameter values for the four-factor, three-level design of experiment (DOE) study.

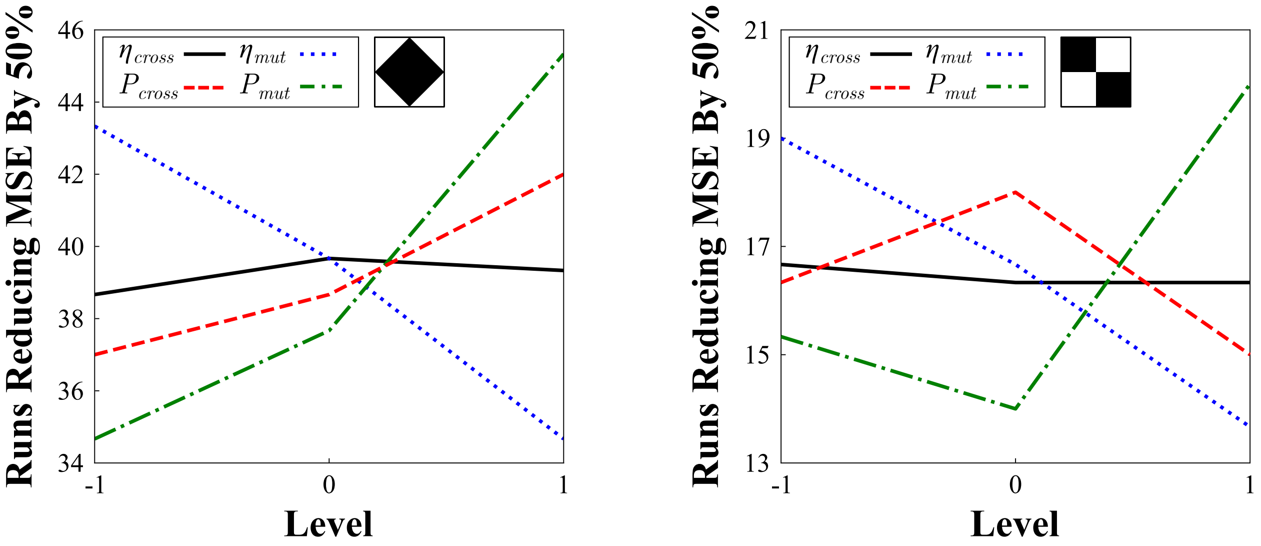
Level	-1	0	1
η_{cross}	20	50	80
P_{cross}	0.5	0.7	0.9
η_{mut}	20	50	80
P_{mut}	0.02	0.04	0.10

Table 3.5: Example of the L9 orthogonal array proposed by Taguchi [117] to reduce the number of analyses required for a four-factor, three-level design of experiments (DOE) to a practical level.

Experiment Number	η_{cross}	P_{cross}	η_{mut}	P_{mut}
1	-1	-1	-1	-1
2	-1	0	0	0
3	-1	1	1	1
4	0	-1	0	1
5	0	0	1	-1
6	0	1	-1	0
7	1	-1	1	0
8	1	0	-1	1
9	1	1	0	-1

population of 100 individuals. The large number of optimizations considered serves to provide a more realistic assessment as to the performance of each set of parameters by eliminating outliers.

The effects of each genetic algorithm parameter on the performance of the topology optimization framework are shown in Figure 3.39 for both images considered. For this study, performance is assessed based on the number of optimizations (out of 50) in which the error of the best performing topology is reduced by more than 50%. For both images, η_{cross} appears to have a nominal effect on the performance of the genetic algorithm. Additionally, there is no discernible trend when considering the effect of P_{cross}. However, there are obvious and interesting trends when considering the parameters that govern mutation. To achieve better performance, Figure 3.39 indicates that one should decrease η_{mut} and increase P_{mut}. In other words, the parameterized L-System is able to encode/generate better topologies when there is a large amount of mutation, both in terms



(a) Runs reducing MSE of a diamond image by 50%.

(b) Runs reducing MSE of a checkerboard image bby 50%.

Figure 3.39: Factor effect plots associated with the DOE study showing the influence of crossover and mutation parameters in a genetic algorithm.

of the number of genes affected and the size of the mutation perturbation. This result is plausible when considering i) that the recursive nature of the L-System dictates that certain genes will have a greater impact on the final topology and thus a larger probability is required to ensure that mutation is actually having an effect, and ii) that because the proposed framework maps discrete L-System characters evenly onto a real number line (cf., Section 2.1.3) larger perturbations are required to mutate one discrete character to another. Based on the results of Figure 3.39, the remaining topology optimization problems presented herein will consider genetic algorithm parameters of $\eta_{cross} = 50$, $P_{cross} = 0.9$, $\eta_{mut} = 20$, and $P_{mut} = 0.10$.

3.6.4 Design Examples

Given the expected improvements made to both the SPIDRS algorithm and optimization framework as a whole, the now familiar cantilevered frame and tensile inverter design problems are revisited. The kinematic rectifier design problem introduced in Section 2.2.3 will also be considered.

3.6.4.1 Cantilevered Frame

The cantilevered frame problem discussed in Section 3.4.1 is reconsidered here to demonstrate the improvements made to the SPIDRS algorithm. The problem is identical to the one illustrated in Figure 3.16 and summarized in Table 3.2 with three exceptions: i) the initial graph is redefined as

$$F = \{[1, 5, 4, 3, 2], [1, 2, 3, 4, 5]\}$$

to account for the consideration of the exterior face (cf., Section 3.6.1), ii) the L-System implementation considers four levels of recursion, and iii) the NSGA-II parameters are modified to match those listed in Section 3.6.3. The Pareto frontier after considering a population size of 100 for 1,000 generations for this problem is shown in Figure 3.40 along with the now familiar frontiers associated with the Michell truss, turtle graphics-interpreted L-System, SIMP implementation, and level set implementation. Clearly, the modifications proposed in Sections 3.6.1-3.6.3 have the desired effect on the performance of the proposed framework, as the generated frontier of solutions shows good agreement with that of the Michell truss across a range of normalized mass values. Figure 3.41 provides a comparison between the topologies of the design closest to the utopia point and stiffest design and the closest comparable Michell truss topology to each, as well as the stress field associated with each structure. In addition to creating topologies that are similar (though not identical) to a Michell truss, SPIDRS is able to generate designs with similar stress distributions which, as discussed in Section 2.2.1.1, can be directly correlated to structural stiffness [103]. Furthermore, it is interesting to note that the frontier generated by the SPIDRS algorithm is able to "fill in" gaps in the frontier created using discrete Michell truss topologies of increasing complexity. These gaps are certainly not a flaw in the Michell truss formulation, but rather a product of specifying a constant structural member cross-section (which will be addressed in Chapter 5). However, this does serve to demonstrate the potential added flexibility in design space exploration that using the SPIDRS algorithm can provide.

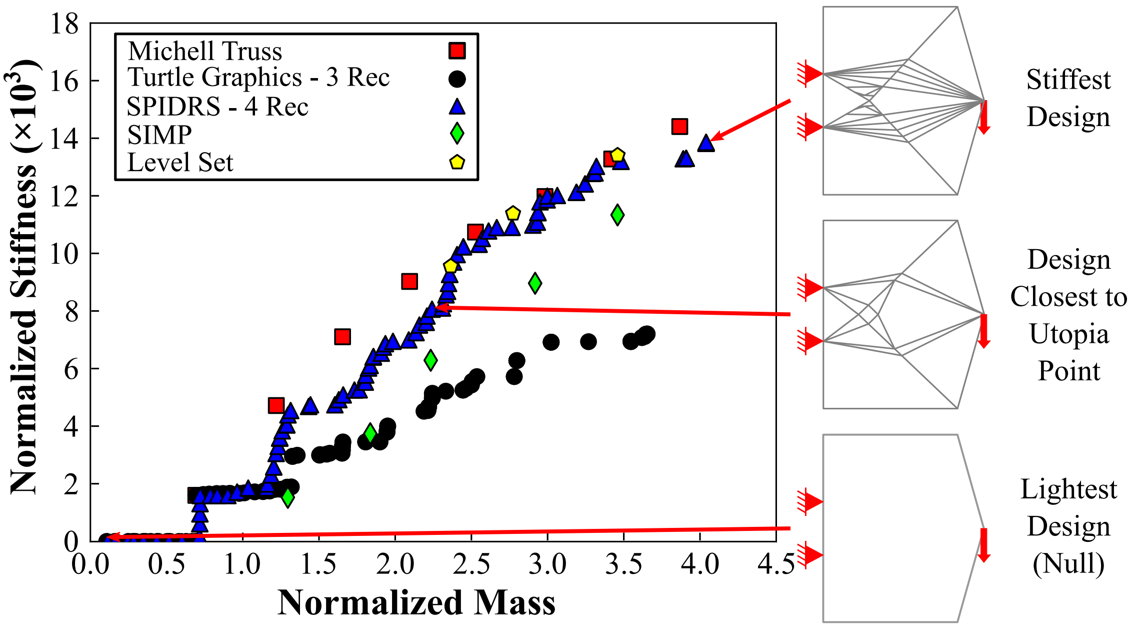
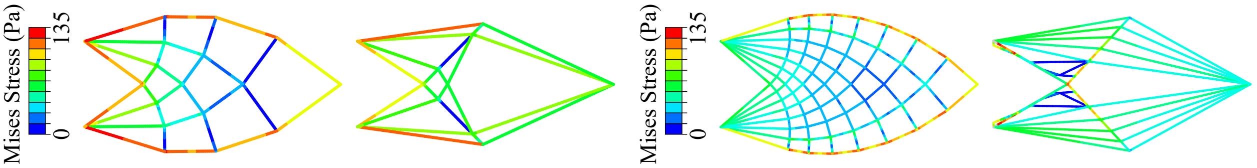


Figure 3.40: Pareto frontier of the cantilevered frame design problem generated using geometrically-parameterized SPIDRS-interpreted L-System encodings, along with comparisons to the frontiers associated with a Michell truss, geometry-based L-System interpretation, SIMP implementation, and level set implementation. Note that the mass and stiffness of the initial graph (cf., Figure 3.16) has been added to these frontiers.



(a) Four-bar Michell truss and SPIDRS-generated design closest to the utopia point.

(b) Eight-bar Michell truss and stiffest SPIDRS-generated design.

Figure 3.41: Comparison between Michell truss SPIDRS-generated cantilevered frame topologies and their associated stress fields. Structural members that made up the initial graph of SPIDRS-generated topologies are removed for clarity.

Figure 3.40 also illustrates how the SPIDRS algorithm compares to both the SIMP and level set topology optimization methods. With the modifications to the proposed framework, SPIDRS now outperforms SIMP-generated solutions across a wide range of normalized masses and is competitive with designs discovered using a level set implementation. This is particularly impressive when recalling that SPIDRS is operating using a fixed cross-sectional area for structural members, while the SIMP and level set implementations allow for members to be thickened to improve structural performance (cf., Figure 2.7).

3.6.4.2 Compliant Mechanism: Tensile Inverter

To further illustrate the improvements made to the SPIDRS algorithm, the tensile inverter problem is revisited. The problem considered is identical to the one illustrated in Figure 3.4.2 and summarized in Table 3.3 with three exceptions: i) the initial graph is redefined as

$$F = \{[1,4,3,2],[1,2,3,4]\} \quad \text{and} \quad M = \{[T,T,V,T],[T,V,T,T]\}$$

to account for the consideration of the exterior face (cf., Section 3.6.1); ii) the spring attached to node 2 is removed, meaning that r_δ represents the free displacement inversion ratio as in Equation 2.1; and iii) the NSGA-II parameters are modified to match those listed in Section 3.6.3. As before, computational analysis considers the full geometric nonlinearity of the solution. The resulting Pareto frontier after considering a population of 60 individuals for 400 generations is shown in Figure 3.42, along with the Pareto frontier generated using the turtle graphics algorithm (cf., Section 2.2.2) and the approximate ideal solution defined by Equation 2.3. Comparing the two L-System topology optimization methods, the frontier associated with the SPIDRS algorithm outperforms that of the turtle graphics algorithm across all values of r_δ and r_f.

It is also interesting to see that solutions generated by the SPIDRS algorithm with higher magnitudes of r_f do surpass the approximate ideal solution derived in Section 2.2.2. To clarify, this solution is based on the assumption that there is a linear relationship between force and displacement, or that

$$W = \int f d\delta = f \int d\delta = f\delta. \tag{3.4}$$

Furthermore, this solution assumes that the the work done by the structure is measurable (i.e., $W \neq$

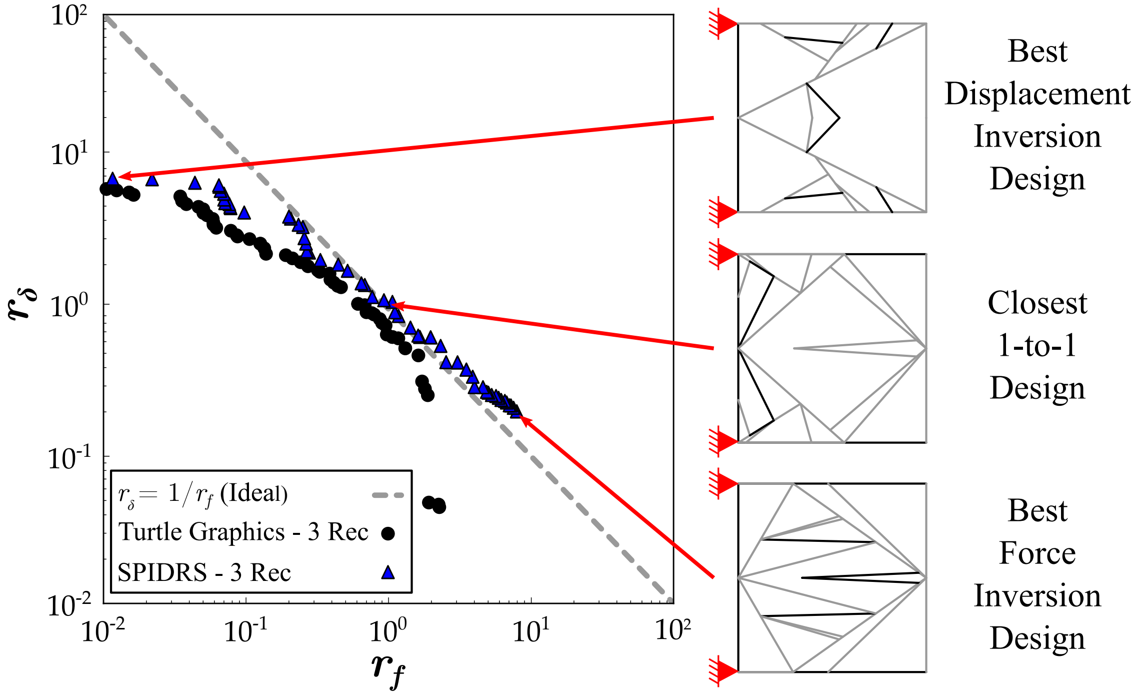


Figure 3.42: Pareto frontier of the tensile inverter design problem generated using geometrically-parameterized SPIDRS-interpreted L-System encodings.

0). However, the free displacement inversion and blocked force inversion ratios in Equation 2.1 are defined for cases where $f_{x,out} = 0$ and $\delta_{x,out} = 0$, respectively, meaning that in these cases $W_{out} = 0$. Additionally, it is intuitive that for a tensile inverter $\delta_{x,out}|_{f_{x,out}=0} > \delta_{x,out}|_{f_{x,out}\neq 0}$ and $f_{x,out}|_{\delta_{x,out}=0} > f_{x.out}|_{\delta_{x,out}\neq 0}$. Thus, it is possible that for this tensile inverter problem, SPIDRS-generated solutions could achieve $r_\delta > 1/r_f$. One possible explanation as to why these designs in particular are able to surpass the approximate ideal solution is that less of the externally applied work is converted into internally stored strain energy.

Also shown in Figure 3.42 are several topologies of interest generated by the SPIDRS algorithm. The best displacement and force inversion designs have ratios of $r_\delta = 7.69$ and $r_f = 7.95$, respectively. Also of note is the design that is closest to having displacement and force inversion ratios equal to one; in this frontier, that design has $r_\delta = 1.12$ and $r_f = 1.07$. This solution features the now familiar load path geometry (shown in Figure 3.43) identical to those observed in Figures 2.12, 3.21, and 3.26, as do the solutions with the best displacement and force inversion

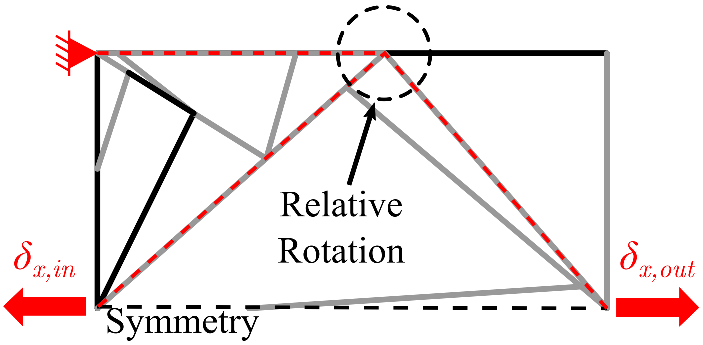


Figure 3.43: Underlying load path geometry for one of many solutions generated by the geometrically-parameterized SPIDRS algorithm. This geometry is consistent with those illustrated in Figures 2.12, 3.21, and 3.26.

behaviors. Furthermore, the load path geometries associated with these solutions follow similar trends to those observed in Figure 3.28, with high displacement inversion solutions featuring rotation points closer to node 3 of the initial graph and high force inversion solutions featuring rotation points closer to node 1. While this load path geometry is a prominent feature in SPIDRS-generated topologies, the algorithm does also create some non-intuitive solutions such as the one illustrated in Figure 3.44. This solution, which has $r_\delta = 3.03$ and $r_f = 0.26$, features a similar mechanism to that of Figure 3.43, but here its rotation point is located relatively close to node 4 of the initial graph and it intersects the x-axis nearly halfway along the length of the inverter rather than at the output node. As the input displacement is converted to a translation at this intersection point, a secondary structural member transmits this translation to the right side of the topology, and by extension the output node. Thus, while it can generate solutions with load path geometries that are well-understood and similar to those of traditional topology optimization methodologies, the SPIDRS algorithm is also able to explore the design space and discover non-intuitive solutions to complex multiobjective design problems.

3.6.4.3 Compliant Mechanism: Kinematic Rectifier

Having demonstrated the ability of a geometrically-parameterized SPIDRS algorithm to match mathematically-proven and idealized optimal solutions, the kinematic rectifier design problem in-

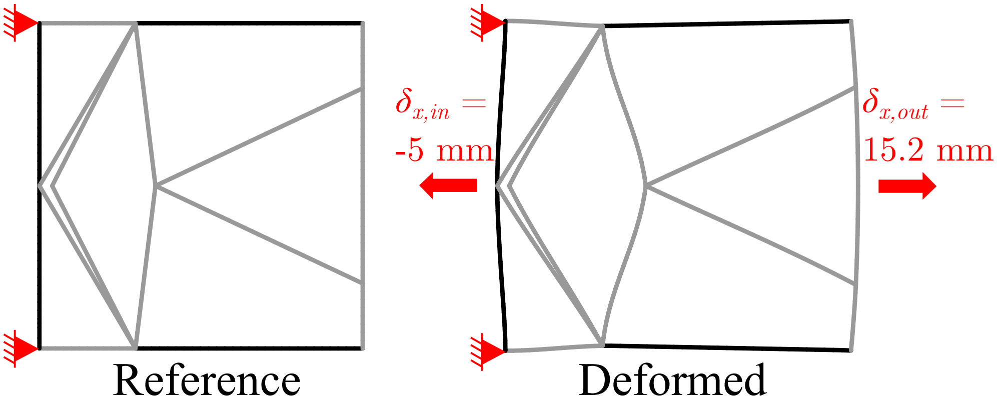


Figure 3.44: Comparison of the reference and deformed configurations of a non-intuitive tensile inverter generated by the geometrically-parameterized SPIDRS framework.

troduced in Section 2.2.3 is reconsidered, where the goal is to maximize an output displacement in a specified direction that is independent of the direction of an input displacement. The boundary conditions and initial graph are identical to those of Figure 3.19 with the exception of the initial graph, which is redefined as

$$F = \{[1, 4, 3, 2], [1, 2, 3, 4]\} \quad \text{and} \quad M = \{[T, T, V, T], [T, V, T, T]\}\,.$$

Details regarding the multiobjective optimization of the kinematic rectifier are shown in Table 3.6. As in Section 2.2.3, two independent structural analyses are performed. In the first analysis a displacement is applied to node 1 ($\delta_{x,in} = 5\,\text{mm}$) while the resulting output displacement $\delta_{x,out}$ is recorded. In the second analysis an input displacement of equal magnitude is applied in the opposite direction ($\delta_{x,in} = -5\,\text{mm}$), and the resulting output displacement is again measured. Both structural analyses account for any potential geometric nonlinear behavior. Given the goal of maximizing $\delta_{x,out}$ to the left (i.e., negative as in Figure 3.19) regardless of the direction of the input, we consider the displacement ratios r_{push} and r_{pull} as defined in Equation 2.4.

The Pareto frontier generated by SPIDRS for the kinematic rectifier is shown in Figure 3.45 along with the frontier generated by the turtle graphics algorithm (cf., Section 2.2.3). For clarity,

Table 3.6: Specifications for the multiobjective topology optimization problem of kinematic rectifier mechanism using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Maximize:	r_{push}, r_{pull}
by varying:	2 axiom characters,
	5 rule assignments (18 genes each),
subject to:	$\sigma_{Mises}^{max} \leq 33.8\,\text{MPa}$
NSGA-II Parameters [68]	
60 members for 400 generations,	
$P_{cross} = 0.9$, $\eta_{cross} = 50$,	
$P_{mut} = 0.1$, $\eta_{mut} = 20$	

designs found in quadrant I are true kinematic rectifiers, designs found in quadrant II are kinematic transmitters, and designs found in quadrant IV are kinematic inverters. Both L-System topology optimization methods result in similar Pareto frontiers in quadrant I, which is interesting when

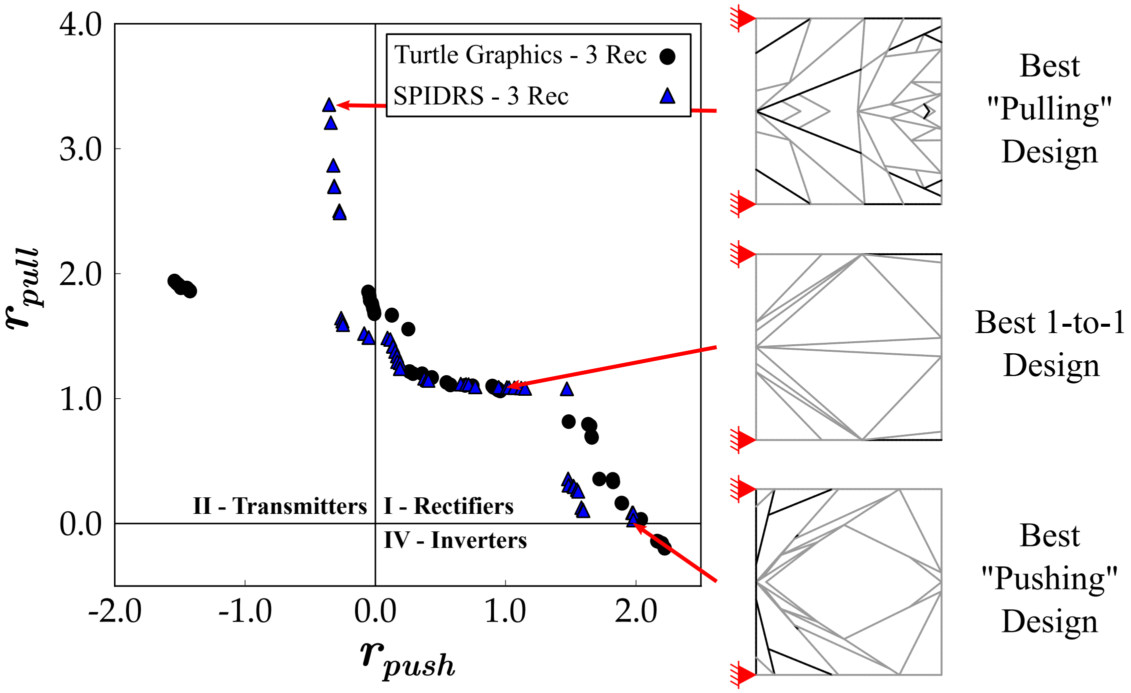


Figure 3.45: Pareto frontier of the kinematic rectifier design problem generated using geometrically-parameterized SPIDRS-interpreted L-System encodings.

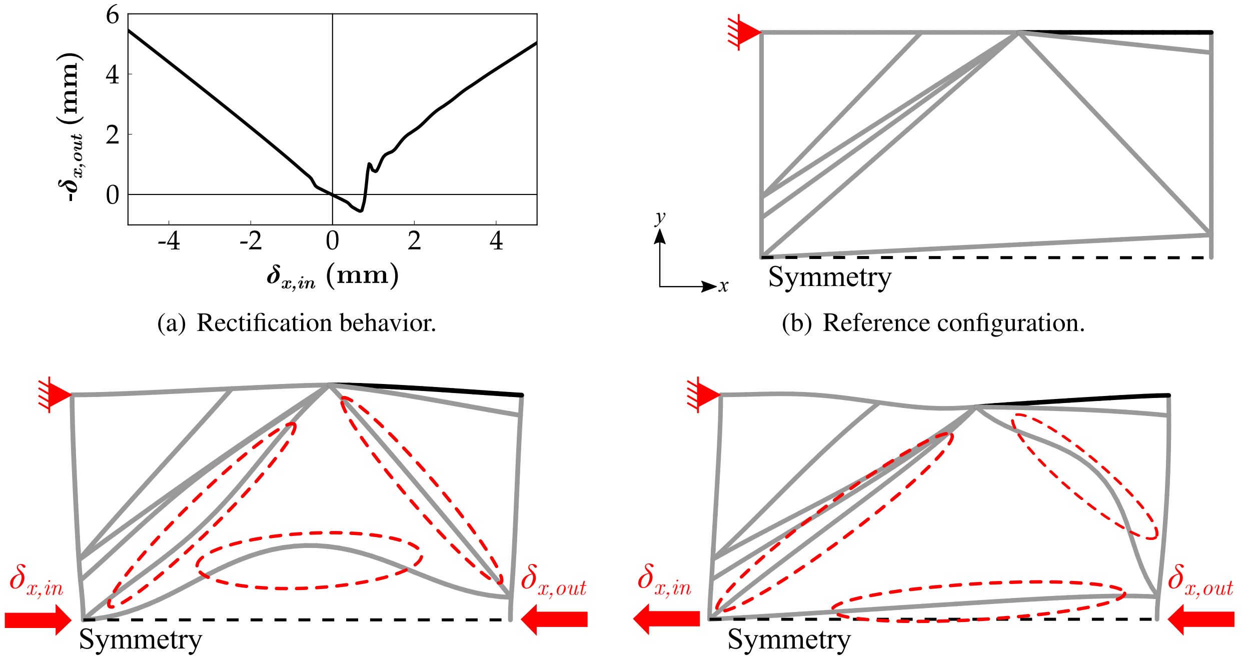
recalling that SPIDRS noticeably outperforms the turtle graphics algorithm when considering a similar problem in the tensile inverter. Additional analyses are needed to determine whether this set of solutions form some sort of local optima and whether, given the complexity of the problem, an increased number of generations are needed for both interpreters to better explore the design space.

To better understand the behavior of a SPIDRS-generated rectifier, consider the design closest to obtaining displacement ratios equal to one ($r_{push} = 1.01$, $r_{pull} = 1.09$) in Figure 3.46. Note that the response of this design is governed by the behavior of a select number of structural members denoted in Figures 3.46c-d. Specifically, during a “pushing” input displacement (Figure 3.46c), a load path similar to those observed in the tensile inverters previously discussed is noticeable. The positive input displacement is translated through the structural member on the left, causing the top of the structure to lift up in the positive y-direction. This causes the structural member on the right to be stiff in tension, which in turn pulls the outpoint point in the desired direction. Notice that during this loading process the bottom structural member, which is in compression, initially pushes the output point in the opposite direction until it eventually buckles at $\delta_{x,in} \approx 1\,\mathrm{mm}$ and allows the output point to move in the desired direction as seen in Figure 3.46a. During a “pulling” input displacement (Figure 3.46d), the structural members on the left and bottom, which are stiff in tension, simply pull the output point in the desired direction while the structural member on the right buckles in compression.

To further explore the ability of the SPIDRS algorithm to develop solutions for non-intuitive problems, a “reverse” kinematic rectifier is considered. This problem has the same boundary conditions, initial graph, and optimization details as discussed previously, but modifies the goal to be maximizing $\delta_{x,out}$ to the right (i.e., positive as in Figure 3.19). Thus, the displacement ratios are redefined as

$$r_{push} = \frac{\delta_{x,out}}{\delta_{x,in}} \quad \text{and} \quad r_{pull} = \frac{-\delta_{x,out}}{\delta_{x,in}}. \tag{3.5}$$

The Pareto frontier obtained for this rectifier design problem after considering 60 individuals for 400 generations is shown in Figure 3.48. Given that the desired output direction has reversed, the



(a) Rectification behavior.

(b) Reference configuration.

(c) Configuration during "pushing".

(d) Configuration during "pulling".

Figure 3.46: a) Illustration of the mechanical motion rectification behavior of the SPIDRS-generated design closest to achieving a 1-to-1 displacement ratio behavior, and comparison of the b) undeformed and c-d) deformed configurations of this mechanism (deformations are unscaled). Notice that the functionalities of specific (circled) branches govern the rectification response of the mechanism.

behavior associated with designs in quadrants II and IV are also reversed such that quadrant II now contains kinematic inverters and quadrant IV now contains kinematic transmitters. The number of solutions along the frontier located in quadrant I, indicating rectification behavior, suggests that reversing the direction of the desired output makes this a much more difficult problem to solve. Considering the potential mechanisms behind each loading cases for both rectifier problems, this proves to be an intuitive result. For the original rectifier, the kinematic inversion step (i.e., "pushing" input displacement) is simply a compressive inverter, which, while not covered in this work, has been briefly explored using the turtle graphics algorithm [85] and has a deformation mechanism that is well-understood. The kinematic transmission step (i.e, "pulling" input displacement) simply requires that structural members remain in tension and pull the output point in the same

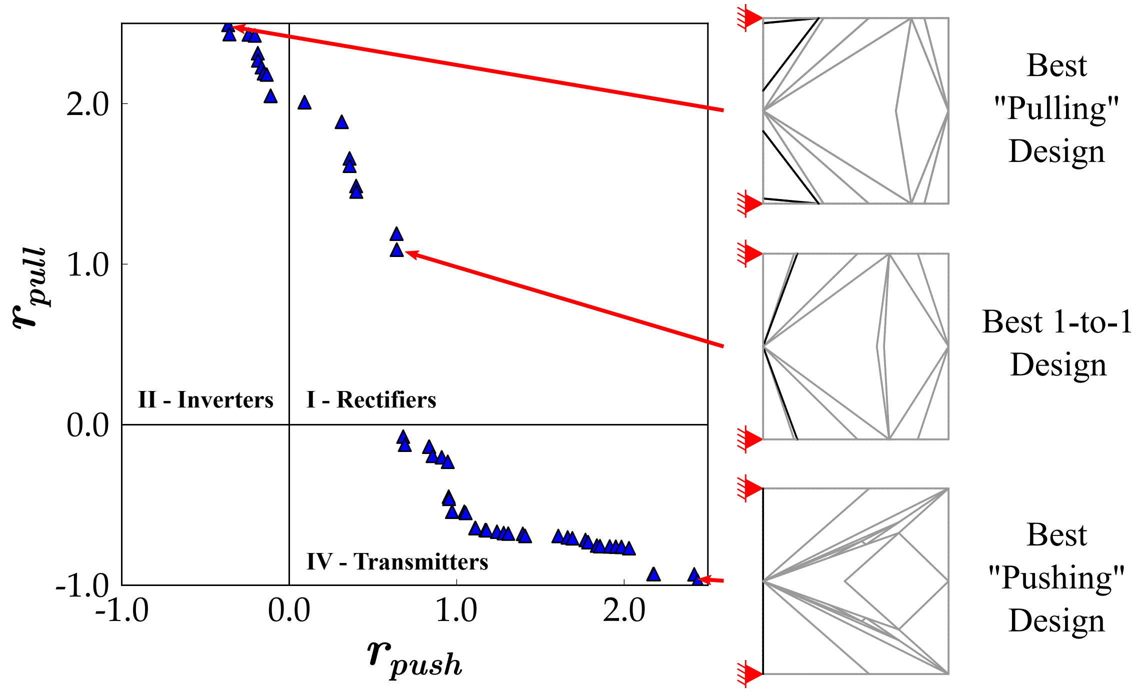
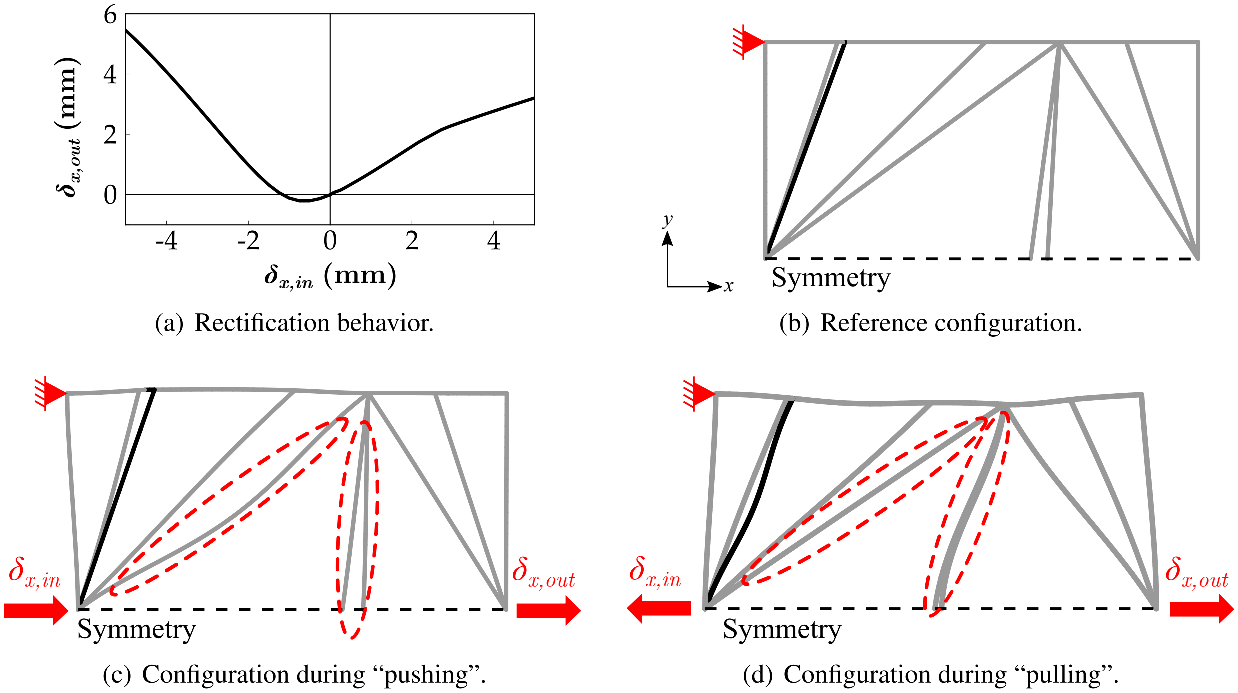


Figure 3.47: Pareto frontier of the “reverse” kinematic rectifier design problem generated using geometrically-parameterized SPIDRS-interpreted L-System encodings.

direction as the input point. For the reverse rectifier, the kinematic inversion step (i.e., “pulling” input displacement) requires the same behavior as a tensile inverter, which has been discussed in great detail in this work. However, the kinematic transmission step (i.e., “pushing” input displacement) needs for structural members to push the output point in the same direction as the input point. In other words, this loading case requires structural members to stay stiff while in compression, which is the opposite of how these members want to behave.

Despite the complexity of the problem, SPIDRS is still able to generate several solutions that demonstrate rectification behavior. Specifically, consider the design closest to obtaining displacement ratios equal to one ($r_{push} = 0.64$, $r_{pull} = 1.09$) in Figure 3.48. As before, the response of the design in governed by only a select number of structural members denoted in Figure 3.48c-d. During a “pushing” input displacement (Figure 3.48c), the structural member on the left translates the input displacement and attempts to cause the top of the structure to lift up in the positive y-direction as in Figure 3.46c. However, the two structural members in the middle, which are oriented almost



(a) Rectification behavior.

(b) Reference configuration.

(c) Configuration during “pushing”.

(d) Configuration during “pulling”.

Figure 3.48: a) Illustration of the “reverse” mechanical motion rectification behavior of the SPIDRS-generated design closest to achieving a 1-to-1 displacement ratio behavior, and comparison of the b) undeformed and c-d) deformed configurations of this mechanism (deformations are unscaled). Notice that the functionalities of specific (circled) branches govern the rectification response of the mechanism.

vertically, resist this movement. Since the displacement of the right side of the structure is constricted in the y-direction by these members, the displacement in the x-direction that is translated by the now buckling structural member on the left dominates and moves the output point in the desired direction. Thus, the SPIDRS algorithm has developed a transmitting structure capable of remaining stiff while some structural members buckle. During a “pushing” input displacement (Figure 3.48d), the structure behave like the tensile inverters discussed throughout this work, with the structural members in the middle initially resisting deformation (indicated by $\delta_{x,out}$ being negative for $\delta_{x,in} \geq\approx -1$ mm) before eventually buckling.

4. THREE-DIMENSIONAL L-SYSTEM TOPOLOGY OPTIMIZATION USING GRAPH-BASED INTERPRETATION

As demonstrated in Chapter 3, the graph-based interpretation of L-System encodings implemented in the SPIDRS algorithm represents a significant improvement over the geometry-based interpretation method previously explored. However, to this point only 2-D structural domains have been considered, while in practice most engineering structural design problems are three-dimensional. Both density-based [118, 119, 120] and level set [121, 122, 123] topology optimization methods are capable of solving and have been extensively applied to 3-D structural design problems. Despite being able to render natural-looking 3-D designs, the turtle graphics interpreter introduced in Chapter 2 is unsuitable for designing 3-D engineering structures. Such a structure typically relies on the intersection of members to create a load path between boundary conditions and applied loads. Given the turtle graphics interpreter generates structure by creating line-segments between two spatial coordinates in 2-D space, it is virtually impossible to guarantee that any two given segments will ever intersect. Having taken this into consideration, the SPIDRS algorithm was initially developed with the goal of eventually becoming a viable option for 3-D topology optimization. Recall that SPIDRS begins with an initial graph (or structure) and generates additional structural members by creating edges between preexisting or newly-created nodes. This subdividing behavior guarantees the intersection of members and increases the probability of the existence of a load path. Thus, provided the half-edge data structure can be adapted to consider 3-D graphs and a graphical operation can be formulated to consider the creation of edges in a 3-D structural design space, the SPIDRS algorithm could feasibly enable the first-ever demonstration of three-dimensional L-System topology optimization.

This chapter describes the adaptation of the coupled parameterized L-System/SPIDRS framework to consider 3-D topology optimization problems and is organized as follows: Section 4.1 discusses how the proposed topology optimization framework is modified to allow for the consideration of 3-D structural design domains, focusing specifically on how the 2-D graphical operations defined in the SPIDRS algorithm can be adapted to consider 3-D graphs and how principles of graph theory enable an elegant solution for creating edges in a 3-D design domain; and Section 4.2 demonstrates the performance of the 3-D L-System topology optimization framework using a series of design problems, including both classic cantilevered frame studies and several compliant mechanism problems.

4.1 Considerations For Extending to 3-D Space

4.1.1 "3-D" Parameterized L-System

Given that it simply encodes design variables and has no notion of the structural design domain, the modifications required to enable the parameterized L-System formulation introduced in Section 3.2.1 to operate in the context of a 3-D space are fairly limited. Here, the alphabet is modified to remove one variable character from the set $\boldsymbol{\alpha}$ and add two constant characters to the set $\boldsymbol{\gamma}$ such that $\boldsymbol{\alpha} = \{\texttt{A}, \texttt{B}, \texttt{C}, \texttt{D}\}$ and $\boldsymbol{\gamma} = \{\texttt{[}, \texttt{]}, \texttt{\{}, \texttt{\}}\}$ (for reasons that will be made clear), while the set $\boldsymbol{\beta} = \{\texttt{-}, \texttt{+}\}$, the axiom, and the set of formal parameters all remain unchanged from Section 3.2.1. Given the modifications to the alphabet, this L-System formulation considers four production rules, each consisting of 10 characters. These are written as

$$\begin{aligned} P_i : \alpha_i(\sigma_{\alpha_1}, \sigma_{\alpha_2}) \rightarrow & \lambda_1^i \lambda_2^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_2 \lambda_3^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_3 \lambda_4^i(\sigma_{\beta_1})_4 \lambda_5^i \\ & \lambda_6^i \lambda_7^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_7 \lambda_8^i(\sigma_{\alpha_1}, \sigma_{\alpha_2})_8 \lambda_9^i(\sigma_{\beta_1})_9 \lambda_{10}^i, \end{aligned}$$

$i = 1, \ldots, 4$ for all $\alpha_i \in \boldsymbol{\alpha}$, where $\lambda_1^i \lambda_2^i \ldots \lambda_{10}^i$ is a string of 10 characters such that $\lambda_j^i \in \Lambda_j$ and

$$\begin{gathered} \Lambda_1 = \{\texttt{[}, \_, \texttt{\{}\}, \ \ \Lambda_2 = \boldsymbol{\alpha}, \ \ \Lambda_3 = \boldsymbol{\alpha}, \ \ \Lambda_4 = \boldsymbol{\beta}, \\ \{\texttt{[}, \_, \texttt{\{}\}, \ \ \Lambda_7 = \boldsymbol{\alpha}, \ \ \Lambda_8 = \boldsymbol{\alpha}, \ \ \Lambda_9 = \boldsymbol{\beta}. \end{gathered}$$

Assignments for λ_5^i and λ_{10}^i are made such that the indicators "[" or "{" in λ_1^i and λ_6^i, if assigned, are closed. It should be noted that, as in Section 3.2.1, each production rule is still defined by only 18 independent design variables. Thus, despite moving from 2-D to 3-D, *there is no increase in the*

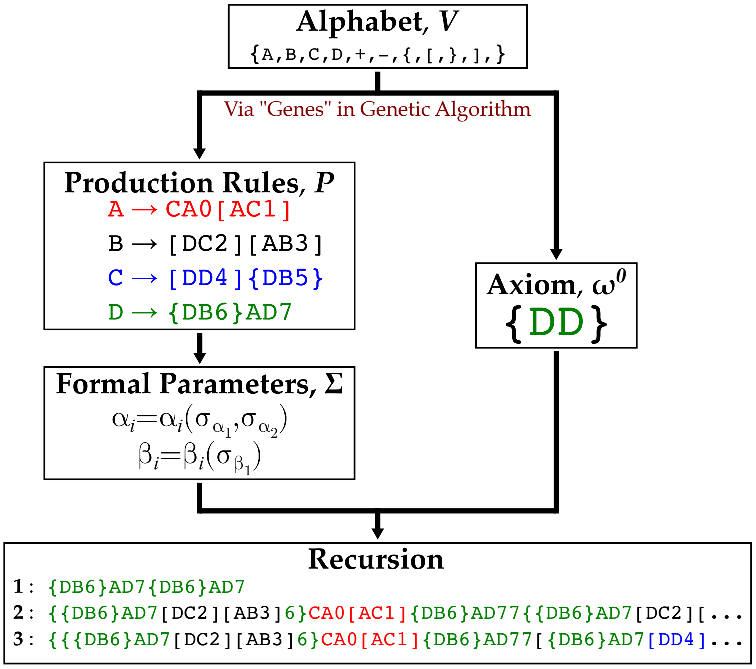

Alphabet, V
{A,B,C,D,+,-,{,[,},],}
Via "Genes" in Genetic Algorithm
Production Rules, P
A → CA0[AC1]
B → [DC2][AB3]
C → [DD4]{DB5}
D → {DB6}AD7
Axiom, ω^0
{DD}
Formal Parameters, Σ
$\alpha_i = \alpha_i(\sigma_{\alpha_1}, \sigma_{\alpha_2})$
$\beta_i = \beta_i(\sigma_{\beta_1})$
Recursion
1: {DB6}AD7{DB6}AD7
2: {{DB6}AD7[DC2][AB3]6}CA0[AC1]{DB6}AD77{{DB6}AD7[DC2][...
3: {{{DB6}AD7[DC2][AB3]6}CA0[AC1]{DB6}AD77[{DB6}AD7[DD4]...

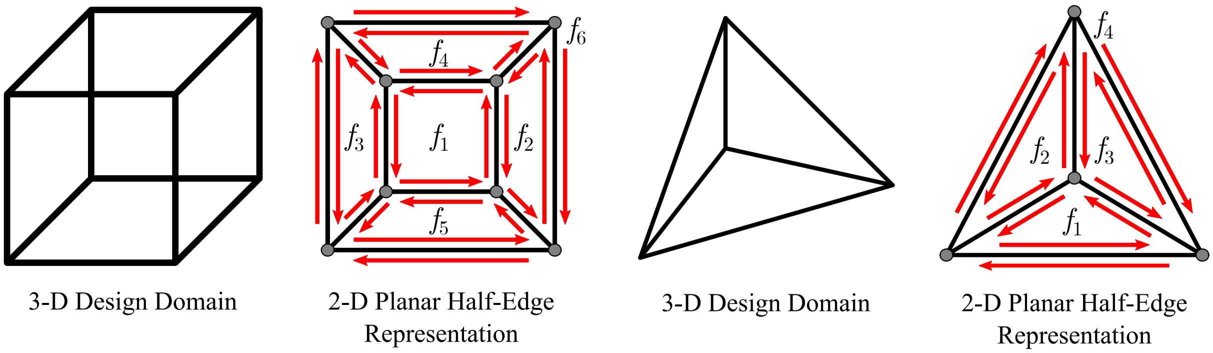
Figure 4.1: Example of recursively generated final strings using a parameterized L-System approach modified for use in a 3-D design domain.

number of design variables required. The recursive generation of final strings using this modified parameterized L-System approach is illustrated in Figure 4.1.

4.1.2 3-D SPIDRS

To enable the consideration of 3-D structural design problems using the SPIDRS algorithm, it initially must be demonstrated that the underlying half-edge data structure defining a graph and previously introduced SPIDRS graphical operations in Section 3.2 remain valid in 3-D space. First, regarding the half-edge data structure, recall that the definition of a graph includes only sets of nodes and edges and a function that relates these two sets, and that these elements are given no spatial significance. Using this, one can say that in the context of this work a given graph, and therefore the half-edge data structure which represents it, is independent of the geometric dimensionality of the structural design domain to which it is applied. Furthermore, the half-edge data structure discussed in Section 3.2.2 is based on the idea that each edge of a graph can be represented using two half-edges (which are assumed to belong to different faces and are oriented in opposite directions) and that edges only intersect one another at points on the graph defined by a node. Thus, provided the 3-D structural design domain can be represented as a simple planar graph and one wishes to create topological modifications only within faces associated with the initial graph, the half-edge data structure as introduced in Section 3.2.2 remains a viable foundation for the SPIDRS algorithm. It will be demonstrated later in this section that creating edges between faces in different geometric planes, which is the goal when considering a 3-D structural design domain, can result in graphs that are not planar (cf., Figure 4.8) but are still able to be represented using the half-edge data structure. Example structural design domains and their associated graphical representations are illustrated in Figure 4.2.

Next, consider the graphical operations implemented by the SPIDRS algorithm as discussed in Section 3.2.3. As with the graph itself, these operations are written as a function of the nodes, edges, and faces of the graph and do not actively consider the spatial positions of these entities, meaning that the graphical operations are independent of the geometric dimensionality of the structural design domain. Additionally, given that a potential 3-D structural design domain should be able to be represented as a simple planar graph (cf., Figure 4.2), it stands to reason that these graphical operations should operate in the same manner as they would on a 2-D structural design domain.



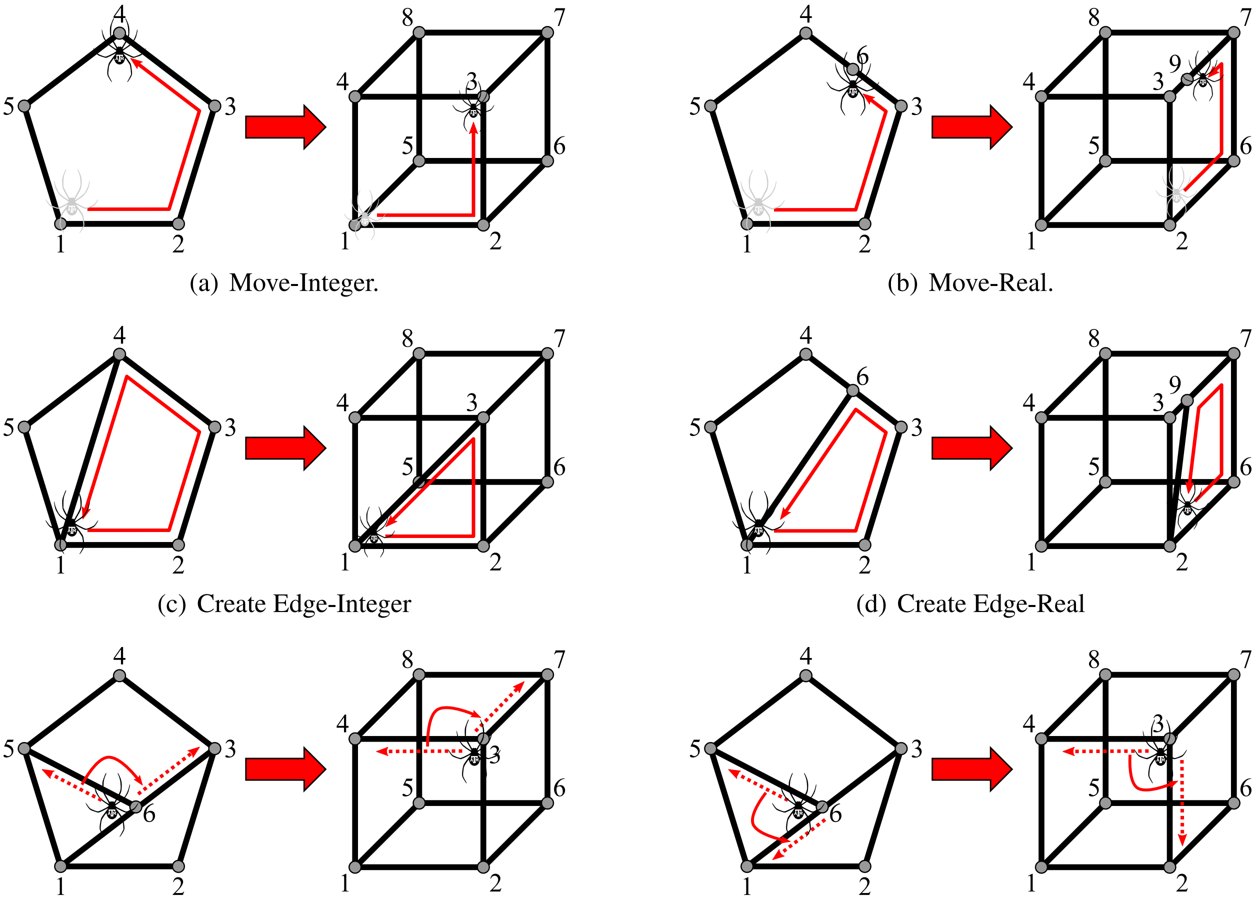
(a) Cube.

(b) Triangular pyramid.

Figure 4.2: Example 3-D structural design domains and their associated 2-D planar graph representation using the half-edge data structure.

Figure 4.3 illustrates the behavior of various SPIDRS graphical operations in both 2-D and 3-D. Note that the in-face operations (cf., Figures 4.3a-d) can traverse and initiate topological modifications on any planar face of the graph as before. Out-of-face turning operations (Figures 4.3e-f) allow the agent to move between faces in different planes that share a common node. Additionally, the bracket operation (Figure 4.6a) operates in a manner consistent with Section 3.2.3.7.

Having shown that the SPIDRS algorithm developed in Section 3.2 remains valid when considering 3-D space, a new graphical operation capable of facilitating the creation of edges between 2-D planar faces is required. Specifically, rather than the subdividing behavior associated with the



(a) Move-Integer.

(b) Move-Real.

(c) Create Edge-Integer

(d) Create Edge-Real

(e) Clockwise turning operation.

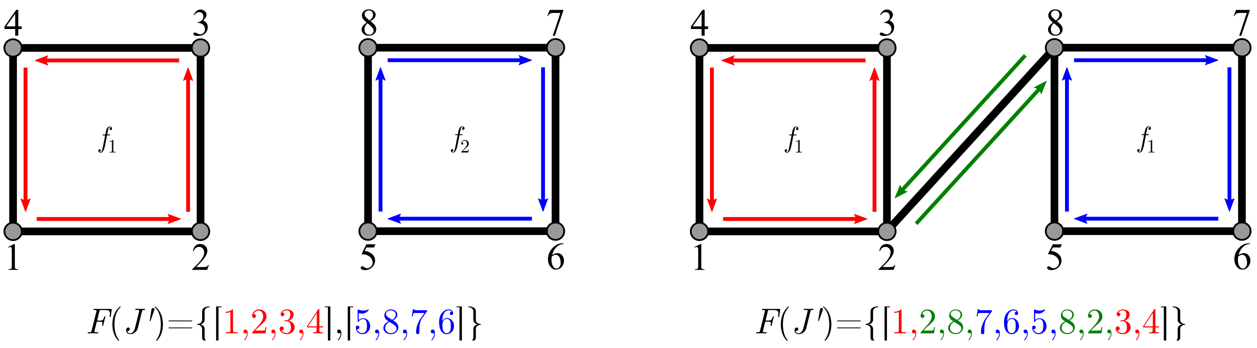
(f) Counterclockwise turning operation.

Figure 4.3: Examples of how the SPIDRS graphical operations introduced in Section 3.2.3 remain valid when considering a 3-D structural design domain.

edge-creation operations previously introduced, this operation allows for the creation of an edge between two nodes that are assumed to make up distinct faces (i.e., there is not a face in the graph that contains both nodes). To elucidate a better understanding of how principles of graph theory can make such an operation possible in the context of SPIDRS' half-edge data structure foundation, consider the directed simple planar graph J' illustrated in Figure 4.4. For demonstration purposes, the half-edges on exterior faces are neglected. Using the notation introduced in Section 3.2.2, this graph is defined by the face set

$$F(J') = \{[1, 2, 3, 4], [5, 8, 7, 6]\} .$$

Notice in Figure 4.4a that J' is a *disconnected* graph, as there is no connectivity defined between f_1 and f_2. Now, consider the creation of an edge between nodes 2 and 8 as illustrated in Figure 4.4b, which must be integrated into the face set $F(J')$ in some manner. Recall that the most general definition of a face in Section 3.2.2 is based on the concept of a directed walk, in which one can move around the graph along adjacent half-edges. Leveraging this concept and considering the orientation of the two-half-edges associated with the newly-created edge, one can redefine the face


4
3
8
7
f_1
f_2
1
2
5
6
$F(J')=\{[1,2,3,4],[5,8,7,6]\}$
4
3
8
7
f_1
f_1
1
2
5
6
$F(J')=\{[1,2,8,7,6,5,8,2,3,4]\}$

(a) Original disconnected graph with two distinct faces.

(b) Graph after the creation of an edge between the two distinct faces.

Figure 4.4: Example of an edge-creation operation between two faces of the disconnected directed planar graph J'. Unlike previous operations, this can be considered a merging of two distinct faces.

set to be

$$F(J') = \{[1, 2, 8, 7, 6, 5, 8, 2, 3, 4]\} .$$

In other words, after traveling along half-edge e_{12} and arriving at node 2, one can simply move along the newly-created half-edge e_{28} to node 8. From here, one can move about the portion of the graph previously defined as f_2 until arriving back at node 8. In a similar manner, moving along the half-edge e_{82} results in a return to node 2, after which one can continue to move about the portion of the graph previously defined as f_1. Note that the modified face set $F(J')$ remains consistent with the definition of a face given in Section 3.2.2. Thus, while the edge-creation operations defined in Sections 3.2.3.3-3.2.3.4 are based on the idea of *subdividing* a given face, this proposed method can be thought of as *merging* two distinct faces together.

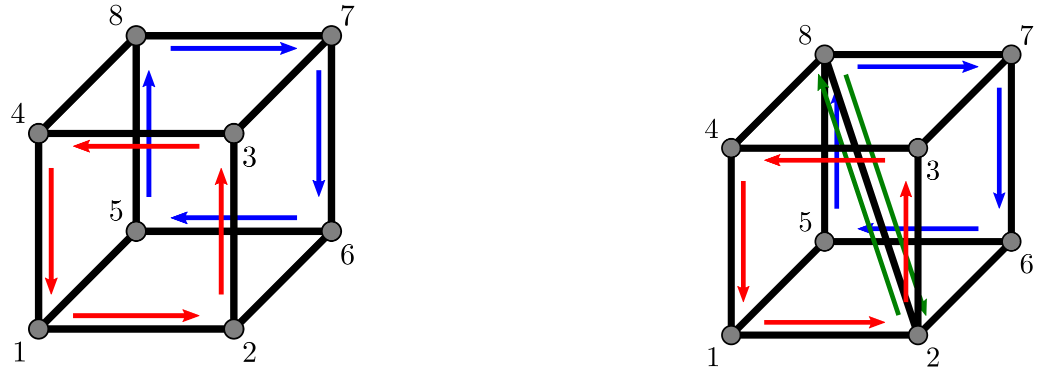
In the same manner, this operation can be applied to 3-D structural design domains as illustrated in Figure 4.5. Consider the directed simple planar graph K' which is defined by

$$F(K') = \{[1, 2, 3, 4], [2, 6, 7, 3], [5, 8, 7, 6], [1, 4, 8, 5], [2, 1, 5, 6], [3, 7, 8, 9]\} .$$

As before, assume that an edge is created between nodes 2 and 8 (Figure 4.5b), which in this context are associated with faces $[1, 2, 3, 4]$ and $[5, 8, 7, 6]$, respectively. Note that while K' is not a disconnected graph, one can say that these two faces are disconnected as they do not share a common node. Once again, leveraging the definition of a face allows for the face set describing K' to be redefined as

$$F(K') = \{[1, 2, 8, 7, 6, 5, 8, 2, 3, 4], [2, 6, 7, 3], [1, 4, 8, 5], [2, 1, 5, 6], [3, 7, 8, 9]\} ,$$

thus merging the faces $[1, 2, 3, 4]$ and $[5, 8, 7, 6]$ via the newly-created edge. More significantly, this operation allows for the creation of edges between faces defined to be in different planes of the structural design space.



$F(K')=\{[1,2,3,4],[2,6,7,3],[5,8,7,6],$
$[1,4,8,5],[2,1,5,6],[3,7,8,4]\}$

(a) Original graph.

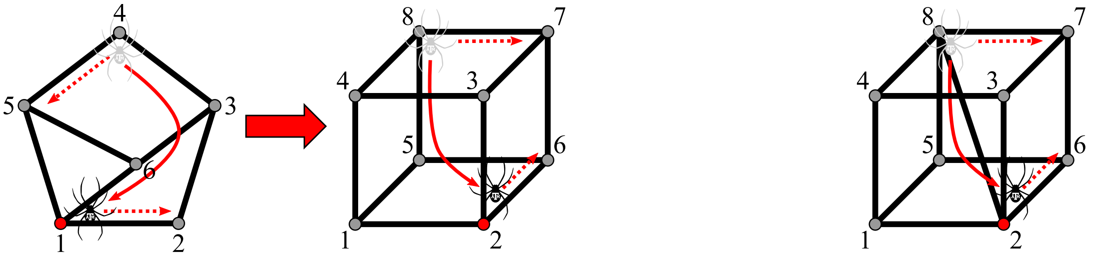
$F(K')=\{[1,2,8,7,6,5,8,2,3,4],[2,6,7,3],$
$[1,4,8,5],[2,1,5,6],[3,7,8,4]\}$

(b) Modified graph.

Figure 4.5: Example of an edge-creation operation between two disconnected faces of the directed planar graph K' associated with a 3-D structural design domain. This operation allows for edges to be created between faces defined to be in different planes of the design space. Note that only the half-edges associated with the faces of interest are denoted.

To implement this edge-creating operation in the SPIDRS algorithm, inspiration is taken from the Bracket operation first introduced in Section 3.2.3.7 and illustrated in Figure 4.6a. Recall that when SPIDRS comes across an open bracket in the L-System encoding, it saves the current location of the agent and executes any commands that follow; a closed bracket returns the agent to the most recently saved location without constructing any additional edges on the graph. In a similar manner, the new Create Interfacial Edge operation illustrated in Figure 4.6b is represented by the open and closed braces added to the L-System alphabet in Section 4.1.1; however, unlike the Bracket operation, as the agent returns to the most recently saved location, it constructs an edge between the two nodes and modifies the topological information of the graph accordingly. It should be noted that SPIDRS checks to ensure that the current node and most recently saved location do not share a face; if this is the case, the Create Interfacial Edge operation is skipped.

The addition of the Create Interfacial Edge operation to the SPIDRS algorithm has several consequences. First, while not explicitly stated, it was assumed in Chapter 3 that an edge marked a division between faces such that twin half-edges corresponded to two distinct faces. As seen in



(a) Bracket operation, [. . .] .

(b) Create Interfacial Edge, { . . . } .

Figure 4.6: Illustration of how the Bracket operation depicted in a) serves as the inspiration the Create Interfacial Edge operation in b).

Figures 4.4-4.5, this is no longer guaranteed; in fact, the idea of twin half-edges corresponding to the same face is a key idea that makes this operation possible. While this does result in slight complications to the SPIDRS algorithm (e.g., when considering a saved location for the Bracket or Create Interfacial Edge operations, the node number, associated face, and next node in the face are now recorded to further define the half-edge the agent should return to), the half-edge data structure remains valid. Second, the addition of this operation leaves open the possibility for non-planar faces (Figure 4.7), severely complicating the area calculations required for the geometric parameterization of turning operations (cf., Section 3.6.2). To maintain computational efficiency while retaining the perceived increase in performance granted by the geometric parameterization of SPIDRS graphical operations, the parameterization of 3-D turning operations is modified to consider the perimeter of potential faces rather than area. This method of parameterization has been demonstrated to compare favorably with area-based parameterizations when comparing 500 different 2-D SPIDRS-generated topologies (cf., Appendix F). Third, one should note that this specific formulation for creating 3-D faces throws into question the use of the Change Face Material operation that increases the dimensionality of generated structure. Ideally, in a 3-D design domain this operation would result in the formation of continuum volumes rather than 2-D planar or parameterized shells, but more research is required to determine how this would be implemented. Fourth, notice that the Create Interfacial Edge operation is represented by characters in the param-

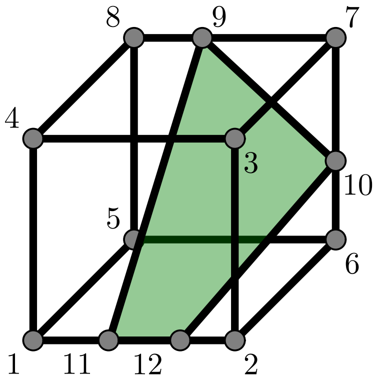


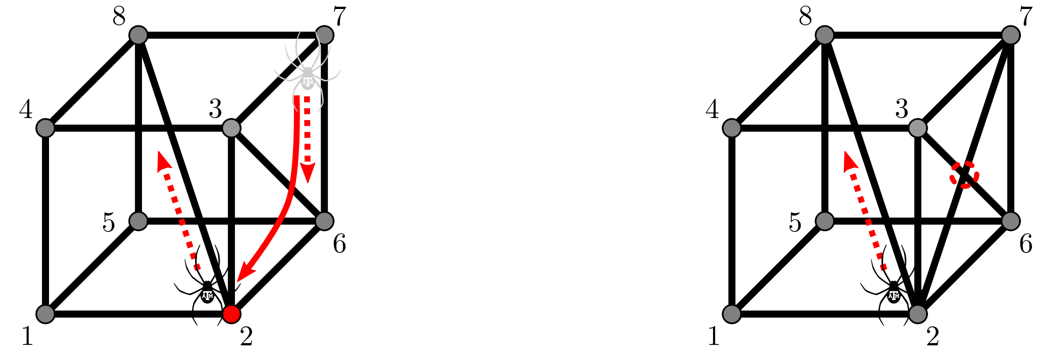
Figure 4.7: Example of a non-planar face within a 3-D structural design domain. The possibility of non-planar faces precludes the use of area as part of the geometric parameterization of SPIDRS turning operations.

eterized L-System set γ (cf., Section 4.1.1) which do not have formal parameters assigned to them. Recall that these formal parameters help define the material assignments of edges created using the Create Edge-Integer and -Real operations. Due to this lack of formal parameters, problems herein that consider a 3-D structural design domain will be limited to a single material, precluding the need for the Change Material operation. However, it should be noted that, in general, multiple materials could be considered in this implementation of the SPIDRS algorithm for 3-D design domains with an *a priori* determination of the material assignments to any edges created using the Create Interfacial Edge operation.

Perhaps the most impactful consequence of extending SPIDRS into 3-D space is that, in general, the graph modified by the algorithm can no longer be guaranteed to be planar. This is caused by the facts that faces are no longer guaranteed to be convex and that the Create Interfacial Edge still holds when the two faces being considered lie in the same plane. For example, assume that the graph K' illustrated in Figure 4.6b has been further modified such that its face set is defined by

$$F(K') = \{[1,2,8,7,6,5,8,2,3,4],[2,6,3],[3,6,7],[1,4,8,5],[2,1,5,6],[3,7,8,9]\}$$

as shown in Figure 4.8a. Now, consider that the agent, which is currently positioned at node 7



$F(K')$={[1,2,8,7,6,5,8,2,3,4],[2,6,3],
[3,6,7],[1,4,8,5],[2,1,5,6],
[3,7,8,4]}

(a) Create Interfacial Edge operation.

$F(K')$={[1,2,7,6,3,7,2,8,7,6,5,8,2,3,4],
[2,6,3],[1,4,8,5],[2,1,5,6],
[3,7,8,4]}

(b) Resulting non-planar graph K'.

Figure 4.8: Example of how the introduction of the Create Interfacial Edge operation can result in the formation of a non-planar graph due to edge intersections at locations not defined by a node, as denoted by the red circle in b).

along the half-edge e_{76}, receives the command $\}$ and returns to node 2 and orients itself along the half-edge e_{28}, creating and edge between nodes 7 and 2 as it moves. This operation is valid, as nodes 2 and 7 no longer share a face; however, given that SPIDRS is concerned only with the graph and not with the spatial position of elements of the graph, the algorithm does not know that the newly-created edge between nodes 2 and 7 intersects the existing edge between nodes 3 and 6 at a point not defined by a node (denoted by a circle in Figure 4.8b). Therefore, the graph K' is no longer planar. While this could be remedied by simply calculating potential intersection points between edges and either skipping the edge-creation process or adding a node at that point, such a process requires the consideration of the spatial positions or geometry of the graph, which is counter to the objective of the SPIDRS algorithm. Thus, herein these intersections are allowed with the understanding that they must be considered during structural analysis.

Example 3-D structural topologies generated by SPIDRS using the parameterized L-System example from Figure 4.1 are illustrated in Figure 4.9, which represents the first-ever demonstration of truly three-dimensional L-System topology optimization. As with the 2-D structural topologies in Figure 3.14, increasing the number of L-System recursions results in increased topological com-

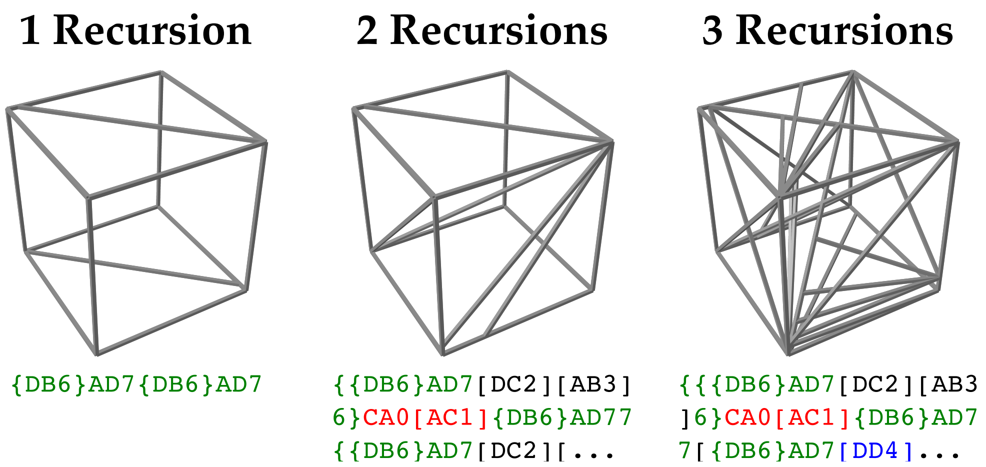


Figure 4.9: Example of 3-D structural topology generation using a SPIDRS interpretation of the parameterized L-System example in Figure 4.1.

plexity. It is also important to stress that, as seen in the single recursion case, the introduction of the Create Interfacial Edge operation does not guarantee the formation of edges between disconnected faces in the graph. It should also be noted that, in general, the creation of one interfacial edge precipitates more due to the in-face edge-creation commands operating on a face that exists on multiple planes, as seen in the three recursion example.

4.2 3-D Design Examples

Having demonstrated that the SPIDRS algorithm presented in Chapter 3 can be extended to consider 3-D structural design domains, a series of design problems are considered. These problems utilize the same genetic optimization framework discussed in Section 3.3 and illustrated in Figure 3.15 with the exception of material overlap constraint outlined in Appendix D, which is neglected. As stated in Section 4.1.2, the 3-D problems presented herein consider only a single material (the previously utilized VeroWhite, the experimental properties of which can be found in Table 2.1); thus, the optimization framework requires only 74 independent design variables. For each design problem, the structural design approach is summarized, the optimization problem is quantified, and results are presented in the form of Pareto frontiers and associated example topological configurations.

4.2.1 Cantilevered Frames

The first set of 3-D topology optimization problems consider a cantilevered frame subjected to both transverse and torsional loading conditions. The initial state of each graph is defined by $F = \{[1,2,3,4],[6,5,8,7],[1,4,8,5],[3,2,6,7],[1,5,6,2],[4,3,7,8]\}$. In both cases, a fully-fixed boundary condition (i.e., $u_x = u_y = u_z = 0$) is placed on all nodes in the plane associated with the face $[1,4,8,5]$. A description of both multiobjective optimization problems is given in Table 4.1. As in Sections 2.2.1, 3.4.1, and 3.6.4.1, the goal of these design problems is to minimize the mass of the structure (which tends toward sparse topologies) while maximizing the stiffness (which favors dense topologies). Created structural members are assigned a constant cross-section of $1\,\text{mm} \times 1\,\text{mm}$, and structural analysis considers a fully linear FEA model with an average element length of 30 mm. Given the assumed linearity of the problem as well as the novel parallelization technique introduced in Section 2.1.3, each problem is able to consider $100 \times 1,000 = 100,000$ individual designs with relative computational efficiency.

4.2.1.1 Transverse Loading

An illustration of the cantilevered truss design problem considering a transverse loading condition is shown in Figure 4.10. The 3-D structural design domain is defined to have a length of 120 mm, a height of 40 mm, and a width of 30 mm. To provide an indication of the transverse

Table 4.1: Specifications for the multiobjective 3-D topology optimization problem of a light and stiff structural frame subjected to transverse/torsional loading using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Minimize (Maximize):	normalized mass (normalized stiffness)
by varying:	2 axiom characters, 4 rule assignments (18 genes each),
subject to:	no constraints
NSGA-II Parameters [68]	
100 members for 1,000 generations, P_{cross} = 0.9, η_{cross} = 20, P_{mut} = 1/52, η_{mut} = 20	

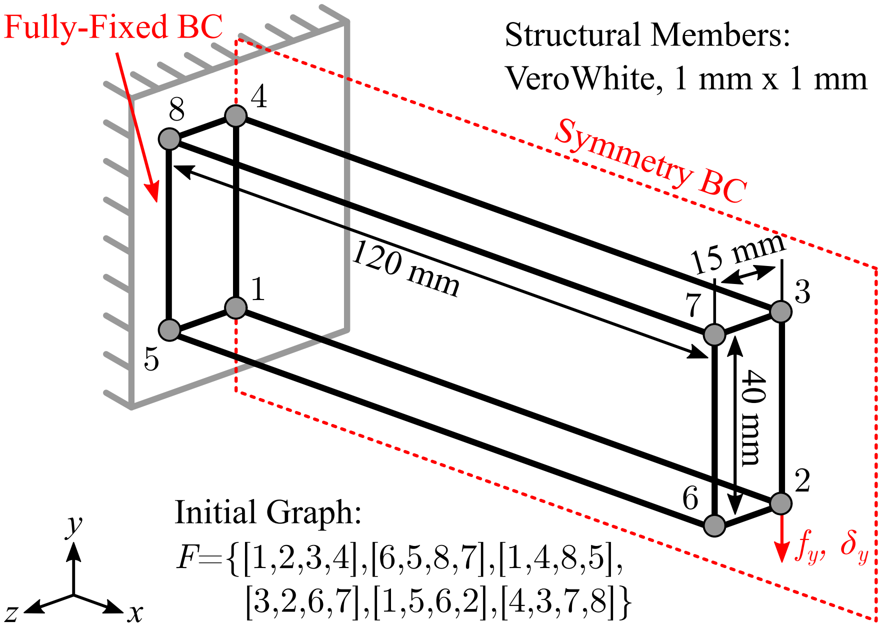


Figure 4.10: Initial graph and boundary conditions associated with the 3-D cantilevered frame subjected to transverse loading.

stiffness of a given structural topology, a concentrated force of $f_y = -0.05\,\mathrm{N}$ is applied to node 2 and the resulting displacement δ_y is measured. Additionally, geometric symmetry is applied about the x-y plane associated with the face $[1, 2, 3, 4]$ such that only half of the topology is generated.

The multiobjective 3-D topology optimization problem described in Table 4.1 was considered for both two and three levels of parameterized L-System recursion (cf., Section 4.1.1), the resulting Pareto frontiers of which are shown in Figure 4.11. Several designs of interest corresponding to these frontiers are illustrated in Figures 4.12-4.13. Notice that these topologies feature elegant truss-like structural layouts and deliberate load paths between boundary conditions. Also note that, as observed in Figure 4.9, increasing the number of recursions leads to increasingly complex structural topologies such that it becomes difficult to properly observe design trends (cf., Figure 4.13c). Due to this result, subsequent 3-D design problems will consider only two levels of recursion.

Also shown in Figure 4.11 are frontiers obtained using a SIMP formulation from the Tosca optimization suite available in Abaqus [124]. This implementation considers a design problem identical to that shown in Figure 4.10 and material properties consistent with those given in Ta-

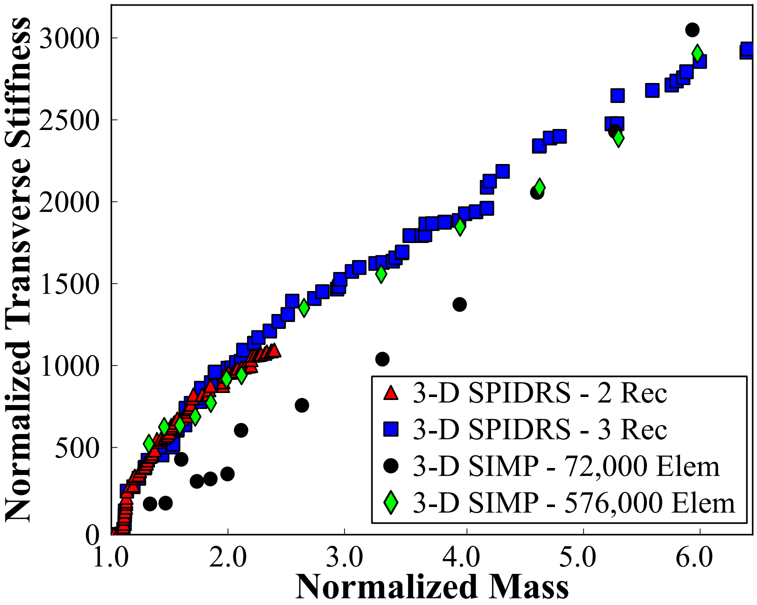
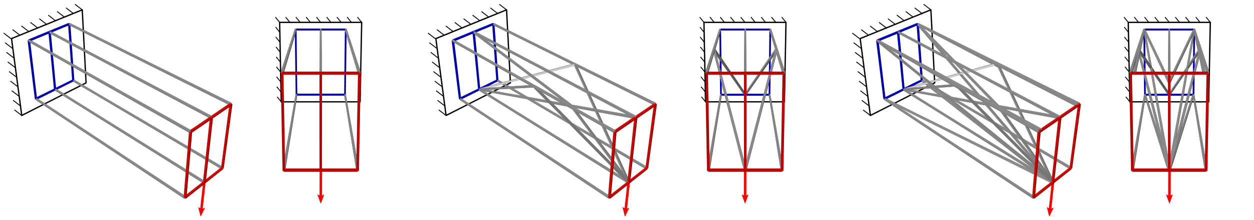


Figure 4.11: Pareto frontiers associated with a 3-D cantilevered frame subjected to transverse loading (cf., Figure 4.10) at two different levels of recursion, along with frontiers generated using a SIMP formulation from the Tosca optimization framework [124] implemented in Abaqus using two different design domain discretizations.

ble 2.1. Two different discretizations of the initial structural domain are considered: i) a mesh made up of elements measuring $1\,\text{mm} \times 1\,\text{mm} \times 1\,\text{mm}$, corresponding to 72,000 design variables, and ii) a mesh made up of elements measuring $0.5\,\text{mm} \times 0.5\,\text{mm} \times 0.5\,\text{mm}$, corresponding to 576,000 design variables. Select resulting structural topologies for both discretizations considered are shown in Figures 4.14-4.15.

Comparing the resulting topologies and associated frontiers to those generated using the SPIDRS algorithm, several interesting trends emerge. First, the SPIDRS-generated structural topologies illustrated in Figures 4.12-4.13 and SIMP generated topologies illustrated in Figures 4.14-4.15 share many topological and geometric similarities, which suggests that SPIDRS is effective in exploring a 3-D structural design space. Second, as seen in the 2-D cantilevered frame design studies, it is noticeable that the SIMP-generated frontier associated with the coarser discretization made up of 72,000 elements is outperformed by SPIDRS-generated topologies (regardless of the number of L-System recursions considered) at lower values of normalized mass. This is believed to be caused by

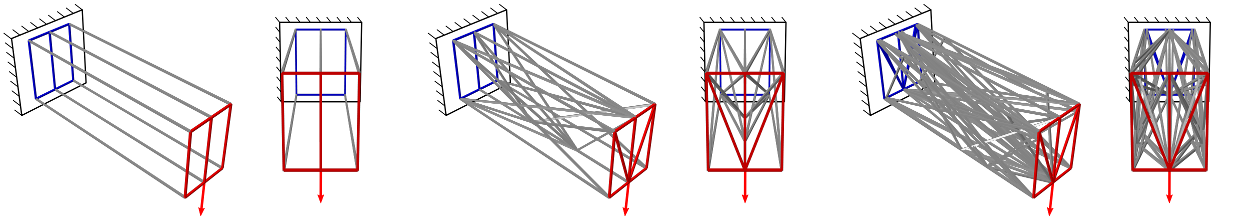


(a) Lightest design (normalized mass of 1, normalized stiffness of 1).

(b) Design closest to the utopia point (normalized mass of 1.514, normalized stiffness of 667.9).

(c) Stiffest design (normalized mass of 2.397, normalized stiffness of 1127.2).

Figure 4.12: Several 3-D structural topologies of interest generated by SPIDRS when considering a cantilevered frame subjected to transverse loading and two levels of parameterized L-System recursion. Structural members at each end are colored to better illustrate the topology.



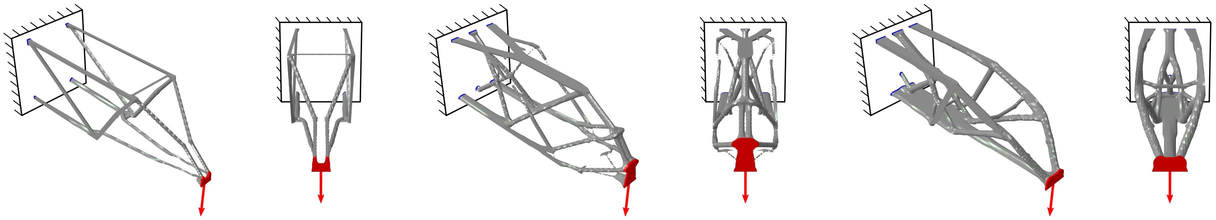
(a) Lightest design (normalized mass of 1, normalized stiffness of 1).

(b) Design closest to the utopia point (normalized mass of 2.551, normalized stiffness of 1425.4).

(c) Stiffest design (normalized mass of 6.456, normalized stiffness of 2963.6).

Figure 4.13: Several 3-D structural topologies of interest generated by SPIDRS when considering a cantilevered frame subjected to transverse loading and three levels of parameterized L-System recursion. Structural members at each end are colored to better illustrate the topology.

the fact that SIMP has difficulty in creating meaningful load paths with a relatively small amount of material when considering such a coarse discretization. However, at higher normalized mass values, the SIMP-generated structures using this discretization perform better relative to those created by SPIDRS because they both have more material with which to create load paths and are not limited to a constant cross-sectional area. Finally, it is interesting to note that there is good agreement between the frontiers associated with the SIMP implementation considering 576,000 design elements and SPIDRS algorithm. In particular, the performance of these SIMP-generated structural topologies at lower normalized mass values highlights the effect that design domain dis-

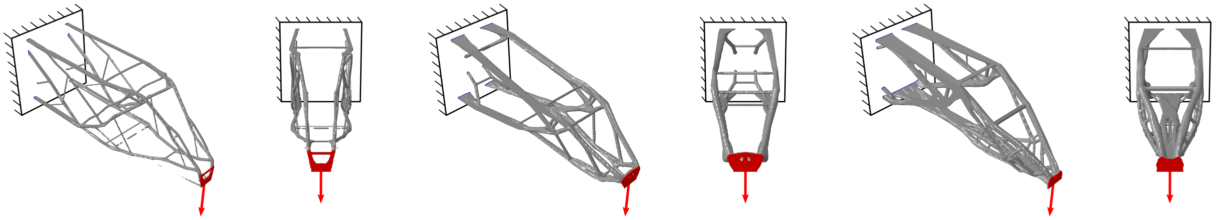


(a) Design with a maximum volume fraction of 0.012 (normalized mass of 1.60, normalized stiffness of 459.68).

(b) Design with a maximum volume fraction of 0.03 (normalized mass of 3.99, normalized stiffness of 1404.87).

(c) Design with a maximum volume fraction of 0.045 (normalized mass of 5.98, normalized stiffness of 3080.87).

Figure 4.14: Several 3-D structural topologies of interest generated using a SIMP implementation with 72,000 elements considering a cantilevered frame subjected to transverse loading. Structural members at each end are colored to better illustrate the topology.



(a) Design with a maximum volume fraction of 0.012 (normalized mass of 1.59, normalized stiffness of 667.39).

(b) Design with a maximum volume fraction of 0.03 (normalized mass of 3.99, normalized stiffness of 1879.64).

(c) Design with a maximum volume fraction of 0.045 (normalized mass of 6.03, normalized stiffness of 2935.73).

Figure 4.15: Several 3-D structural topologies of interest generated using a SIMP implementation with 576,000 elements considering a cantilevered frame subjected to transverse loading. Structural members at each end are colored to better illustrate the topology.

cretization has on potential solutions obtained when using density-based topology optimization methodologies.

However, to truly compare the two 3-D topology optimization methods a comparison of their respective computation times is also required and is illustrated in Table 4.2[1]. Using the SIMP

[1] It is recognized that the number of functional evaluations required for each method is perhaps a stronger metric for this comparison. However, such a comparison is complicated by the unique parallelization technique utilized by the proposed L-System topology optimization framework (cf., Section 3.2.3) as well as the inability to obtain the number of evaluations required for gradient calculations within the SIMP implementation. Therefore, throughout the remainder of this work, computation time will be used.

Table 4.2: Comparison of computation times necessary to generate a single design and a Pareto frontier consisting of 100 designs. 3-D SPIDRS is capable of generating a frontier comparable with a 576,000 element SIMP discretization in approximately 2.5% of the computation time.

	Computation Time (hr)	
	Single Design	Pareto Frontier (100 Designs)
3-D L-System - 2 Rec	–	22.08
3-D L-System - 3 Rec	–	27.20
3-D SIMP - 72,000 Elem	1.38	138
3-D SIMP - 576,000 Elem	10.75	1,075

formulation implemented in Tosca, the generation of a single 3-D structural topology requires an average computation time of 1.38 hours for a 72,000 element design domain discretization and 10.75 hours for a 576,000 element design domain discretization based on the SIMP-generated solutions shown in Figure 4.11. Recall that one of the specified benefits of the proposed L-System topology optimization framework is its ability to easily generate a frontier of potential solutions, which is a key aspect of the preliminary design process where the goal is simply to explore the structural design space and identify potential trends that merit further investigation. The SPIDRS-generated Pareto frontiers illustrated in Figure 4.11 consist of 100 3-D structural topologies analyzed over 1,000 generations and required 22.08 hours for 2 parameterized L-System recursions and 27.20 hours for 3 recursions. Generating a similar frontier with 100 distinct 3-D topologies using the aforementioned SIMP formulation would require approximately 138 and 1,075 hours for the coarse and fine design domain discretizations, respectively. All analyses were performed on a computer with 132 GB of RAM and up to 12 total cores available. Thus, not only do the SPIDRS-generated frontiers compare favorably with those created using a SIMP formulation, but they can be obtained in approximately 2.5% of the computation time. This could greatly expedite the preliminary design step, and by extension the design process as a whole, by exploring the structural design space, identifying trends and commonalities between well-performing designs, and gaining insights into specific solutions, all in a fraction of the time required by traditional topology optimization methods.

4.2.1.2 Torsional Loading

An illustration of the cantilevered truss design problem considering a torsional loading condition is shown in Figure 4.16. The 3-D structural design domain is defined to have a length of 120 mm, a height of 40 mm, and a width of 40 mm. To provide an indication of the torsional stiffness of a given structural topology, a moment of $M_x = 0.005\,\mathrm{N}\cdot\mathrm{m}$ is applied to all nodes in the plane associated with the face $[3,2,6,7]$ and the resulting angular displacement θ_x is measured. Unlike the transverse loading problem, the torsional loading problem is defined to be non-symmetric.

The Pareto frontier for the multiobjective 3-D topology optimization problem illustrated in Figure 4.16 is shown in Figure 4.17, along with several topologies of interest. As in the transverse loading case (cf., Section 4.2.1.1), the lightest and least stiff design is simply the topology defined by the initial graph. The stiffest and heaviest design features a series of triangular structures that alternate in tension and compression to oppose the rotation generated by the applied moment. Perhaps the most interesting topology generated by the SPIDRS algorithm is the design closest to

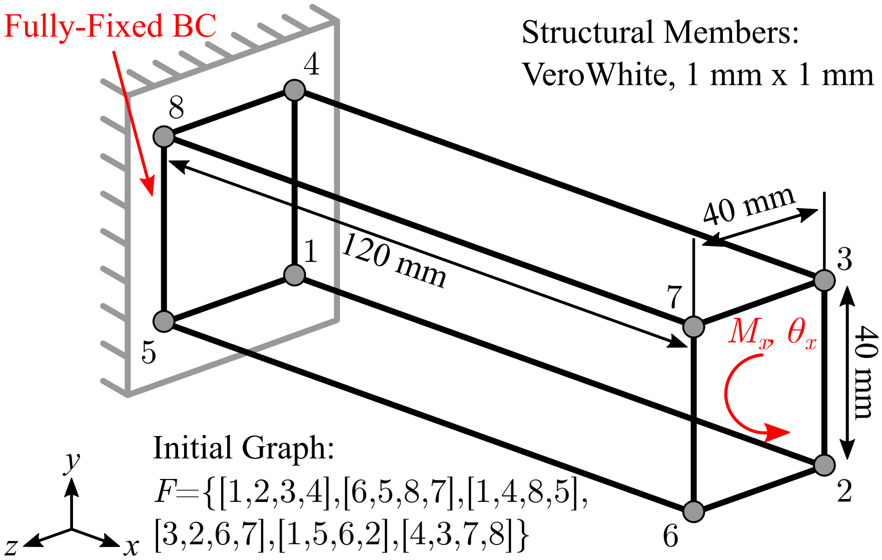


Figure 4.16: Initial graph and boundary conditions associated with the 3-D cantilevered frame subjected to torsional loading.

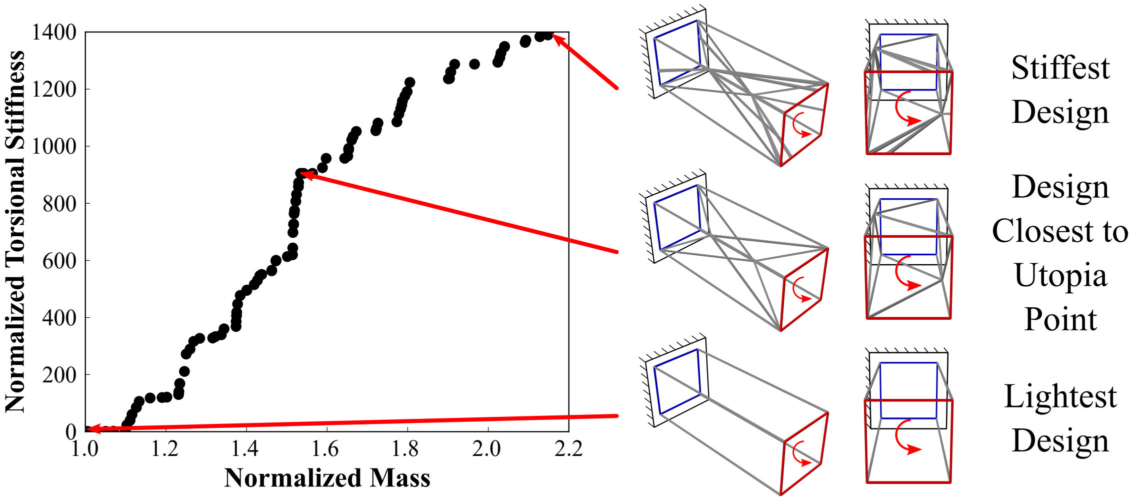


Figure 4.17: Pareto frontier associated with a 3-D cantilevered frame subjected to torsional loading (cf., Figure 4.16), along with several 3-D structural topologies of interest generated by SPIDRS. Structural members at each end are colored to better illustrate the topology.

the utopia point, which features a similar series of triangular structures. However, this topology demonstrates symmetry across the plane illustrated in Figure 4.18. This is a fascinating result when considering that no symmetry boundary conditions were specified in the design problem, and further demonstrates the ability of the SPIDRS algorithm to effectively explore a 3-D structural design domain.

4.2.2 Compliant Mechanisms

The second set of 3-D topology optimization problems features three compliant mechanisms inspired by the work of Ansola et al. [125], who explored extending the evolutionary structural optimization (ESO) method into 3-D. The initial state of each graph is defined by

$$F = \{[1,2,3,4],[6,5,8,7],[1,4,8,5],[3,2,6,7],[1,5,6,2],[4,3,7,8].\}$$

A description of each multiobjective optimization problem is given in Table 4.3. As in previous compliant mechanism studies in this work (cf., Sections 2.2.2, 3.4.2, and 3.6.4.2), the goal of these problems is to maximize both the displacement inversion ratio r_δ and force inversion ratio r_f.

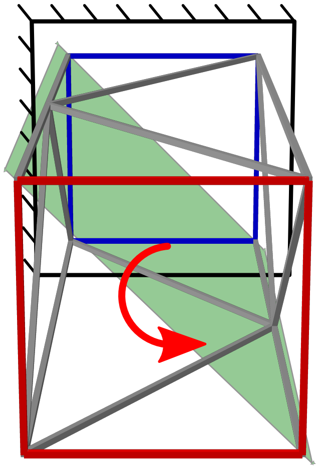


Figure 4.18: Despite the lack of symmetry boundary conditions defined in the problem, the design closest to the utopia point generated by the SPIDRS algorithm demonstrates symmetry about the highlighted plane.

Created structural members are assigned a constant cross-section of $2\,\text{mm} \times 2\,\text{mm}$, and structural analysis considers a geometrically nonlinear FEA model with an average element length of $30\,\text{mm}$ and a stress constraint in structural members of $\sigma^{max}_{Mises} = 33.8\,\text{MPa}$. Given that analysis allows for the buckling and large rotations of structural members, each problem analyzes $60 \times 400 = 24,000$ individual designs.

Table 4.3: Specifications for the multiobjective 3-D topology optimization problems considering various compliant mechanisms using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Maximize:	r_δ, r_f
by varying:	2 axiom characters,
	4 rule assignments (18 genes each),
subject to:	$\sigma^{max}_{Mises} \leq 33.8\,\text{MPa}$
NSGA-II Parameters [68]	
60 members for 400 generations,	
P_{cross} = 0.9, η_{cross} = 20,	
P_{mut} = 1/52, η_{mut} = 20	

4.2.2.1 Tensile Inverter

The first 3-D compliant mechanism problem considers a tensile inverter similar to those explored previously in this work and shown in Figure 4.19. The 3-D structural design domain is defined to have a length of 250 mm, a height of 125 mm, and a width of 125 mm. Symmetry boundary conditions are applied about the x-y plane associated with the initial face $[1, 2, 3, 4]$ and about the x-z-plane associated with the initial face $[1, 5, 6, 2]$ such that only a quarter of the structural topology generated. A displacement boundary condition of $u_x = \delta_{x,in}$ is applied to node 1, while fully-fixed boundary conditions (i.e., $u_x = u_y = u_z = 0$) are applied to nodes 4 and 5. Note that the edge between nodes 1 and 2 and all associated subdivisions thereof are removed from the final graph topology. As in previous compliant mechanism studies, two distinct loading steps are considered. First, a displacement of $\delta_{x,in} = -5$ mm is applied to node 1, while the resulting output displacement $\delta_{x,out}$ at node 2 is measured. Next, node 2 is forced back to its reference positions (i.e, $\delta_{x,out} = 0$) and the reaction forces at nodes 1 and 2 ($f_{x,in}$ and $f_{x,out}$, respectively) are mea-

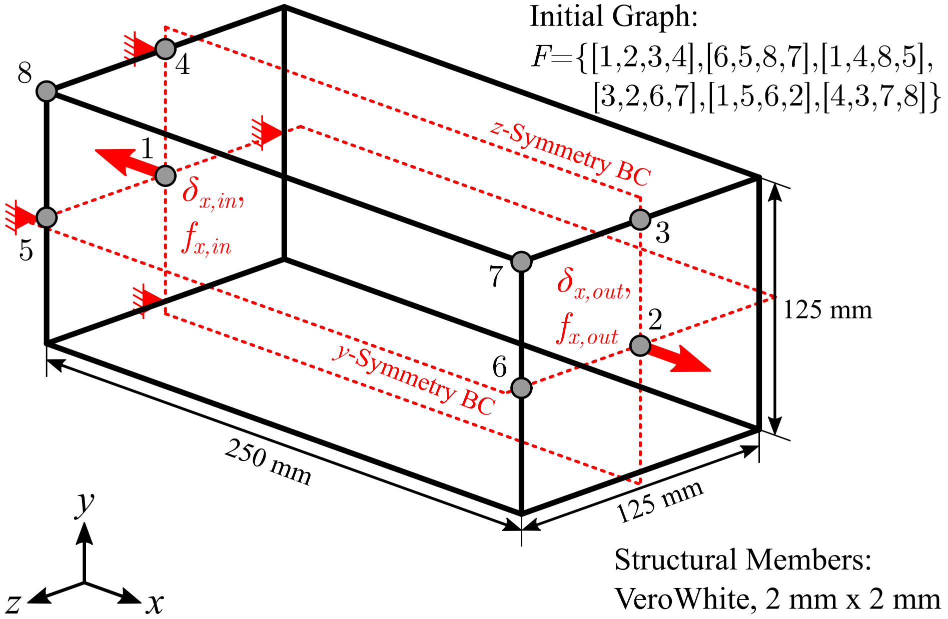


Figure 4.19: Initial graph and boundary conditions associated with the 3-D tensile inverter design study.

sured. Thus, the displacement and force inversion ratios that are to be maximized for this problem are defined as

$$r_\delta = \left.\frac{\delta_{x,out}}{\delta_{x,in}}\right|_{f_{x,out}=0} \quad \text{and} \quad r_f = \left.\frac{-f_{x,out}}{f_{x,in}}\right|_{\delta_{x,out}=0} . \tag{4.1}$$

The resulting Pareto frontier for the 3-D tensile inverter problem is shown in Figure 4.20, along with several SPIDRS-generated solutions of interest. Also plotted is the ideal solution for the problem assuming a linear relationship between force and displacement such that $r_\delta = 1/r_f$ (cf., Equation 2.3). It is noticeable that solutions along the frontier generated by the SPIDRS algorithm are much further away from the ideal solution than those seen in the 2-D problem discussed in Section 3.6.4.2. This is believed to be due to the additional strain energy stored in buckling structural members along axes of symmetry that oppose the desired response of the mechanism (e.g., the edge between nodes 5 and 6 in Figure 4.19). Observing the example topologies generated by SPIDRS, it is immediately apparent that the creation of structural members is concentrated in the plane associated with the initial face $[1, 2, 3, 4]$, essentially creating a 2-D tensile inverter inside of a 3-D structural design domain. This is not unexpected, as this is the simplest way to

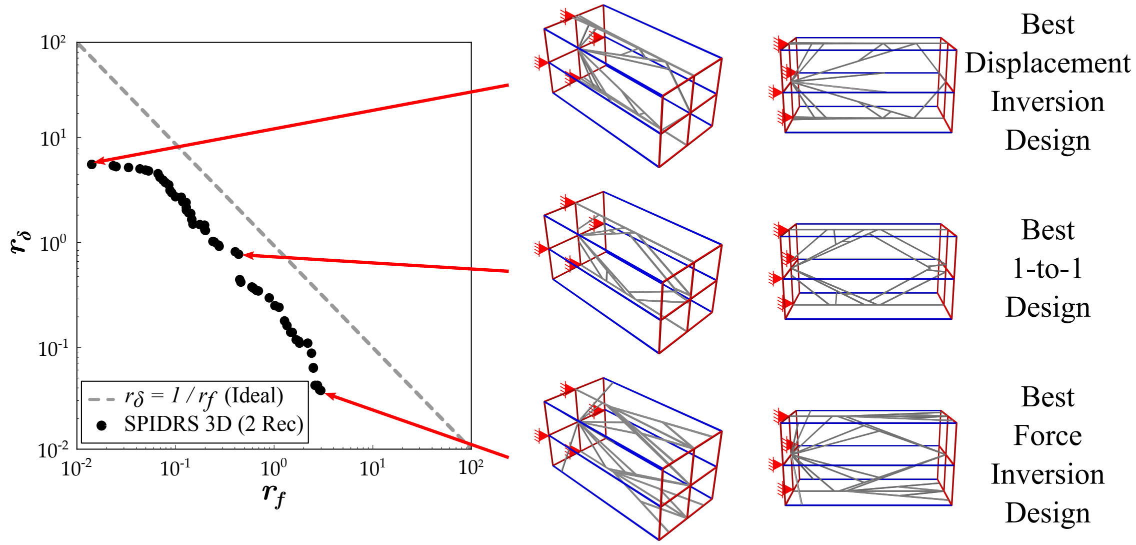


Figure 4.20: Pareto frontier associated with the 3-D tensile inverter design study generated by the SPIDRS algorithm. Structural members at each end are colored to better illustrate the topology.

generate the load paths needed for meaningful structural responses. Additionally, it is interesting to note that once again the load paths of these topologies are the same as those seen in previous 2-D tensile inverter studies and follow similar trends to those observed in Figure 3.28. The deformed configuration of the closest 1-to-1 SPIDRS solution during the displacement inversion load case is shown in Figure 4.21. Comparing this solution solution with the solution obtained by Ansola et al. (Figure 4.22), one can observe similarities in the load paths of both mechanisms, although the solution obtained using ESO features additional load paths in the x-x plane that are not featured in any SPIDRS solution.

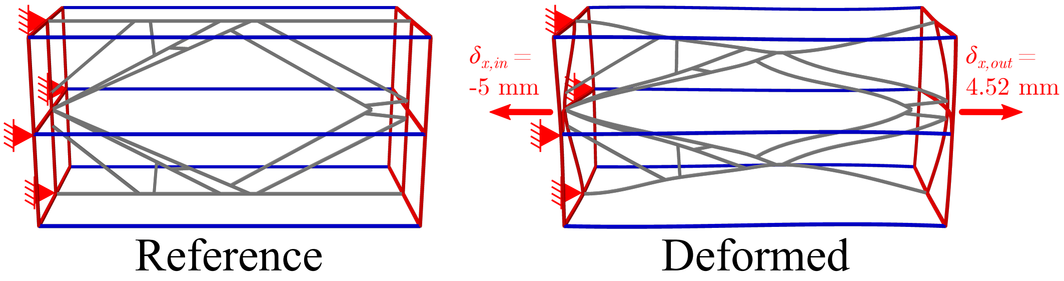
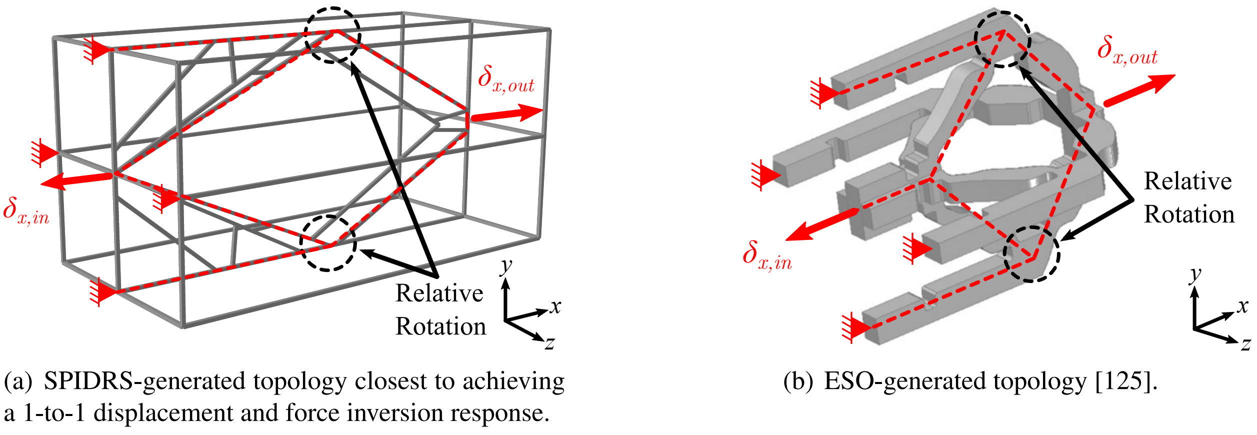


Figure 4.21: Reference and deformed configurations of the SPIDRS-generated 3-D tensile closest to achieving a 1-to-1 displacement and force inversion response (deformation scale factor of 3).



(a) SPIDRS-generated topology closest to achieving a 1-to-1 displacement and force inversion response.

(b) ESO-generated topology [125].

Figure 4.22: Comparison of load path geometries (denoted by dashed red lines) for a 3-D tensile inverter generated using a) SPIDRS and b) a density-based topology optimization approach.

4.2.2.2 *Elevator*

The second 3-D compliant mechanism problem considers an *elevator* mechanism in which the applied input and corresponding output displacements are oriented along different axes as outlined in Figure 4.23. The 3-D structural domain is defined to have a length of 250 mm, a height of 62.5 mm, and a width of 150 mm. Symmetry boundary conditions are placed on the x-y plane associated with the initial face $[1, 5, 6, 2]$ and the y-z plane associated with the initial face $[1, 5, 6, 2]$ such that only a quarter of the structural topology is generated. A fully-fixed boundary condition (i.e., $u_x = u_y = u_z = 0$) is applied to node 1, and an input displacement of $u_x = \delta_{x,in}$ is applied to node 8. Inspired by the results of the previous section, all edges along symmetry planes and subdivisions thereof are removed from the final graph topology with the exception of the edge between nodes 6 and 7, which is kept to insure that the node defining the output point (node 7) always exists in the final graph. As with the 3-D tensile inverter, two distinct loading conditions are considered. First, an input displacement of $\delta_{x,in} = -5$ mm is applied to node 8, and the corresponding output displacement in the y-direction at node 7, $\delta_{y,out}$, is measured. Next, node 7 is returned to its reference configuration ($\delta_{y,out} = 0$) and the reaction forces at nodes 8 and 7 ($f_{x,in}$

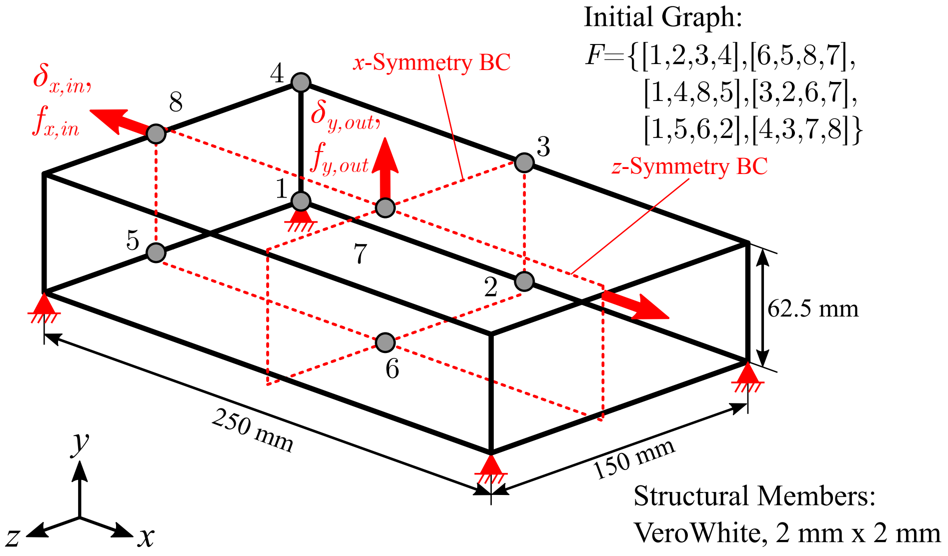


Figure 4.23: Initial graph and boundary conditions associated with the 3-D elevator design study.

and $f_{y,out}$, respectively) are measured. With this change in desired behavior, the displacement and force inversion ratios to be maximized for the elevator mechanism are defined as

$$r_\delta = \left.\frac{\delta_{y,out}}{\delta_{x,in}}\right|_{f_{y,out}=0} \quad \text{and} \quad r_f = \left.\frac{-f_{y,out}}{f_{x,in}}\right|_{\delta_{y,out}=0}. \tag{4.2}$$

The SPIDRS-generated Pareto frontier associated with the elevator mechanism is shown in Figure 4.24. These solutions demonstrate much better agreement with the ideal solution curve defined by Equation 2.3 (which remains valid for this problem formulation despite the change in orientation of the desired output behavior) than was seen from the frontier associated with the 3-D tensile inverter problem. While the elimination of structural members along the axes of symmetry may help contribute to this, an examination of the deformed configuration of the SPIDRS-generated elevator closest to achieving a 1-to-1 displacement and force inversion behavior illustrated in Figure 4.25 shows that there is actually a relatively small amount of buckling in structural members throughout the mechanism. Figure 4.26a illustrates the load path of this mechanism, which is also seen in the solution generated by Ansola et al. using a density-based topology optimization

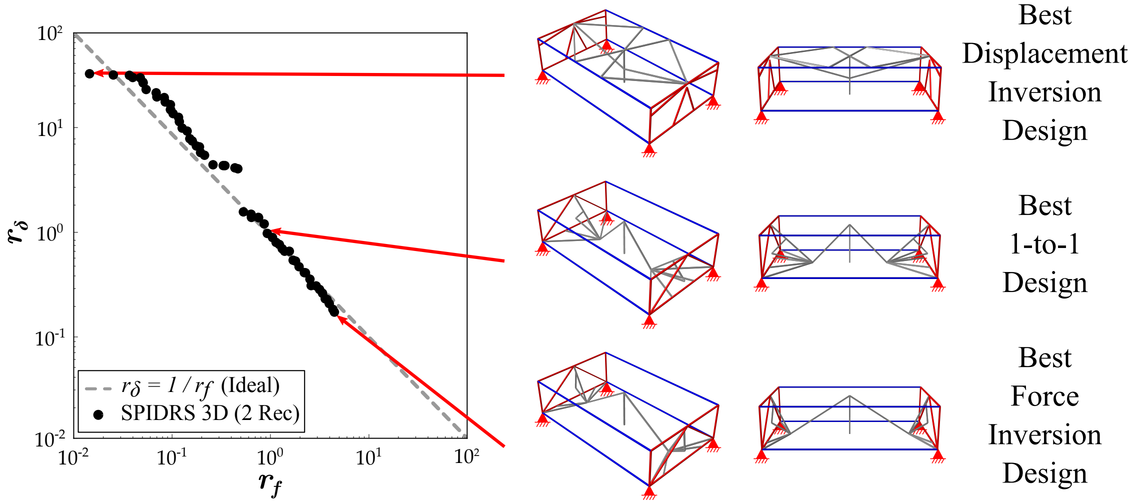


Figure 4.24: Pareto frontier associated with the 3-D elevator design study generated by the SPIDRS algorithm. Structural members at each end are colored to better illustrate the topology.

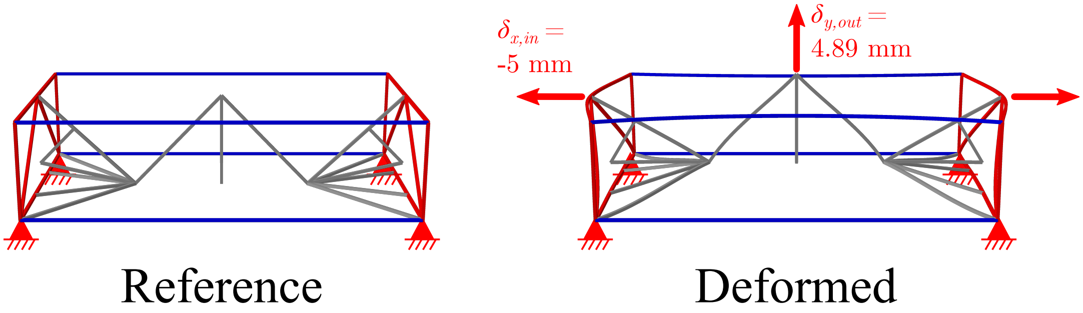
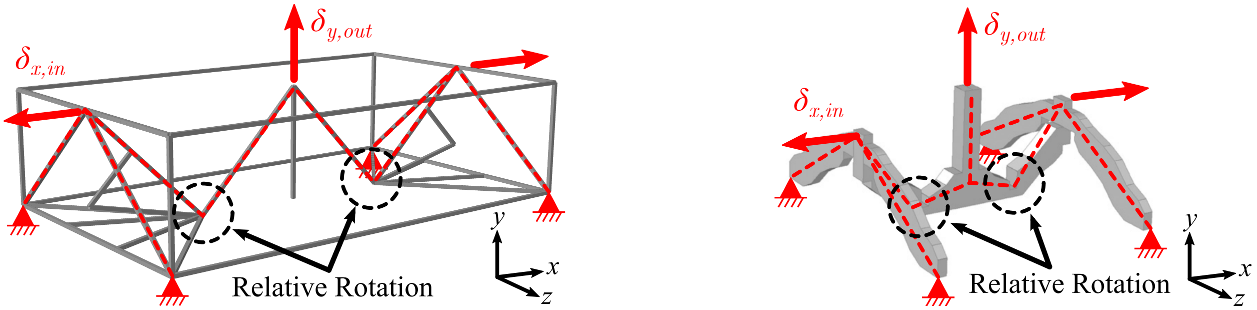


Figure 4.25: Reference and deformed configurations of the SPIDRS-generated 3-D elevator mechanism closest to achieving a 1-to-1 displacement and force inversion response (deformation scale factor of 3).



(a) SPIDRS-generated topology closest to achieving a 1-to-1 displacement and force inversion response.

(b) ESO-generated topology [125].

Figure 4.26: Comparison of load path geometries (denoted by dashed red lines) for a 3-D elevator mechanism generated using a) SPIDRS and b) a density-based topology optimization approach.

approach (Figure 4.26b). The triangular structure that forms between the input node, the applied fully-fixed boundary conditions, and a point located on the x-z plane act similar to a rigid body and translates the input displacement into a relative rotation. A lever-like structural member then uses this rotation to drive the output point upwards in the positive y-direction. In fact, one can say that this same behavior would be seen in the various tensile inverter problems explored in this work if those problems did not make use of symmetric boundary conditions which force the resulting output displacements to be oriented along the x-axis.

4.2.2.3 Cruncher

The third and final 3-D compliant mechanism problem considered is a *cruncher* mechanism which not only features applied input and corresponding output displacements oriented in different directions, but also places these points in different planes relative to the initial graph (cf., Figure 4.27). The 3-D structural design domain is defined to have a length of 250 mm, a height of 125 mm, and a width of 125 mm. Symmetry boundary conditions are applied about the x-y plane associated with the initial face $[1,2,3,4]$ and about the x-z-plane associated with the initial face $[1,5,6,2]$ such that only a quarter of the structural topology generated. A displacement boundary condition of $u_y = \delta_{y,in}$ is applied to node 8, while fully-fixed boundary conditions (i.e., $u_x = u_y = u_z = 0$) are applied to node 3. As with the elevator mechanism, all edges along symmetry planes and subdivisions thereof are removed from the final graph topology; additionally, the edge between nodes 5 and 8 is also removed due to the boundary conditions applied to the problem. The two distinct loading cases to test for the displacement and force inversion behavior of

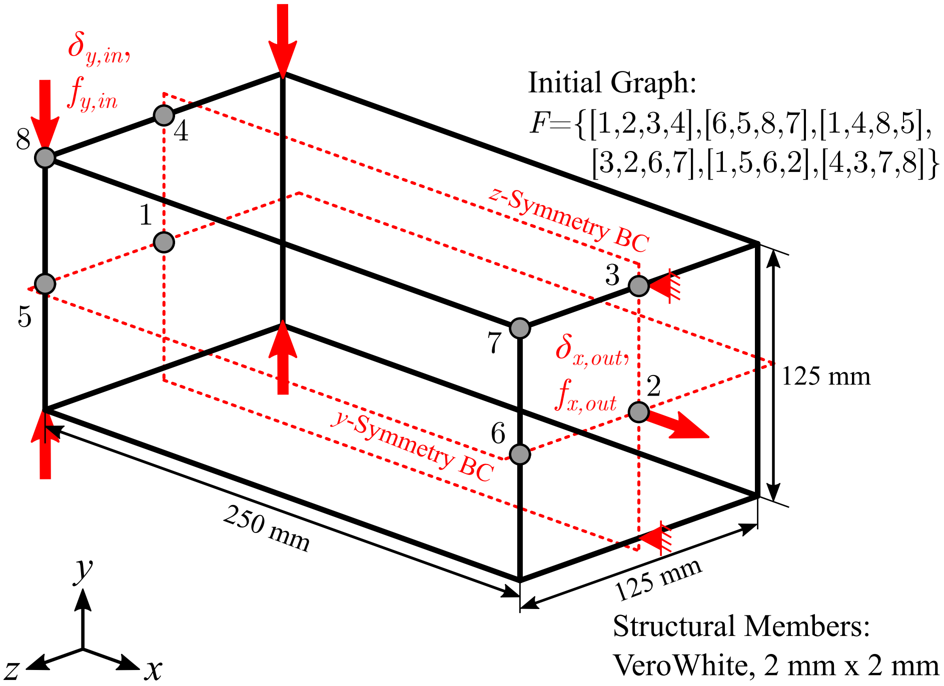


Figure 4.27: Initial graph and boundary conditions associated with the 3-D cruncher design study.

each generated topology are defined similarly to the previous two design problems. First, an input displacement of $\delta_{y,in} = -5\,\mathrm{mm}$ is applied to node 8, and the corresponding output displacement $\delta_{x,out}$ at node 2 is measured. Next, node 2 is returned to its original configuration ($\delta_{x,out} = 0$) and the resulting reaction forces at nodes 8 and 2 ($f_{y,in}$ and $f_{x,out}$, respectively) are measured. The displacement and force inversion ratios for the cruncher mechanism are given as

$$r_\delta = \left.\frac{\delta_{x,out}}{\delta_{y,in}}\right|_{f_{x,out}=0} \quad \text{and} \quad r_f = \left.\frac{-f_{x,out}}{f_{y,in}}\right|_{\delta_{x,out}=0}. \tag{4.3}$$

The Pareto frontier consisting of SPIDRS-generated cruncher mechanisms is shown in Figure 4.28 along with several topologies of interest. Once again, the frontier generated by the SPIDRS algorithm agrees well with the ideal solution defined by Equation 2.3. As with the elevator mechanism design problem, the SPIDRS-generated design closest to achieving a 1-to-1 displacement and force inversion ratio shown in Figure 4.29 features a relatively low amount of buckling of structural members in its deformed configuration. The geometry of the load path for this topology is illustrated by the dashed lines in Figure 4.30a, which again features the input

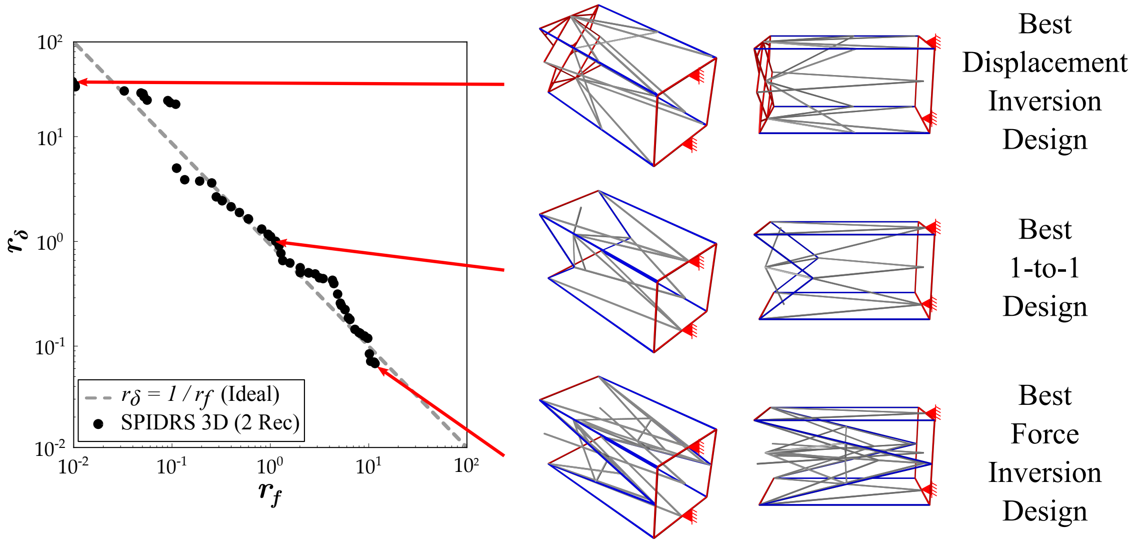


Figure 4.28: Pareto frontier associated with the 3-D cruncher design study generated by the SPIDRS algorithm. Structural members at each end are colored to better illustrate the topology.

displacement being transmitted to a corresponding output response via a relative rotation about a specific node of the graph. This same load path geometry is also seen in the topology reported by Ansola et al., demonstrating once again that SPIDRS is capable of generating similar mechanisms to those created using more traditional topology optimization methodologies.

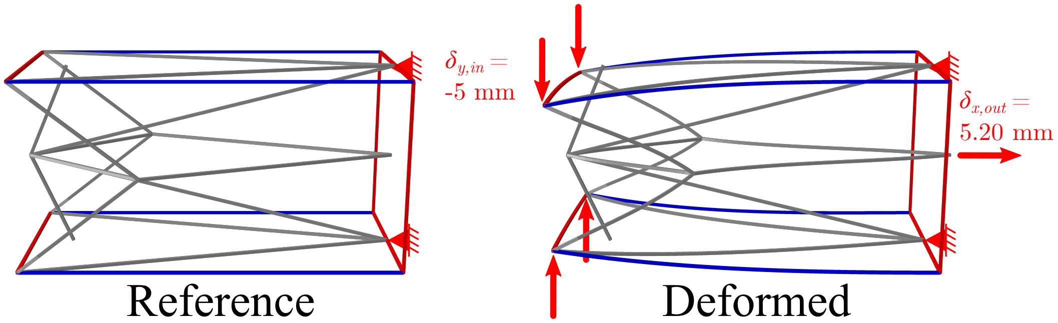
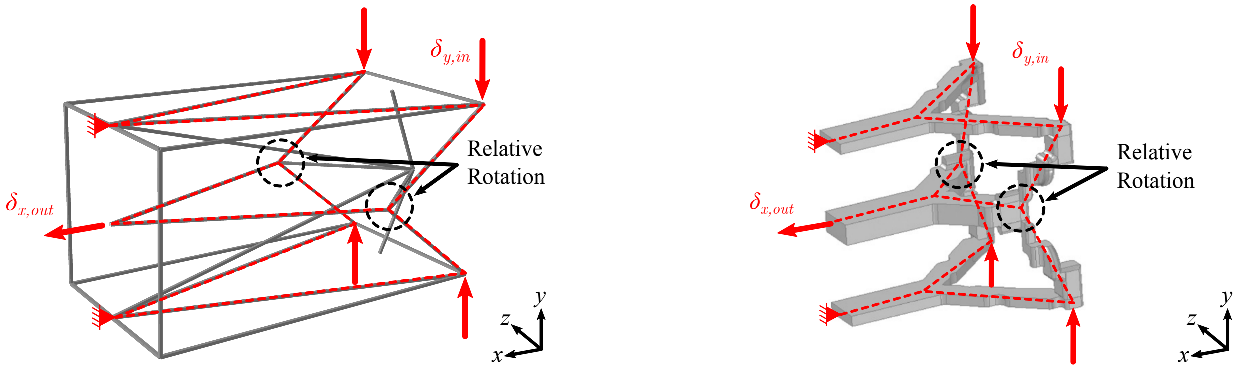


Figure 4.29: Reference and deformed configurations of the SPIDRS-generated 3-D cruncher mechanism closest to achieving a 1-to-1 displacement and force inversion response (deformation scale factor of 3).



(a) SPIDRS-generated topology closest to achieving a 1-to-1 displacement and force inversion response.

(b) ESO-generated topology [125].

Figure 4.30: Comparison of load path geometries (denoted by dashed red lines) for a 3-D elevator mechanism generated using a) SPIDRS and b) a density-based topology optimization approach.

5. SIZING OPTIMIZATION OF GRAPH-BASED L-SYSTEM-GENERATED TOPOLOGIES

Chapters 3-4 have demonstrated the effectiveness of the proposed parameterized L-System framework in creating structural topologies the compare well with optimal or ideal solutions and in some cases outperform traditional topology optimization methodologies. However, one of the recognized inherent drawbacks of this method is the existence and creation of unnecessary structural members that form either indirect or auxiliary load paths within a structural topology. This arises both from the necessity of an initial graph on which SPIDRS can operate and from the fact that SPIDRS creates structural members based only on a parameterized L-System encoding generated by a genetic algorithm and has no information regarding how these members influence the performance of the structure. The existence of these indirect or auxiliary load paths can artificially inflate structural mass measurements, which is crucial when mass is an optimization objective, and requires increased computational run-times due to the existence of the elements that comprise them.

Additionally, to this point all L-System topology optimization studies in this work have assumed a constant predetermined cross-section (and therefore thickness or cross-sectional area) for all structural members that make up a given structural topology. As first observed in Figure 2.7, density-based and level set topology optimization methodologies have no such restriction, and thus can achieve improved structural performance by thickening select members of the structure. While the thickness (or any other attribute that could impact performance) assigned to a given edge could be assigned and changed within the SPIDRS algorithm in a manner consistent with material assignments discussed in Chapter 3, this is beyond the scope of this work. Furthermore, the elimination of superfluous and thickening of critical structural members can further clarify trends and commonalities between well-performing designs and elucidate a better understanding of the mechanisms behind specific solutions, which is critical to the preliminary design process. Thus, this chapter will focus on the inclusion of a *sizing optimization* scheme within the overall topology optimization framework to eliminate unnecessary structural members and allow for variable member thicknesses to further optimize the structural performance of a given topology.

In the traditional topology optimization methods discussed in Chapter 1, sizing optimization is often implemented in the form of length-scale constraints [8, 15, 16]. These restrictions on the length or thickness of structural members can be used to satisfy manufacturing constraints or avoid numerical issues such as checkerboarding. However, modifying this constraint can result in a change in the optimal structural topology as illustrated in Figure 5.1. Given that SPIDRS-generated topologies consist of sets of discrete frame elements, the proposed addition of a sizing optimization scheme closely resembles the procedure used by the ground structure method (GSM) (cf., Chapter 1), where the cross-sectional area of discrete members are varied using sensitivity analysis [126] to improve structural performance and members are removed when their cross-sectional area reaches zero [43] as shown in Figure 5.2. The GSM, however, is somewhat limited by its dependence on a predefined, regular layout of nodal locations and discrete structural members. Replacing the ground structure with a SPIDRS-generated graph may represent an improvement, as SPIDRS has demonstrated the capability of producing irregular, well-performing topologies that would be difficult to produce using the GSM.

This chapter details the creation of a hybrid genetic/gradient-based optimization framework capable of considering both the topology and sizing optimization of parameterized L-System-

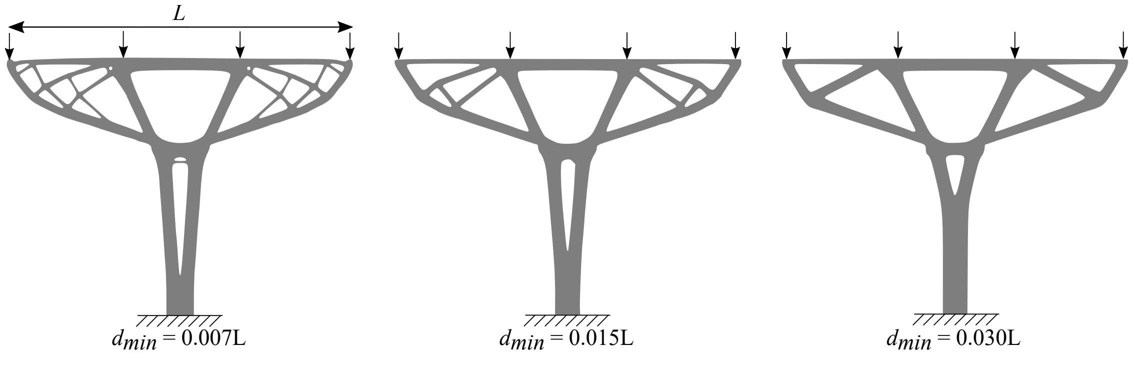


Figure 5.1: Effects of changing the thickness constraint d_{min} on the optimized topology of a hammerhead pier using a density-based topology optimization approach (adapted from Gaynor [127]), where L is the length of the pier.

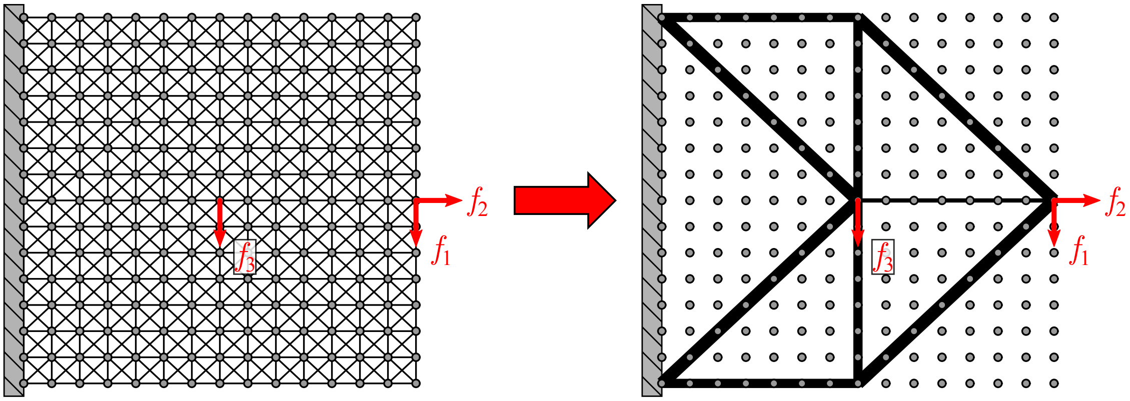


Figure 5.2: Demonstration of the use of the GSM on a multi-load minimum compliance topology optimization problem (adapted from Bendsøe et al. [43]). Notice that the GSM starts with a predefined, regular layout of discrete structural members, then uses sensitivity analysis to determine the thickness of each member such that the structural response is optimized.

generated structural topologies and is organized as follows: Section 5.1 introduces the hybrid optimization framework in the context of 2-D problems, including a derivation of 2-D frame FEA theory, an overview of structural sensitivity analysis, how the genetic topology and gradient-based sizing optimization approaches are combined for multiobjective problems, and a demonstration of the proposed framework using both cantilevered frame and compliant mechanism design problems; and Section 5.2 explores extending this frame to consider 3-D problems, focusing on 3-D frame FEA theory, the associated changes to sensitivity analysis calculations, and examples of the framework being applied to 3-D cantilevered frame design problems.

5.1 Two-Dimensional Sizing Optimization

5.1.1 2-D Frame FEA

To this point, the proposed topology optimization framework has utilized the Abaqus FEA suite for structural analysis. However, at the time of writing there is no straightforward method for obtaining the sensitivities of certain output variables with respect to specified design parameters (e.g., structural member thickness) from Abaqus results, which is a key component of the hybrid optimization framework being developed. Therefore, this framework will utilize a linear frame

FEA solver written in Python from which these sensitivities are easily obtainable. The remainder of this section will provide a brief overview of the theory behind 2-D linear frame analysis (see [128, 126]).

The fundamental system of equations for finite element model is given as

$$[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{F}\}. \tag{5.1}$$

In the context of a single frame element with three degrees of freedom (two translational, one rotational), such as the one illustrated in Figure 5.3,

$$[\mathbf{K^e}] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{5.2}$$

is the element stiffness matrix, where A, E, I, and L are the cross-sectional area, elastic modulus,

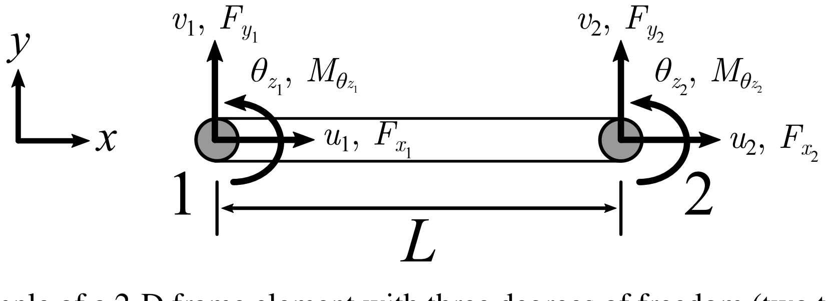


Figure 5.3: Example of a 2-D frame element with three degrees of freedom (two translational, one rotational).

area moment of inertia, and length associated with the element, respectively;

$$\{\mathbf{u^e}\} = \left\{ \begin{array}{cccccc} u_1 & v_1 & \theta_{z_1} & u_2 & v_2 & \theta_{z_2} \end{array} \right\}^T \tag{5.3}$$

is the element displacement vector; and

$$\{\mathbf{F^e}\} = \left\{ \begin{array}{cccccc} F_{x_1} & F_{y_1} & M_{\theta_{z_1}} & F_{x_2} & F_{y_2} & M_{\theta_{z_2}} \end{array} \right\}^T \tag{5.4}$$

is the element force vector. Note that the element stiffness matrix in Equation 5.2 is defined in the element coordinate system such that the x-axis is parallel with the element axis and the y-axis is perpendicular to the element axis (cf., Figure 5.3). To convert between the element coordinate system and the global coordinate system, a coordinate transformation matrix of the form

$$[\mathbf{T}] = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

is used, where θ is the angle between the element axis and the global x-axis. Applying Equation 5.5 to Equations 5.2 and 5.4 leads to a system of equations for a single frame element in the global coordinate system,

$$[\mathbf{T}]^T [\mathbf{K^e}] [\mathbf{T}] \{\mathbf{u^e}\} = [\mathbf{T}]^T \{\mathbf{F^e}\}. \tag{5.6}$$

The system of equations for a structure consisting of multiple frame elements is formed by assembling Equation 5.6 for each individual frame element into a single linear system of equations in the form of Equation 5.1, where common nodes share degrees of freedom. Once the global system of equations has been assembled, the global displacement vector can be determined algebraically by multiplying both sides of Equation 5.1 by the inverse of the global stiffness matrix, $[\mathbf{K}]^{-1}$. However, inverting the global stiffness matrix can be computationally burdensome for even rudimentary structures, and several numerical alternatives have been proposed to increase computational efficiency. One such approach, known as LU decomposition, factors the matrix as the product of lower- and upper-triangular matrices, allowing the problem to be split into two systems of equations that can be solved directly using backward substitution and eliminating the need to directly compute the full inverse of the global stiffness matrix. For a more detailed discussion on LU decomposition and similar techniques, see Kincaid and Cheney [129]. In this work, an implementation of LU decomposition included in the *NumPy* Python module is utilized.

5.1.2 Sensitivity Analysis

Determining the thickness of each discrete member that results in the optimal structural performance of the SPIDRS-generated topology requires a quantification of the effect these thicknesses have on certain output variables. The proposed hybrid optimization framework utilizes structural sensitivity analysis as detailed by Choi and Kim [126], which is especially efficient when derivatives can be obtained analytically using explicit expressions of the finite element matrix equations (cf., Equations 5.2-5.4). To begin, one takes the derivative of Equation 5.1 with respect to a given structural member thickness t:

$$[\mathbf{K}(t)]\left\{\frac{d\mathbf{u}(t)}{dt}\right\}+\left[\frac{d\mathbf{K}(t)}{dt}\right]\{\mathbf{u}(t)\}=\left\{\frac{d\mathbf{F}}{dt}\right\}. \tag{5.7}$$

Herein, it is assumed that the force vector is independent of the thickness associated with each member such that $\left\{\frac{d\mathbf{F}}{dt}\right\}=\{\mathbf{0}\}$. Thus, Equation 5.7 can be rearranged as

$$[\mathbf{K}(t)]\left\{\frac{d\mathbf{u}(t)}{dt}\right\}=-\left[\frac{d\mathbf{K}(t)}{dt}\right]\{\mathbf{u}(t)\}. \tag{5.8}$$

Multiplying both sides of Equation 5.8 by the inverse of the global stiffness matrix, $[\mathbf{K}]^{-1}$, one can obtain the equation for the sensitivity of the displacement vector with respect to a given structural member thickness, t,

$$\left\{ \frac{d\mathbf{u}(t)}{dt} \right\} = [\mathbf{K}(t)]^{-1} \left(- \left[\frac{d\mathbf{K}(t)}{dt} \right] \{\mathbf{u}(t)\} \right). \tag{5.9}$$

Sensitivity analysis is typically performed after structural analysis has already been carried out (i.e., Equation 5.1 has already been solved). Therefore, the only contribution to Equation 5.9 that must be calculated is $\left[\frac{d\mathbf{K}(t)}{dt}\right]$. Assuming a cross-sectional area of $A = wt$ and an area moment of inertial of $I = wt^3/12$, where t is the thickness and w is the out-of-plane width of a given structural member, Equation 5.2 can be rewritten as

$$[\mathbf{K}^{\mathbf{e}}] = \begin{bmatrix} \frac{Ewt}{L} & 0 & 0 & -\frac{Ewt}{L} & 0 & 0 \\ 0 & \frac{Ewt^3}{L^3} & \frac{Ewt^3}{2L^2} & 0 & -\frac{Ewt^3}{L^3} & \frac{Ewt^3}{2L^2} \\ 0 & \frac{Ewt^3}{2L^2} & \frac{Ewt^3}{3L} & 0 & -\frac{Ewt^3}{2L^2} & \frac{Ewt^3}{6L} \\ -\frac{Ewt}{L} & 0 & 0 & \frac{Ewt}{L} & 0 & 0 \\ 0 & -\frac{Ewt^3}{L^3} & -\frac{Ewt^3}{2L^2} & 0 & \frac{Ewt^3}{L^3} & -\frac{Ewt^3}{2L^2} \\ 0 & \frac{Ewt^3}{2L^2} & \frac{Ewt^3}{6L} & 0 & -\frac{Ewt^3}{2L^2} & \frac{Ewt^3}{3L} \end{bmatrix}. \tag{5.10}$$

Taking the derivative of Equation 5.10 with respect to t, one arrives at

$$\left[\frac{d\mathbf{K}^{\mathbf{e}}}{dt}\right] = \begin{bmatrix} \frac{Ew}{L} & 0 & 0 & -\frac{Ew}{L} & 0 & 0 \\ 0 & \frac{3Ewt^2}{L^3} & \frac{3Ewt^2}{2L^2} & 0 & -\frac{3Ewt^2}{L^3} & \frac{3Ewt^2}{2L^2} \\ 0 & \frac{3Ewt^2}{2L^2} & \frac{Ewt^2}{L} & 0 & -\frac{3Ewt^2}{2L^2} & \frac{Ewt^2}{2L} \\ -\frac{Ew}{L} & 0 & 0 & \frac{Ew}{L} & 0 & 0 \\ 0 & -\frac{3Ewt^2}{L^3} & -\frac{3Ewt^2}{2L^2} & 0 & \frac{3Ewt^2}{L^3} & -\frac{3Ewt^2}{2L^2} \\ 0 & \frac{3Ewt^2}{2L^2} & \frac{Ewt^2}{2L} & 0 & -\frac{3Ewt^2}{2L^2} & \frac{Ewt^2}{L} \end{bmatrix}. \tag{5.11}$$

Notice that this matrix is defined in the element coordinate system and must be converted to the global coordinate system using the coordinate transformation matrix in a manner consistent with Section 5.1.1. Similarly, the element stiffness derivative matrices for multiple elements must be assembled into a single matrix as before; however, assuming that each structural member is represented by a single element and noticing that Equation 5.9 considers the effect of a single structural member thickness, all but one of the element stiffness derivative matrices will equal $[\mathbf{0}]$ and thus $\left[\frac{d\mathbf{K}}{dt}\right]$ will be a sparse matrix. Plugging the resulting global stiffness derivative matrix into Equation 5.9, one can obtain the effect that a given structural member thickness has on any component of the global displacement vector, $\left\{\frac{d\mathbf{u}}{dt}\right\}$.

5.1.3 Combined Topology/Sizing Optimization Framework

The implementation of a sizing optimization scheme within the overall topology optimization framework requires the development of a hybrid genetic/gradient-based optimization method. which is illustrated in Figure 5.4. Such an approach has been utilized in topology optimization problems where calculated gradients are used to inform or refine topologies generated using a genetic algorithm [63, 64]. A general multiobjective topology optimization problem to be considered by the genetic algorithm can be written as

$$\begin{aligned} \mathbf{J}^* = &\max_{\boldsymbol{\omega},\boldsymbol{P},\boldsymbol{\Sigma}} \ \{f_1(\boldsymbol{\omega},\boldsymbol{P},\Sigma), f_2(\boldsymbol{\omega},\boldsymbol{P},\Sigma)\} \\ \text{s.t.} \quad & g_k \leq 0, \quad k = 1 \ldots N_{leq}, \end{aligned} \tag{5.12}$$

where the goal is to find a set of N_{pop} solutions $\mathbf{J}^*$ defined as $\left\{J_1(\boldsymbol{t}_1), J_2(\boldsymbol{t}_2), \ldots, J_{N_{pop}}(\boldsymbol{t}_{N_{pop}})\right\}$ using the parameterized L-System variables $\boldsymbol{\omega}$, $\boldsymbol{P}$, and $\boldsymbol{\Sigma}$ such that the objectives f_1 and f_2 cannot simultaneously be improved for a given solution $J_i(\boldsymbol{t_i})$ subject to specified inequality constraints g_k for $k = 1 \ldots N_{leq}$. The genetic topology optimization framework remains consistent with that outlined in Chapter 3.15, where a set of design variables associated with the ith individual are generated by the NSGA-II genetic algorithm, encoded by the parameterized L-System introduced in Section 3.2.1, and interpreted to evolve the topological information of a graph by the SPIDRS algorithm detailed in Section 3.2. However, note that the efficient parallelization technique whereby an entire population of designs are analyzed simultaneously is not employed in the hybrid optimization framework; instead, single individuals in a population are analyzed sequentially. The

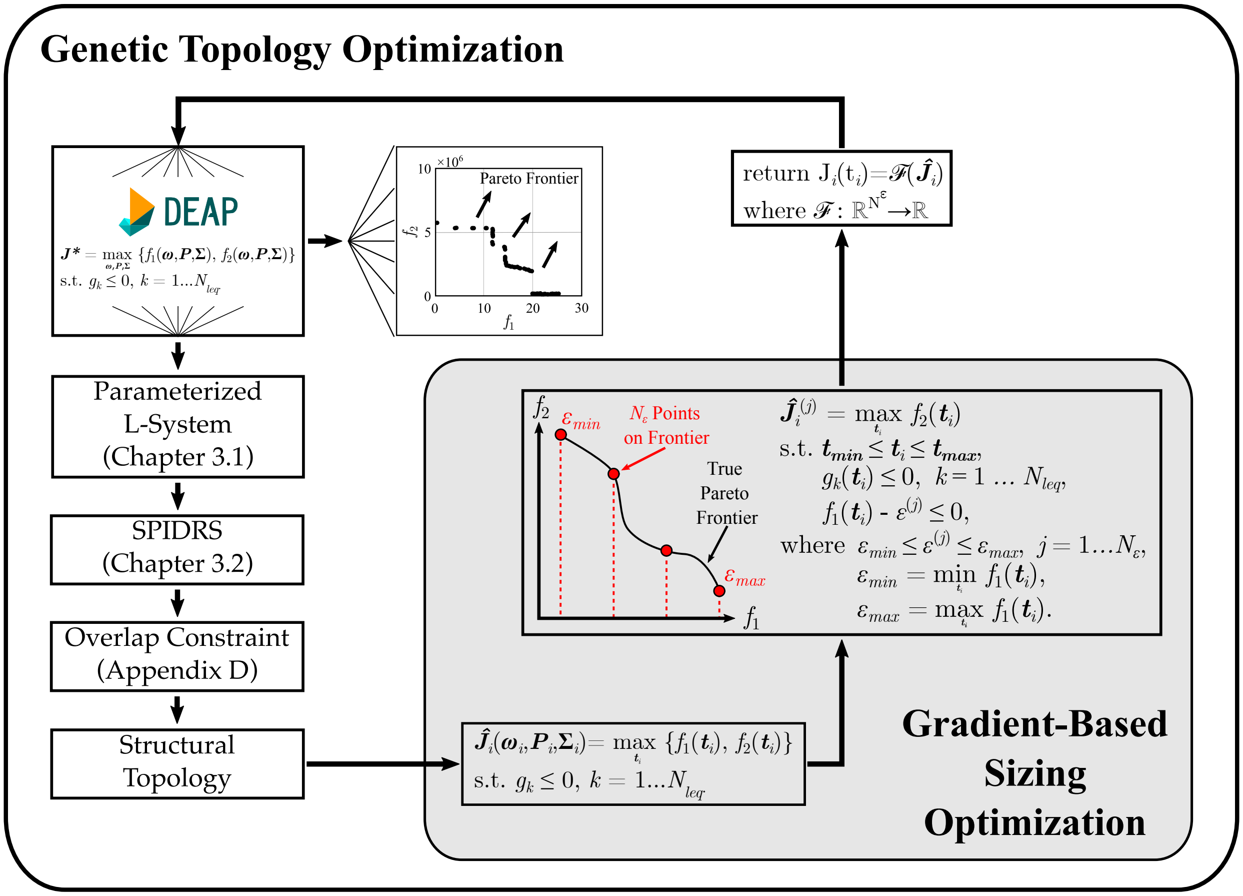

Genetic Topology Optimization
DEAP
$\boldsymbol{J^*} = \max_{\boldsymbol{\omega},\boldsymbol{P},\boldsymbol{\Sigma}} \{f_1(\boldsymbol{\omega},\boldsymbol{P},\boldsymbol{\Sigma}),\ f_2(\boldsymbol{\omega},\boldsymbol{P},\boldsymbol{\Sigma})\}$
s.t. $g_k \leq 0,\ k = 1 \ldots N_{leq}$
$\times 10^6$
10
5
0
f_2
Pareto Frontier
0
10
20
30
f_1
return $\mathrm{J}_i(\mathrm{t}_i) = \mathscr{F}(\hat{\boldsymbol{J}}_i)$
where $\mathscr{F}: \mathbb{R}^{N^\varepsilon} \to \mathbb{R}$
Parameterized L-System (Chapter 3.1)
SPIDRS (Chapter 3.2)
Overlap Constraint (Appendix D)
Structural Topology
f_2
ε_{min}
N_ε Points on Frontier
True Pareto Frontier
ε_{max}
f_1
$\hat{\boldsymbol{J}}_i^{(j)} = \max_{\boldsymbol{t}_i} f_2(\boldsymbol{t}_i)$
s.t. $\boldsymbol{t_{min}} \leq \boldsymbol{t}_i \leq \boldsymbol{t_{max}},$
$g_k(\boldsymbol{t}_i) \leq 0,\ \ k = 1 \ldots N_{leq},$
$f_1(\boldsymbol{t}_i)$ - $\varepsilon^{(j)} \leq 0,$
where $\varepsilon_{min} \leq \varepsilon^{(j)} \leq \varepsilon_{max},\ \ j = 1 \ldots N_\varepsilon,$
$\varepsilon_{min} = \min_{\boldsymbol{t}_i} f_1(\boldsymbol{t}_i),$
$\varepsilon_{max} = \max_{\boldsymbol{t}_i} f_1(\boldsymbol{t}_i).$
$\hat{\boldsymbol{J}}_i(\boldsymbol{\omega}_i,\boldsymbol{P}_i,\boldsymbol{\Sigma}_i) = \max_{\boldsymbol{t}_i} \{f_1(\boldsymbol{t}_i),\ f_2(\boldsymbol{t}_i)\}$
s.t. $g_k \leq 0,\ k = 1 \ldots N_{leq}$
Gradient-Based Sizing Optimization

Figure 5.4: Flowchart depicting the hybrid genetic/gradient-based optimization framework developed to couple topology and sizing optimization of L-System generated topologies. The gradient-based sizing optimization scheme utilizes a ε-constraint approach to decompose the multiobjective topology optimization problem into a series of constrained single-objective problems to construct a Pareto frontier for each topology.

topological information generated by SPIDRS is then given structural significance by defining the edges of the graph as frame elements, and the resulting structure is analyzed using the FEA frame theory developed in Section 5.1.1.

The gradient-based sizing optimization scheme uses sensitivity analysis (cf., Section 5.1.2) to determine the thickness of each structural member that results in the optimal performance of the

ith topology. This optimization problem can be written as

$$\begin{aligned} \hat{\boldsymbol{J}}_i(\boldsymbol{w}_i, \boldsymbol{P}_i, \boldsymbol{\Sigma}_i) &= \max_{\boldsymbol{t}_i} \ \{f_1(\boldsymbol{t}_i), f_2(\boldsymbol{t}_i)\} \\ \text{s.t.} \quad g_k &\leq 0, \quad k = 1 \dots N_{leq}, \end{aligned} \tag{5.13}$$

However, as discussed in Chapter 1, gradient-based optimization seeks to optimize a single scalar value, while the problems considered by the genetic topology optimization framework are almost exclusively multiobjective. Rather than combine the two objectives into a single objective via a weighted sum method, which often presupposes an intuition for the design problem at hand (cf., Chapter 1), the framework reduces the multiobjective problem into a single-objective problem using an ε-constraint approach [130]. More specifically, this approach decomposes the multiobjective constrained optimization problem given in Equation 5.13 into N_ε single-objective constrained optimization problems given as

$$\begin{aligned} \hat{J}_i^{(j)}(\boldsymbol{w}_i, \boldsymbol{P}_i, \boldsymbol{\Sigma}_i) &= \max_{\boldsymbol{t}_i} \ f_2(\boldsymbol{t}_i), \\ \text{s.t.} \quad & \boldsymbol{t_{min}} \leq \boldsymbol{t}_i \leq \boldsymbol{t_{max}} \\ & g_k \leq 0, \quad k = 1 \dots N_{leq}, \\ & f_1(\boldsymbol{t}_i) - \varepsilon^{(j)} \leq 0, \\ \text{where} \quad & \varepsilon_{min} \leq \varepsilon^{(j)} \leq \varepsilon_{max}, \quad j = 1 \dots N_\varepsilon, \end{aligned} \tag{5.14}$$

where $\boldsymbol{t}_i$ is the set of thicknesses corresponding to each member in the ith structural topology, $\boldsymbol{t_{min}}$ and $\boldsymbol{t_{max}}$ are sets of the minimum and maximum allowable member thicknesses, respectively, and ε_{min} and ε_{max} are bounds for the ε-constraint and defined as

$$\begin{aligned} \varepsilon_{min} &= \min_{\boldsymbol{t}_i} \ f_1(\boldsymbol{t}_i) \\ \text{s.t.} \quad & \boldsymbol{t_{min}} \leq \boldsymbol{t}_i \leq \boldsymbol{t_{max}} \\ & g_k \leq 0, \quad k = 1 \dots N_{leq} \end{aligned} \tag{5.15}$$

and

$$\begin{aligned} \varepsilon_{max} = &\max_{\boldsymbol{t}_i} \; f_1(\boldsymbol{t}_i) \\ \text{s.t.} \quad & \boldsymbol{t_{min}} \leq \boldsymbol{t}_i \leq \boldsymbol{t_{max}} \\ & g_k \leq 0, \quad k = 1 \dots N_{leq}, \end{aligned} \tag{5.16}$$

respectively. Each gradient-based optimization in the framework (e.g., Equations 5.14-5.16) uses the Sequential Least Squares Programming (SLSQP) algorithm [131] implemented in Python's *SciPy* module, which is capable of considering constrained, bounded objective functions and user defined sensitivity calculations (cf., Section 5.1.2).

The solution to the jth single-objective optimization given in Equation 5.14 represents a point on the Pareto frontier corresponding to $f_1 \leq \varepsilon^{(j)}$ and, for large portions of the frontier, $f_1 \approx \varepsilon^{(j)}$ in particular, as shown in Figure 5.4. However, the operations within the genetic algorithm require a single set of objective function values for a given set of genes, meaning that the set of solutions $\hat{\boldsymbol{J}}_i$ solved for using Equation 5.13 must be reduced to the single solution J_i needed in Equation 5.12. This reduction takes the form

$$\begin{aligned} J_i(\boldsymbol{t}_i) = \mathscr{F}(\hat{\boldsymbol{J}}_i) \\ \text{where} \quad \mathscr{F} : \mathbb{R}^{N_\varepsilon} \rightarrow \mathbb{R}, \end{aligned} \tag{5.17}$$

where $\mathscr{F}$ is a functional operator representing a specified decision metric. Depending on the relationship between the objective functions, this operator can range from choosing a point based on a knowledge of the given optimization problem (e.g., when maximizing stiffness while minimizing mass, an intuitive selection would be the point at which increasing mass does not result in a noticeable increase in stiffness) to simply selecting a point closest to the utopia point or some known optimal or ideal solution. The solution $J_i(\boldsymbol{t}_i)$ is then then returned to the genetic algorithm and the next individual is analyzed in the same manner.

5.1.4 Design Examples

To demonstrate the benefits of employing the proposed hybrid optimization framework to 2-D topology optimization problems, this section revisits the cantilevered frame and tensile inverter design problems discussed in Chapter 3.

5.1.4.1 Cantilevered Frame

The cantilevered frame problem discussed in Section 3.6.4.1 is reconsidered here with the same dimensions and boundary conditions (cf., Figure 3.16), initial graph, and NSGA-II parameters. As before, the goal of the problem is to minimize the mass while maximizing the stiffness of a structural topology (cf., Table 3.2). The addition of a gradient-based sizing optimization scheme (cf., Section 5.1.3) allows for the thickness of a given structural member t, to vary, but requires gradient calculations to quantify the relationship between this thickness and the objective functions. The sensitivity of the mass m of the structural topology with respect to t can be trivially written as

$$\frac{dm(t)}{dt} = \rho w L, \tag{5.18}$$

where ρ is the density of the material (cf., Table 2.1), w is the out-of-plane width of the structural member (defined to be a constant of 20 mm as in Figure 3.16), and L is the length of the structural member. The sensitivity of the stiffness K of the structural topology is given as

$$\frac{dK(t)}{dt} = \frac{d}{dt}\left(\frac{f_y}{\delta_y(t)}\right) = -\frac{d\delta_y}{dt}\frac{f_y}{{\delta_y}^2}, \tag{5.19}$$

where f_y is a constant applied force of $-0.05\,\mathrm{N}$, δ_y is the resulting tip displacement solved for using Equation 5.1, and $\frac{d\delta_y}{dt}$ is obtained by solving Equation 5.9. For this problem, the ε-constraint is applied to the mass objective with $N_\varepsilon = 15$, and the single set of objective function values for each individual topology returned to the genetic algorithm (cf., Equation 5.17) are defined by calculating the last value of $\varepsilon^{(j)}$ for which there is a significant (i.e., $\geq$1%) increase in stiffness. For a direct comparison to the results obtained in Section 3.6.4.1, the thickness of each structural member is bounded such that $0\,\mathrm{mm} \leq t \leq 2\,\mathrm{mm}$.

The hybrid optimization framework discussed in Section 5.1.3 is employed with a specified population size of 100 for 1,000 generations and three parameterized L-System recursions. The resulting Pareto frontier is shown in Figure 5.5, along with the frontiers associated with the Michell truss and the optimization in Section 3.6.4.1 which considered SPIDRS-generated topologies with a constant member thickness and four parameterized L-System recursions. It is noticeable that designs generated by the SPIDRS algorithm with in situ sizing optimization (i.e., using the hybrid optimization framework) converge to the frontier generated by discrete Michell truss topologies with a constant member thickness of 2 mm. Several topologies of interest generated using the hybrid topology/sizing optimization approach are shown in Figure 5.6. Here, the effect of sizing optimization can clearly be seen, as structural members that form auxiliary load paths or partially serve as "construction lines" to facilitate the creation of critical load paths are removed, resulting

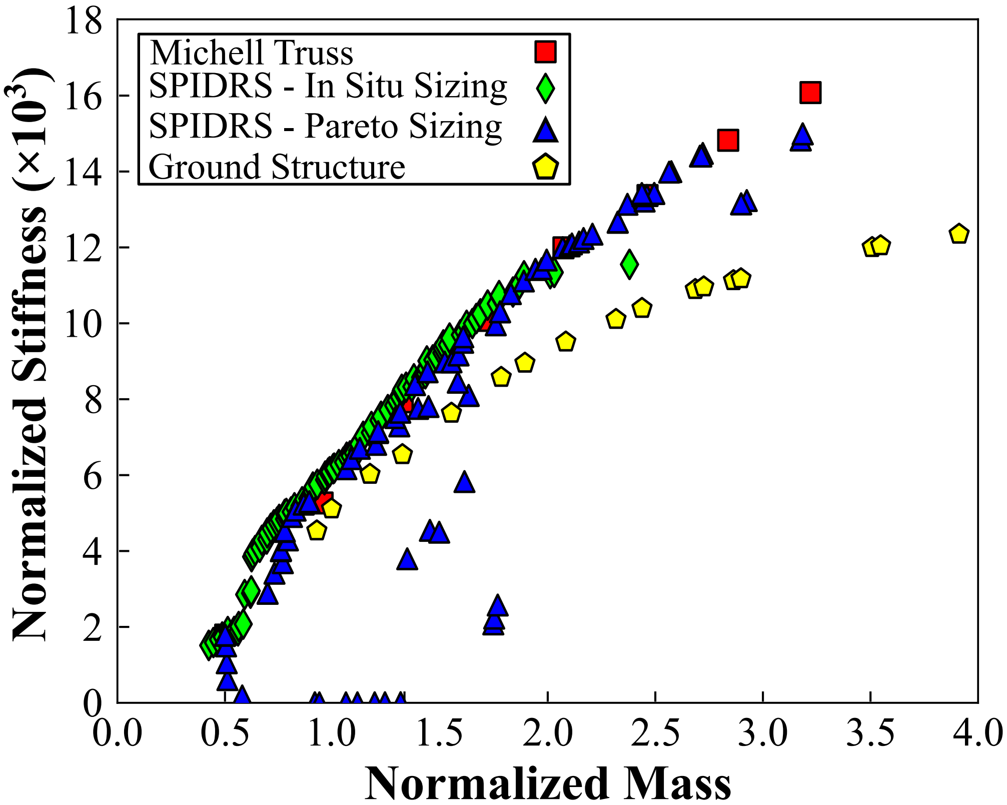
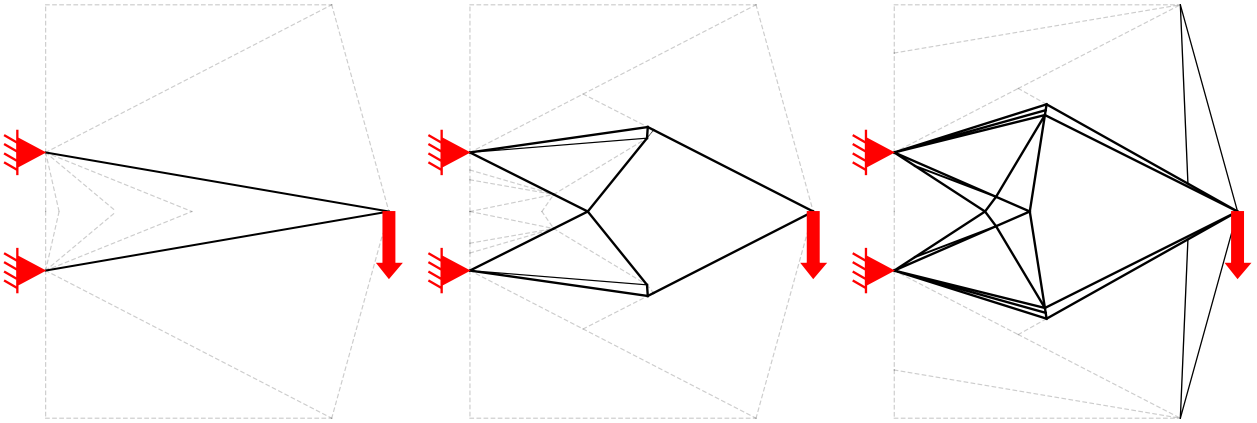


Figure 5.5: Pareto frontier of the cantilevered frame design problem generated using the hybrid topology/sizing optimization framework with thickness bounds of $0\,\text{mm} \leq t \leq 2\,\text{mm}$, along with comparisons to the frontiers associated with the Michell truss (constant thickness), SPIDRS topology optimization (constant thickness), SPIDRS topology optimization with sizing optimization applied to the final Pareto optimal solutions, and GSM.



(a) Lightest design (normalized mass of 0.42, normalized stiffness of 1511.6).

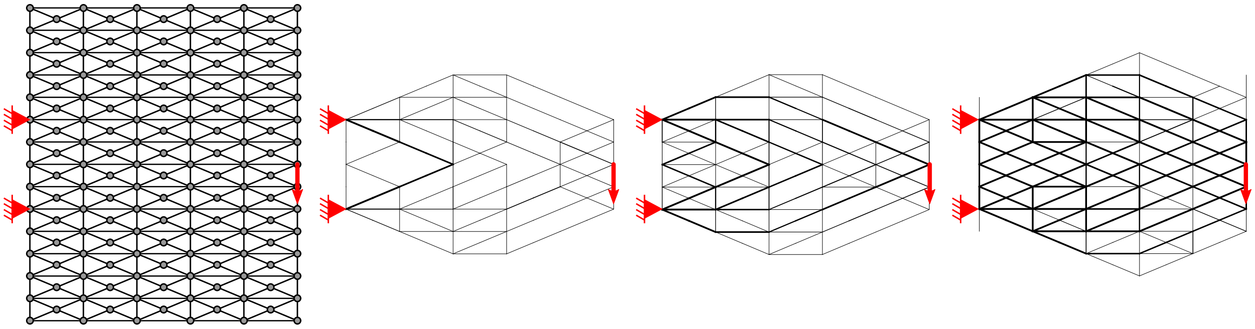
(b) Design closest to the utopia point (normalized mass of 1.32, normalized stiffness of 8257.2).

(c) Stiffest design (normalized mass of 2.38, normalized stiffness of 11546.3).

Figure 5.6: Several 2-D cantilevered frame topologies of interest generated by SPIDRS using a hybrid optimization framework. Structural members that do not contribute to the stiffness of the structure (represented by thin grey lines) are removed, resulting in structural topologies that are capable of matching known optimal solutions.

in Michell-like topologies that are capable of matching known optimal solutions.

Also shown in Figure 5.5 is a frontier generated by subjecting the predefined GSM layout shown in Figure 5.7a to the sizing optimization scheme detailed in Chapter 5.4. This layout was chosen based on a parameterized study that observed how changing the number of nodes in both the x- and y-direction affected the performance of resulting structural topologies. Several resulting topologies are provided in Figure 5.7b-d. Given that all potential structural members are predefined, it is unsurprising that sizing optimization results in a single structural topology with increases in allowable mass being used to increase the thickness of critical members. Comparing the GSM frontier to that obtained by applying the sizing optimization framework to SPIDRS-generated topologies, it is evident that the GSM performs well at lower normalized mass values; however, with increased mass the improvement in structural stiffness begins to degrade. This is once again intuitive, as, given a single topology and a structural member thickness constraint, increases in mass are eventually used to thicken less critical members that have a smaller effect on the stiffness of the topology. Overall, while the GSM performs admirably, this result highlights the benefits of



(a) Predefined structural member layout.

(b) Lightest design (normalized mass of 0.93, normalized stiffness of 4529.6).

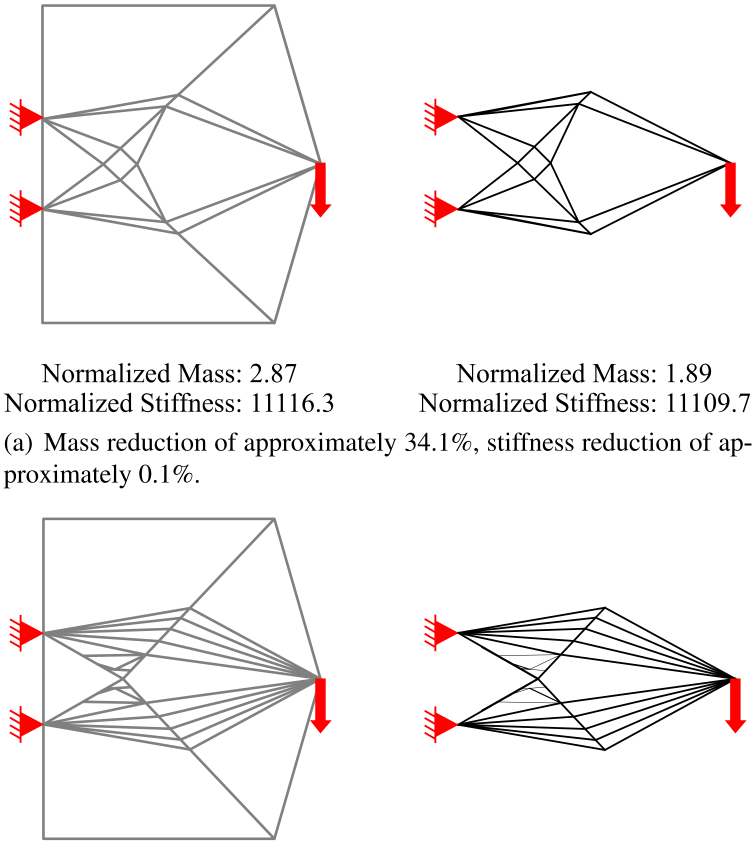
(c) Design closest to the utopia point (normalized mass of 1.55, normalized stiffness of 7628.3).

(d) Stiffest design (normalized mass of 3.91, normalized stiffness of 12342.4).

Figure 5.7: The predefined GSM structural member layout (a) for the associated frontier in Figure 5.5 and several generated topologies of interest (b-d). While designs with a lower normalized mass compare favorably with SPIDRS-generated solutions, as mass increases the improvements in stiffness begin to degrade.

using SPIDRS to generate diverse, irregular ground structures.

Figure 5.5 also demonstrates the effect of where in the topology optimization process sizing optimization is implemented. Specifically, rather than performing sizing optimization in situ as previously discussed, consider the application of the sizing optimization scheme on the final Pareto frontier generated using only topology optimization as presented in Section 3.6.4.1. A comparison between the unsized and sized topologies for both the design closest to the utopia point and stiffest design on this frontier is shown in Figure 5.8. Performing sizing optimization on these Pareto optimal topologies results in a reduction of mass of approximately 34.1% and 24.2% with decreases in stiffness of 0.1% and 3.1%. This reduction in mass comes primarily from eliminating the structural members that make up the initial graph and form an auxiliary load path that provides negligible stiffness when compared to the rest of the structure. Designs that use part or all of the initial graph in their critical load paths do not benefit from the same magnitude of mass decrease, and thus the resulting set of solutions do not form a true Pareto frontier (cf., Figure 5.5). However, the performances of the majority of these sized designs compare favorably with those of SPIDRS-generated designs that consider both topology and sizing optimization simultaneously, which indicates that



Normalized Mass: 2.87
Normalized Stiffness: 11116.3

Normalized Mass: 1.89
Normalized Stiffness: 11109.7

(a) Mass reduction of approximately 34.1%, stiffness reduction of approximately 0.1%.

Normalized Mass: 4.20
Normalized Stiffness: 15463.7

Normalized Mass: 3.18
Normalized Stiffness: 14983.9

(b) Mass reduction of approximately 24.2%, stiffness reduction of approximately 3.1%.

Figure 5.8: Examples of how employing sizing optimization on Pareto optimal designs from the topology optimization problem in Section 3.6.4.1 can considerably reduce the mass of the structure while maintaining structural performance.

implementing the sizing optimization scheme within the topology optimization framework may not have as much of an effect on the evolution of the frontier as originally predicted. Considering computation time (cf., Table 5.1), analyzing 100-individual populations for 1,000 generations using hybrid topology/sizing optimization required approximately 236 hours, while analyzing the same problem using only topology optimization, then applying sizing optimization to designs on the final Pareto frontier required approximately 27 hours. It should be noted that, unlike the topology optimization problem in Section 3.6.4.1, the hybrid optimization framework does not leverage

Table 5.1: Comparison of computation times necessary for both the 2-D hybrid optimization framework and the 2-D topology optimization framework with Pareto optimal designs sized during post-processing.

	Computation Time (hr)
SPIDRS - In Situ Sizing (Serialized)	236
SPIDRS - Pareto Sizing (Parallelized)	27

any computational parallelization techniques, which could dramatically accelerate the optimization process. Nevertheless, the 89% decrease in computation time indicates that applying sizing optimization only to the final topology optimization Pareto frontier should be recognized as a computationally efficient alternative to the hybrid optimization approach.

One of the drawbacks of the SIMP method noted in Chapter 3 was its relatively poor performance relative to the frontiers associated with both the Michell truss and SPIDRS algorithm at lower normalized mass values, caused in part by a difficulty in creating load paths with a limited amount of material. To provide a better comparison at higher normalized masses, an identical hybrid topology/sizing optimization is considered with the structural member thickness constraint modified such that $0\,\text{mm} \leq t \leq 4\,\text{mm}$. The same thickness constraint and sizing optimization scheme is also applied to discrete Michell truss topologies such that the frontier associated with these topologies can be represented as a continuous line. The resulting Pareto frontiers are shown in Figure 5.9, along with a comparison to a frontier generated using a SIMP implementation [18] assuming a discretization of the design domain into unit elements (i.e., each element measures $1\,\text{mm} \times 1\,\text{mm}$ such that the problem is defined by 105,000 design variables). Once again, the performances of SPIDRS-generated topologies show good agreement with those associated with the Michell truss and outperform those of SIMP-generated designs across a range of normalized mass values. Note that the lack of topological diversity observed in the SPIDRS-generated designs are attributed to the application of the topological equivalency constraint discussed in Appendix D.

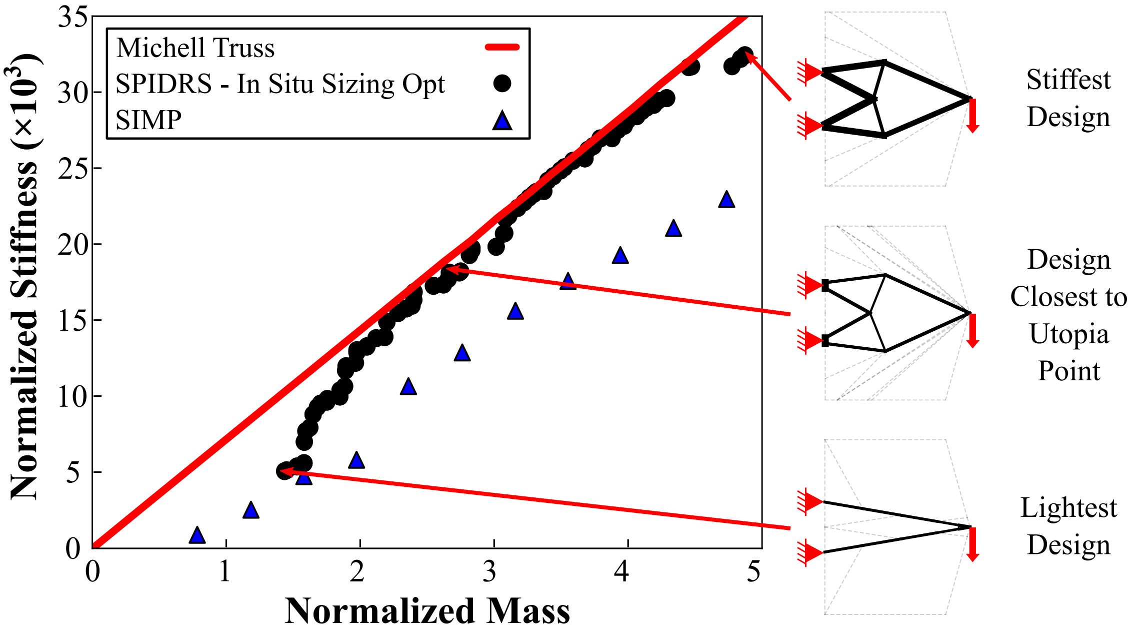


Figure 5.9: Pareto frontier of the cantilevered frame design problem generated using the hybrid topology/sizing optimization framework with a thickness bounds of $0\,\text{mm} \leq t \leq 4\,\text{mm}$, along with comparisons to the frontiers associated with the Michell truss and a SIMP implementation.

5.1.4.2 Compliant Mechanism: Tensile Inverter

The hybrid optimization framework discussed in Section 5.1.3 can also be applied to the tensile inverter problem considered in Section 3.6.4.2. The dimensions and boundary conditions (cf., Figure 3.19), initial graph, and NSGA-II parameters remain consistent with the previous design study. As before, the goal of the problem is to maximize both the displacement and force inversion behavior of the topology; however, for this sizing optimization study only a single material, VeroWhite, is considered (cf., Table 2.1). Given that one of the objective functions for this problem is a function of the internal reaction forces in structural members, it is necessary to calculate both these reaction forces and their sensitivities with respect to the thicknesses of these members using the 2-D frame FEA theory detailed in Section 5.1.1. After one has solved for the global displacement vector $\mathbf{u}$ in Equation 5.1, the internal reaction force vector in the global coordinate system $\mathbf{f}$ can be calculated as

$$\{\mathbf{f}\} = [\mathbf{K}]\{\mathbf{u}\} - \{\mathbf{F}\}. \tag{5.20}$$

Assuming that the force vector is independent of member thickness, the sensitivity of this internal reaction force vector with respect to a given member thickness t is written as

$$\left\{\frac{d\mathbf{f}(t)}{dt}\right\} = \left[\frac{d\mathbf{K}(t)}{dt}\right]\{\mathbf{u}(t)\} + [\mathbf{K}(t)]\left\{\frac{d\mathbf{u}(t)}{dt}\right\}, \tag{5.21}$$

where the derivative terms for the global stiffness matrix and global displacement vector are calculated using Equations 5.11 and 5.9, respectively. It should be noted that these FEA equations assume a linear structure response and thus do not capture the effects of local buckling and large displacements/rotations that were considered and found to govern the responses of compliant mechanisms in previous studies. Furthermore, work by Bendsøe and Sigmund has demonstrated the effect that considering nonlinear structural responses has on the solutions to compliant mechanism problems [2]. Thus, this study should only be viewed as an academic exercise meant to further demonstrate the utility of applying the proposed hybrid optimization framework to SPIDRS-generated topologies.

Given the displacement and force inversion ratios defined in Equation 2.1, the sensitivities of these objective functions with respect to a given structural member thickness t can be written as

$$\frac{dr_\delta(t)}{dt} = \frac{d}{dt}\frac{\delta_{x,out}(t)}{\delta_{x,in}} = \frac{d\delta_{x,out}(t)}{dt}\frac{1}{\delta_{x,in}}, \tag{5.22}$$

and

$$\frac{dr_f(t)}{dt} = \frac{d}{dt}\left(\frac{-f_{x,out}(t)}{f_{x,in}(t)}\right) = -\frac{df_{x,out}(t)}{dt}\frac{1}{f_{x,in}} + \frac{df_{x,in}}{dt}\frac{f_{x,out}}{f_{x,in}^2}, \tag{5.23}$$

respectively, where $\delta_{x,in}$ is a constant input displacement of -1 mm, $\delta_{x,out}$ is the resulting output displacement solved for using Equation 5.1, $\frac{d\delta_{x,out}(t)}{dt}$ is obtained by solving Equation 5.9, $f_{x,in}$ and $f_{x,out}$ are the resulting input and output reaction forces determined using Equation 5.20, and $\frac{df_{x,in}(t)}{dt}$ and $\frac{df_{x,out}(t)}{dt}$ are calculated using Equation 5.21. For this problem, the ε-constraint is applied to

$r_f(t)$ with $N_\varepsilon = 10$, and the single set of objective function values returned to the genetic algorithm for each individual topology (cf., Equation 5.17) are defined as the point along the ε-constraint-generated Pareto frontier closest to the utopia point. The thickness of each structural member is bounded such that $0\,\text{mm} \leq t \leq 4\,\text{mm}$. Additionally, to ensure that any generated solutions remain relatively plausible, buckling and stress constraints of the form

$$f_x > -\frac{\pi^2 EI}{L^2} \quad \text{and} \quad \sigma_x \leq \sigma_{crit} \tag{5.24}$$

are applied to each structural member, where f_x is the axial internal reaction force in the element coordinate system, $-\pi^2 EI/L^2$ is the Euler critical load for first mode buckling [132], σ_x is the axial stress in the element coordinate system, and $\sigma_{crit} = 33.8\,\text{MPa}$.

Applying the hybrid optimization framework developed in Section 5.1.3 with a specified population size of 40 for 100 generations and two L-System recursions results in the Pareto frontier shown in Figure 5.10. It is immediately noticeable that this frontier agrees well with the derived ideal solution for the tensile inverter problem given in Equation 2.3. Also shown in Figure 5.10 are several topologies of interest generated by the hybrid optimization framework, from which one can observe several notable qualities. First, each design features an identical load path geometry to those previously observed in SPIDRS-generated tensile inverters in this work. Furthermore, the load path geometries associated with these solutions follow similar trends to those observed in Figure 3.28, with high displacement inversion solutions featuring rotation points closer to node 3 of the initial graph and high force inversion solutions featuring rotation points closer to node 1. Second, the effect of the sizing optimization scheme is perhaps more noticeable in this design problem than in the cantilevered truss problems of Section 5.1.4.1, as each mechanism features structural members of varying thickness rather than the binary "min/max" thicknesses prevalent in the frame topologies. This is due both to the fact that in the cantilevered frame problems member thickness is more directly related to one of the objectives (minimizing mass) and the presence of the buckling and stress constraints (cf., Equation 5.24) in the tensile inverter, which dictate that any structural members experiencing a nontrivial magnitude of axial reaction force (either tensile or compressive) must have the cross-sectional area, and by extension thickness, to withstand that load.

Finally, consider the SPIDRS-generated design closest to demonstrating displacement and force inversion ratios equal to one, the deformed displacement inversion configuration of which is illustrated in Figure 5.11. In this design, the structural member that connects the input point to the point of relative rotation in the load path (denoted by the dashed red line) is loaded in tension and has an optimized thickness of $t = 0.13\,\text{mm}$. Thus, it stands to reason that this structural member could be replaced by a string with a comparable tensile modulus. While intuitive, as that member only experiences tensile forces in both load cases, this result would not be possible to achieve using a SIMP implementation unless the design domain was very finely discretized (e.g., on the order of 1E6 total elements). The structural member that extends from the point of relative rotation to the output point, however, has an optimized thickness of $t = 2.25\,\text{mm}$ due to the fact

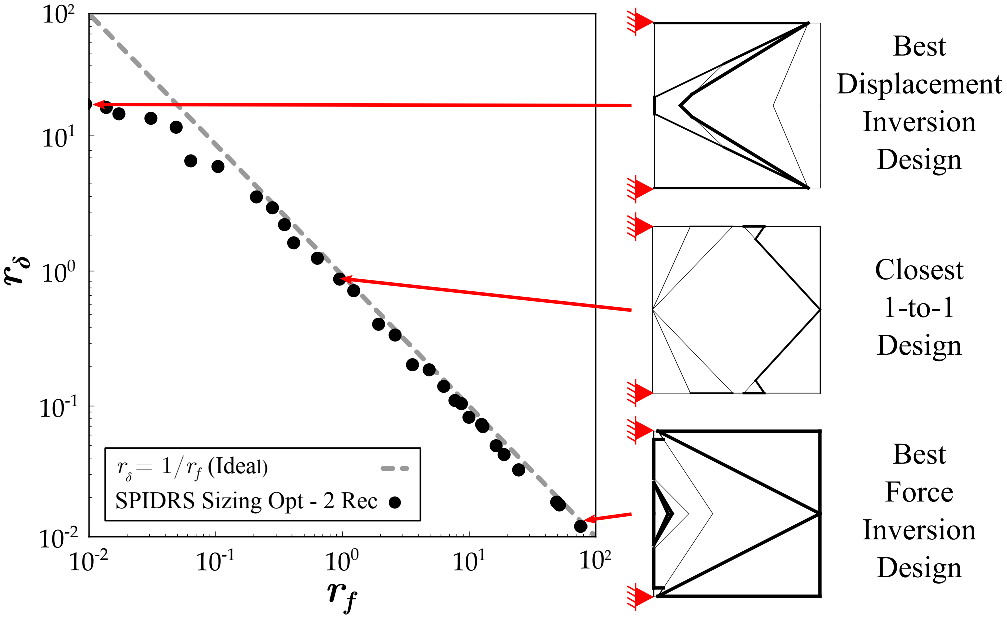


Figure 5.10: Pareto frontier associated with the tensile inverter design problem generated by applying the hybrid optimization framework to SPIDRS-generated topologies. This frontier shows good agreement with the derived ideal solution in Equation 2.3.

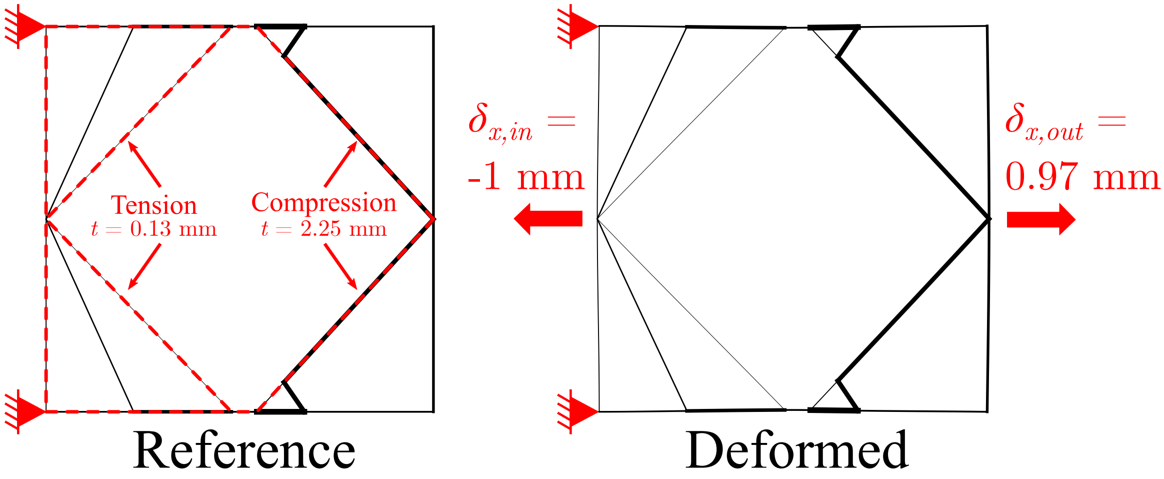


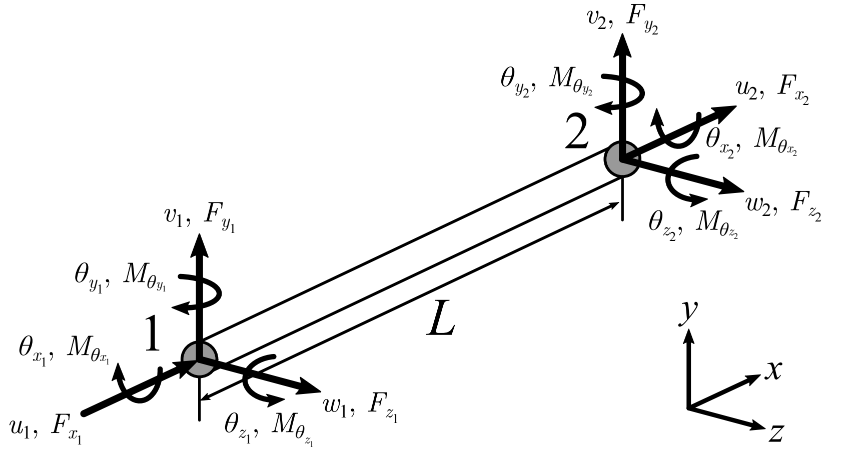
Figure 5.11: Comparison of the reference and deformed configuration of the SPIDRS-generated topologies that most closely demonstrates a displacement and force inversion ratios equal to one. This design clearly demonstrates the effect sizing optimization has on solutions, as some structural members in tension can be represented similarly to strings, while members in compression are thickened to satisfy buckling constraints.

that it is loaded in compression in both load cases and thus must be able to satisfy the buckling constraint in Equation 5.24.

5.2 Three-Dimensional Sizing Optimization

5.2.1 3-D Frame FEA and Sensitivity Analysis

The hybrid optimization framework detailed in Section 5.1.3 is also applicable to 3-D SPIDRS-generated topologies. However, this requires adapting the frame FEA solver in Section 5.1.1 to consider elements with six degrees of freedom (three translational, three rotational) as illustrated in Figure 5.12. The fundamental system of equations for the finite element model remains the same as that stated in Equation 5.1. For a single frame element with six degrees of freedom, the element



v_2, F_{y_2}
θ_{y_2}, $M_{\theta_{y_2}}$
u_2, F_{x_2}
2
θ_{x_2}, $M_{\theta_{x_2}}$
w_2, F_{z_2}
θ_{z_2}, $M_{\theta_{z_2}}$
v_1, F_{y_1}
θ_{y_1}, $M_{\theta_{y_1}}$
L
y
1
θ_{x_1}, $M_{\theta_{x_1}}$
x
w_1, F_{z_1}
u_1, F_{x_1}
θ_{z_1}, $M_{\theta_{z_1}}$
z

Figure 5.12: Example of a 3-D frame element with six degrees of freedom (threes translational, three rotational).

stiffness matrix is redefined as

$$[\mathbf{K^e}] = \begin{bmatrix}
\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
0 & 0 & \frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 & 0 & 0 & -\frac{12EI_y}{L^3} & 0 & -\frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
-\frac{AE}{L} & 0 & 0 & 0 & 0 & 0 & \frac{AE}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & -\frac{6EI_z}{L^2} \\
0 & 0 & -\frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\
0 & 0 & 0 & -\frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & -\frac{6EI_z}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
\end{bmatrix} \qquad (5.25)$$

where A is the cross-sectional area, E is the elastic modulus, L is the length associated with the element, I_y and I_z are the area moments of inertia with respect to the y- and z-axis, G is the shear modulus, and J is the polar moment of inertia. The adapted element displacement and force vectors are then given as

$$\{\mathbf{u^e}\} = \left\{ \begin{array}{cccccccccccc} u_1 & v_1 & w_1 & \theta_{x_1} & \theta_{y_1} & \theta_{z_1} & u_2 & v_2 & w_2 & \theta_{x_2} & \theta_{y_2} & \theta_{z_2} \end{array} \right\}^T \tag{5.26}$$

and

$$\{\mathbf{F^e}\} = \left\{ \begin{array}{cccccccccccc} F_{x_1} & F_{y_1} & F_{z_1} & M_{\theta_{x_1}} & M_{\theta_{y_1}} & M_{\theta_{z_1}} & F_{x_2} & F_{y_2} & F_{z_2} & M_{\theta_{x_2}} & M_{\theta_{y_2}} & M_{\theta_{z_2}} \end{array} \right\}^T, \tag{5.27}$$

respectively. Converting these element matrices/vectors into the global coordinate system requires a redefined transformation matrix of the form

$$[\mathbf{T}] = \begin{bmatrix} \mathbf{T_3} & 0 & 0 & 0 \\ 0 & \mathbf{T_3} & 0 & 0 \\ 0 & 0 & \mathbf{T_3} & 0 \\ 0 & 0 & 0 & \mathbf{T_3} \end{bmatrix} \tag{5.28}$$

in which

$$[\mathbf{T_3}] = \begin{bmatrix} \cos(x,X) & \cos(x,Y) & \cos(x,Z) \\ \cos(y,X) & \cos(y,Y) & \cos(y,Z) \\ \cos(z,X) & \cos(z,Y) & \cos(z,Z) \end{bmatrix}, \tag{5.29}$$

where each of these terms represents a direction cosine between the specified axis that defines the element coordinate system (e.g., x, y, or z) and the specified axis that defines the global coordinate system (e.g., X, Y, or Z). Applying Equation 5.28 to Equations 5.25 and 5.27 results in an equation consistent with Equation 5.6. As in 2-D, the system of equations for a structure consisting of multiple frame elements is formed by assembling Equation 5.6 into a single linear system of equations where common nodes share degrees of freedom. This system of equations is then solved to determine the global displacement vector, $\mathbf{u}$.

Regarding sensitivity analysis, the process in 3-D is identical to that laid out in Section 5.1.2, where the only unknown term that must be calculated is $\left[\frac{d\mathbf{K}(t)}{dt}\right]$. For simplicity, herein it is assumed that out-of-plane width of a structural element is equal to its thickness such that $A = t^2$, $I_y = I_z = t^4/12$, and $J = t^4/6$. Thus, Equation 5.25 can be rewritten as

$$[\mathbf{K^e}] = \begin{bmatrix} \frac{Et^2}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{Et^2}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{Et^4}{L^3} & 0 & 0 & 0 & \frac{Et^4}{2L^2} & 0 & -\frac{Et^4}{L^3} & 0 & 0 & 0 & -\frac{Et^4}{2L^2} \\ 0 & 0 & \frac{Et^4}{L^3} & 0 & -\frac{Et^4}{2L^2} & 0 & 0 & 0 & -\frac{Et^4}{L^3} & 0 & -\frac{Et^4}{2L^2} & 0 \\ 0 & 0 & 0 & \frac{Gt^4}{6L} & 0 & 0 & 0 & 0 & 0 & -\frac{Gt^4}{6L} & 0 & 0 \\ 0 & 0 & -\frac{Et^4}{2L^2} & 0 & \frac{Et^4}{3L} & 0 & 0 & 0 & \frac{Et^4}{2L^2} & 0 & \frac{Et^4}{6L} & 0 \\ 0 & \frac{Et^4}{2L^2} & 0 & 0 & 0 & \frac{Et^4}{3L} & 0 & -\frac{Et^4}{2L^2} & 0 & 0 & 0 & \frac{Et^4}{6L} \\ -\frac{Et^2}{L} & 0 & 0 & 0 & 0 & 0 & \frac{Et^2}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{Et^4}{L^3} & 0 & 0 & 0 & -\frac{Et^4}{2L^2} & 0 & \frac{Et^4}{L^3} & 0 & 0 & 0 & -\frac{Et^4}{2L^2} \\ 0 & 0 & -\frac{Et^4}{L^3} & 0 & \frac{Et^4}{2L^2} & 0 & 0 & 0 & \frac{Et^4}{L^3} & 0 & \frac{Et^4}{2L^2} & 0 \\ 0 & 0 & 0 & -\frac{Gt^4}{6L} & 0 & 0 & 0 & 0 & 0 & \frac{Gt^4}{6L} & 0 & 0 \\ 0 & 0 & -\frac{Et^4}{2L^2} & 0 & \frac{Et^4}{6L} & 0 & 0 & 0 & \frac{Et^4}{2L^2} & 0 & \frac{Et^4}{3L} & 0 \\ 0 & -\frac{Et^4}{2L^2} & 0 & 0 & 0 & \frac{Et^4}{6L} & 0 & -\frac{Et^4}{2L^2} & 0 & 0 & 0 & \frac{Et^4}{3L} \end{bmatrix}. \tag{5.30}$$

Taking the derivative of Equation 5.30 with respect to t, one arrives at

$$\left[\frac{d\mathbf{K^e}}{dt}\right] = \begin{bmatrix} \frac{2Et}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{2Et}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4Et^3}{L^3} & 0 & 0 & 0 & \frac{2Et^3}{L^2} & 0 & -\frac{4Et^3}{L^3} & 0 & 0 & 0 & -\frac{2Et^3}{L^2} \\ 0 & 0 & \frac{4Et^3}{L^3} & 0 & -\frac{2Et^3}{L^2} & 0 & 0 & 0 & -\frac{4Et^3}{L^3} & 0 & -\frac{2Et^3}{L^2} & 0 \\ 0 & 0 & 0 & \frac{2Gt^3}{3L} & 0 & 0 & 0 & 0 & 0 & -\frac{2Gt^3}{3L} & 0 & 0 \\ 0 & 0 & -\frac{2Et^3}{L^2} & 0 & \frac{4Et^3}{3L} & 0 & 0 & 0 & \frac{2Et^3}{L^2} & 0 & \frac{2Et^3}{3L} & 0 \\ 0 & \frac{2Et^3}{L^2} & 0 & 0 & 0 & \frac{4Et^3}{3L} & 0 & -\frac{2Et^3}{L^2} & 0 & 0 & 0 & \frac{2Et^3}{3L} \\ -\frac{2Et}{L} & 0 & 0 & 0 & 0 & 0 & \frac{2Et}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{4Et^3}{L^3} & 0 & 0 & 0 & -\frac{2Et^3}{L^2} & 0 & \frac{4Et^3}{L^3} & 0 & 0 & 0 & -\frac{2Et^3}{L^2} \\ 0 & 0 & -\frac{4Et^3}{L^3} & 0 & \frac{2Et^3}{L^2} & 0 & 0 & 0 & \frac{4Et^3}{L^3} & 0 & \frac{2Et^3}{L^2} & 0 \\ 0 & 0 & 0 & -\frac{2Gt^3}{3L} & 0 & 0 & 0 & 0 & 0 & \frac{2Gt^3}{3L} & 0 & 0 \\ 0 & 0 & -\frac{2Et^3}{L^2} & 0 & \frac{2Et^3}{3L} & 0 & 0 & 0 & \frac{2Et^3}{L^2} & 0 & \frac{4Et^3}{3L} & 0 \\ 0 & -\frac{2Et^3}{L^2} & 0 & 0 & 0 & \frac{2Et^3}{3L} & 0 & -\frac{2Et^3}{L^2} & 0 & 0 & 0 & \frac{4Et^3}{3L} \end{bmatrix}. \tag{5.31}$$

5.2.2 Design Examples: Cantilevered Frame

To demonstrate the effectiveness of applying a hybrid topology/sizing optimization framework to 3-D SPIDRS-generated topologies, the design problem of a 3-D cantilevered frame subjected to transverse loading (cf., Section 4.2.1) is revisited. The boundary conditions and initial graph are identical to those shown in Figure 4.10 and the overall topology optimization problem is consistent with that given in Table 4.1. The sensitivities of the objective functions mass and stiffness are equivalent to Equations 5.18-5.19. As in Section 5.1.4.1, the ε-constraint is applied to the mass objective with $N_\varepsilon = 15$, and the single set of objective function values returned to the genetic algorithm for each individual structural topology (cf., Equation 5.17) are defined by calculating the last value of $\varepsilon^{(j)}$ for which there is a significant (i.e., $\geq 1\%$) increase in stiffness. For a direct comparison to the results obtained in Section 4.2.1.1, the thickness (and as mentioned in Section 5.2.1 out-of-plane width) for each structural member is bounded such that $0\,\text{mm} \leq t \leq 1\,\text{mm}$.

The hybrid optimization framework is employed with a specified population size of 100 for 1,000 generations and two L-System recursions. The resulting Pareto frontier is shown in Figure 5.13 along with a comparison to the frontiers associated with both the optimization discussed in Section 4.2.1.1 and a SIMP implementation with a structural design domain discretization of 576,000 elements. It is noticeable that the hybrid optimization frontier significantly outperforms SIMP-generated designs despite the extremely fine discretization of the design domain. Additionally, even with a large number of elements, SIMP is unable to truly converge on a solution at normalized mass below approximately 1.3[1]. Several resulting topologies from the frontier generated by the hybrid optimization approach are shown in Figure 5.14. Interestingly, topologies with low normalized mass values (cf., Figure 5.14a) have actually been reduced down to 2-D structures by the sizing optimization scheme. In hindsight, this is an intuitive result that stems from the symmetry boundary condition that shares a plane with the applied transverse load (cf., Figure 4.10) and the fact that generating structural members on the face associated with this boundary condition is the easiest way to construct critical load paths. However, as the normalized mass of designs along this frontier increases, more 3-D structural features are utilized as observed in Figures 5.14b-c.

As in Section 5.1.4.1, Figure 5.13 also features a set of solutions generated by applying the

[1] The design shown with a normalized mass of approximately 0.65 does not feature a load path, and its normalized stiffness value is believed to represent some numerical artifact from structural analysis.

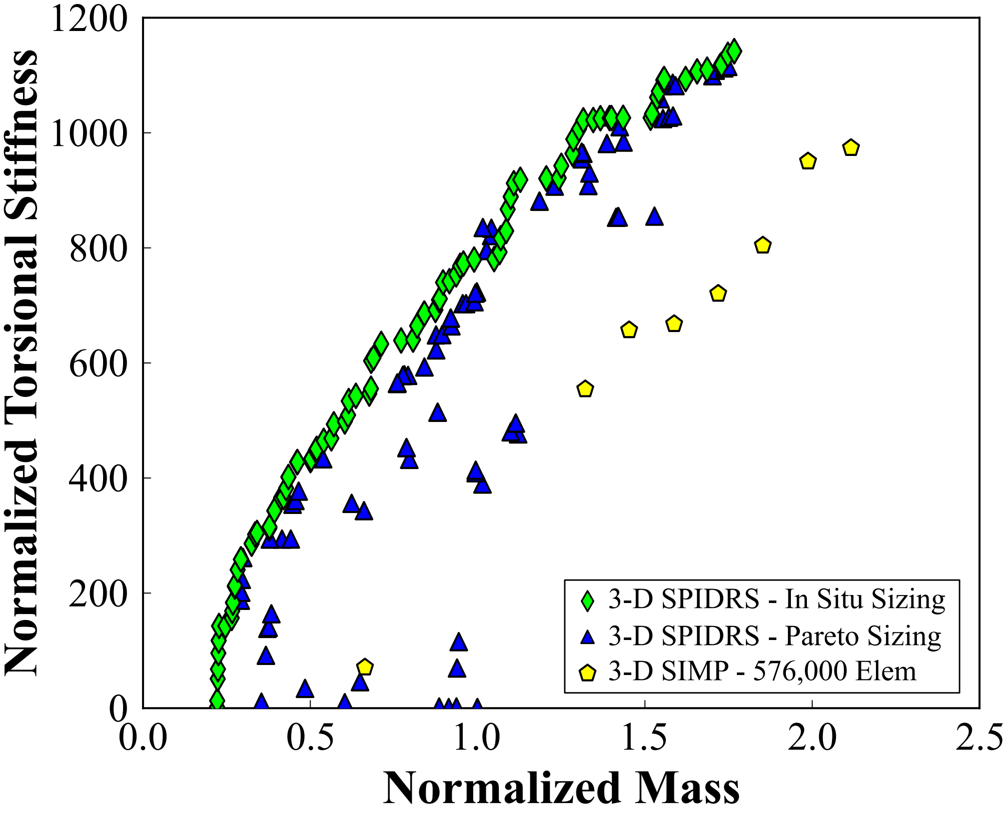
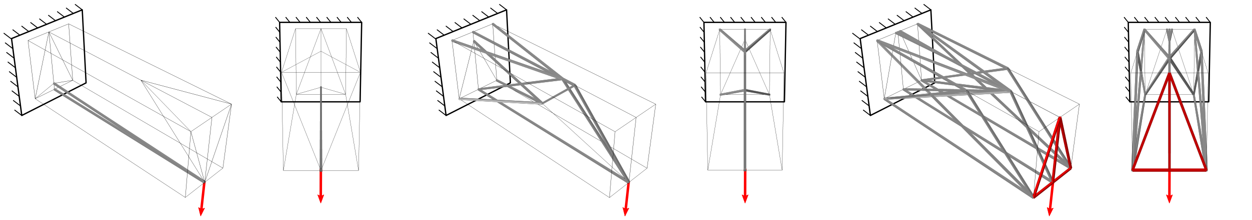


Figure 5.13: Pareto frontier of the design of a 3-D cantilevered frame subjected to transverse loading generated using the hybrid topology/sizing optimization framework with member thickness bounds of $0\,\mathrm{mm} \leq t \leq 1\,\mathrm{mm}$, along with comparisons to the frontier associated with SPIDRS topology optimization (constant thickness), SPIDRS topology optimization with sizing optimization applied to the final Pareto optimal solutions, and a SIMP implementation with 576,000 elements.

sizing optimization scheme on the final Pareto frontier generated using only topology optimization (cf., Section 4.2.1.1), with the goal of illustrating the effect of where in the topology optimization process sizing optimization is implemented. A comparison between the unsized and sized topologies for both the design closest to the utopia point and stiffest design on this frontier in shown in Figure 5.15. Performing sizing optimization on these Pareto optimal topologies results in a reduction of mass of approximately 39.1% and 35.3% with decreases in stiffness of 0.6% and 3.3%, respectively. As before, this reduction in mass comes primarily from eliminating structural members from the initial graph that form auxiliary load paths and provide negligible added stiff-



(a) Lightest design (normalized mass of 0.22, normalized stiffness of 1.99).

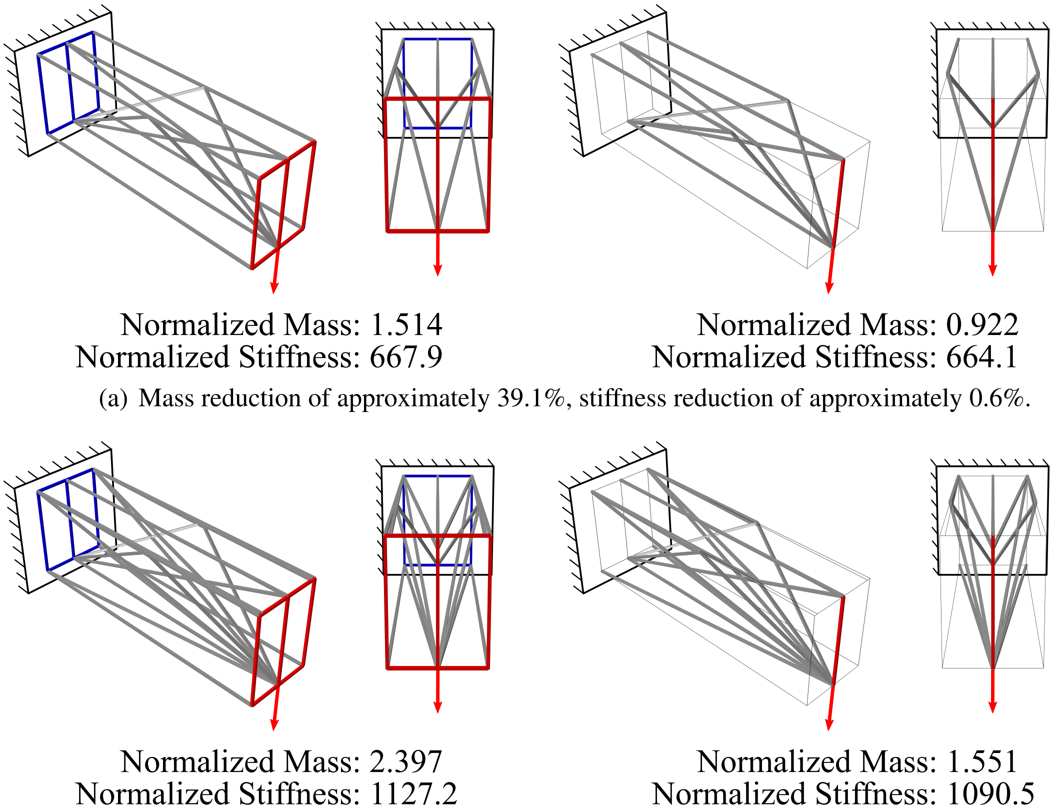
(b) Design closest to the utopia point (normalized mass of 0.71, normalized stiffness of 632.80).

(c) Stiffest design (normalized mass of 1.77, normalized stiffness of 1141.21).

Figure 5.14: Several 3-D cantilevered frame topologies of interest generated by SPIDRS using a hybrid optimization approach, where members removed from the structure are represented by thin lines. At low normalized mass values, topologies are reduced to 2-D structures, while topologies with higher normalized mass values feature more 3-D structural features.

ness compared to the rest of the structure. From Figure 5.13, it is noticeable that a majority of the designs subjected to sizing after topology optimization compare well with the frontier generated using in situ sizing optimization in terms of performance. Considering computation time (cf., Table 5.2), analyzing 100-individual populations for 1,000 generations using the hybrid topology/sizing optimization approach required approximately 430 hours, while analyzing the same problem using only topology optimization, then applying sizing optimization to designs along the final Pareto frontier required approximately 25 hours. Once again, it is crucial to note that the hybrid optimization framework does not take advantage of any computational parallelization techniques, which could dramatically reduce computation time. However, the roughly 94% reduction in computation time once again indicates that applying sizing optimization only to the final topology optimization Pareto frontier could be implemented as a computationally efficient alternative to the hybrid optimization approach.

Given the demonstrated effect of symmetry boundary conditions on resulting designs generated using a hybrid topology/sizing optimization approach, an additional 3-D cantilevered frame design problem is considered as illustrated in Figure 5.16. The 3-D structural design domain is defined to have a length of 120 mm, a height of 40 mm, and a width of 30 mm. The initial state of the graph


Normalized Mass: 1.514
Normalized Stiffness: 667.9
Normalized Mass: 0.922
Normalized Stiffness: 664.1
Normalized Mass: 2.397
Normalized Stiffness: 1127.2
Normalized Mass: 1.551
Normalized Stiffness: 1090.5

(a) Mass reduction of approximately 39.1%, stiffness reduction of approximately 0.6%.

(b) Mass reduction of approximately 35.3%, stiffness reduction of approximately 3.3%.

Figure 5.15: Examples of how employing sizing optimization on Pareto optimal designs from the topology optimization problem in Section 4.2.1.1 can considerably reduce the mass of structure while maintaining structural performance.

Table 5.2: Comparison of computation times necessary for both the 3-D hybrid optimization framework and the 3-D topology optimization framework with Pareto optimal designs sized during post-processing.

	Computation Time (hr)
3-D SPIDRS - In Situ Sizing (Serialized)	430
3-D SPIDRS - Pareto Sizing (Parallelized)	25

is defined as

$$\{[1,2,3,4],[6,5,8,7],[1,4,8,5],[3,2,9,6,7],[1,5,6,9,2],[4,3,7,8]\}\,.$$

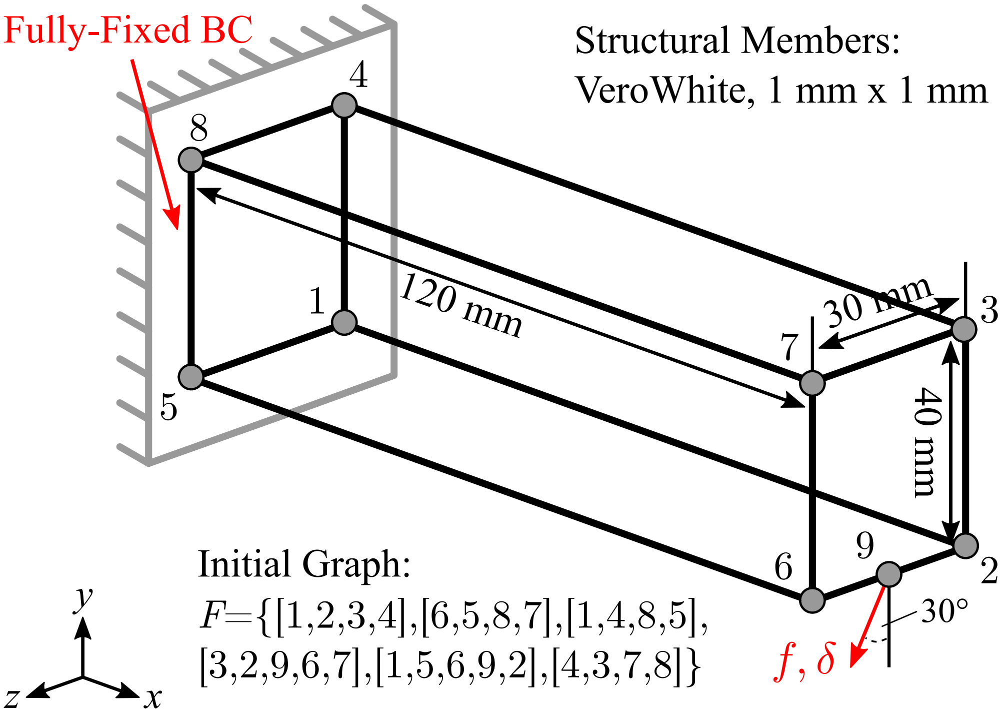


Figure 5.16: Initial graph and boundary conditions associated with the 3-D cantilevered frame subjected to off-axis loading.

A fully-fixed boundary condition (i.e., $u_x = u_y = u_z = 0$) is placed on all nodes in the plane associated with the face $[1, 4, 8, 5]$. To provide an indication of the stiffness of a given structural topology, a concentrated force of of $f = 0.05\,\mathrm{N}$ is applied to node 9 in a direction 30° off of the y-axis (cf., Figure 5.16), while the displacement δ *in the same direction* is measured. The multiobjective hybrid topology/sizing optimization problem is considered in a manner consistent with the previous problem.

The resulting Pareto frontier after considering a population size of 100 for 1,000 generations is shown in Figure 5.17, along with several topologies of interest. It is interesting to note that the lightest design (normalized mass of 0.30, normalized stiffness of 186.0) is a simple two-member triangle-like structure that is oriented at an angle off of the y-axis almost equivalent with the applied force ($\approx 29.8^\circ$). As normalized mass increases, the sizing optimization scheme tends to keep structural members located on the plane associated with the face $[1, 2, 3, 4]$, perhaps to provide more stiffness in the y-direction given that the majority of the applied force acts in this direction, with additional members extending from nodes 5 and 8 to the edge between nodes 2 and 3 providing additional stiffness in the z-direction. The design with the largest normalized mass and highest stiffness retains all of its structural members after sizing optimization with the exception of members in the same plane as the fully-fixed boundary condition, which is consistent with the results of the 3-D sizing optimized truss previously discussed. Thus, without the ability to rely on a symmetry boundary condition to easily produce critical load paths, the hybrid optimization framework is still capable of producing elegant solutions to 3-D structural design problems using SPIDRS-generated topologies.

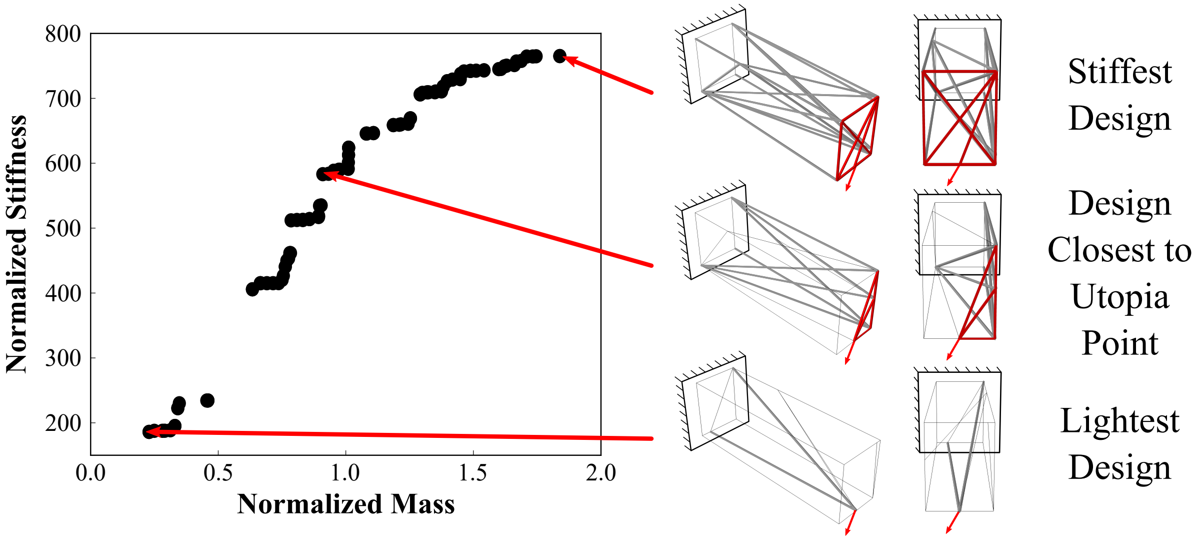


Figure 5.17: Pareto frontier and several topologies of interest after considering the design problem illustrated in Figure 5.16 for 1,000 generations with a population size of 100 using the proposed hybrid optimization framework. Structural members removed from each design by the sizing optimization scheme are denoted by thin lines.

6. APPLICATIONS OF GRAPH-BASED L-SYSTEM TOPOLOGY OPTIMIZATION*

To this point, this work has largely considered the proposed parameterized L-System/SPIDRS algorithm in the context of benchmark topology optimization problems, as such problems offer an opportunity for comparison to both other topology optimization methodologies and mathematically-proven optimal or ideal solutions. However, the motivation behind the development of a graph-based interpreter for L-System encodings stems from the need for a preliminary design method capable of considering complex multiobjective problems involving multiple physics for which the user may not have an intuition. This chapter briefly explores several applications of graph-based L-System topology optimization which include the design and optimization of tailorable stiffness structures, thermomechanical structures, and airfoils in supersonic flow.

6.1 Design and Optimization of Tailorable Stiffness Structures

In topology optimization problems, objective values to be minimized/maximized are typically calculated for a specific static loading condition. For example, in the tensile inverter problems discussed in this work one objective is to maximize the displacement inversion ratio for a specified input displacement of $\delta_{x,in} = -5\,\text{mm}$, which is independent of the response of the mechanism for input displacements up to this value. However, it can be desirable to tailor the response of the structure across the full loading envelope. Existing work in this area is sparse and has been applied only to the spatial response of compliant mechanisms [133, 134, 135]. This section demonstrates how a parameterized L-System coupled with the SPIDRS interpretation algorithm can be used to design structures capable of matching specified nonlinear force responses across a range of displacements. Specifically, as illustrated in Figure 6.1, given a specified goal stiffness curve, the proposed topology optimization framework can be used to generate designs with a minimum stiffness deviation. The ability to design and optimize tailorable stiffness structures could be beneficial

*Portions reprinted with permission from “Graph-Based Interpretation of L-System Encodings Toward Aeroelastic Topology Optimization of a Morphing Airfoil in Supersonic Flow’ by Bielefeldt, B. R., Hodson, J. D., Reich, G. W., Beran, P. S., Pankonien, A. M., Deaton, J. D., and Hartl, D. J., 2019. *Proceedings of the ASME 2019 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, Copyright 2019 by ASME.

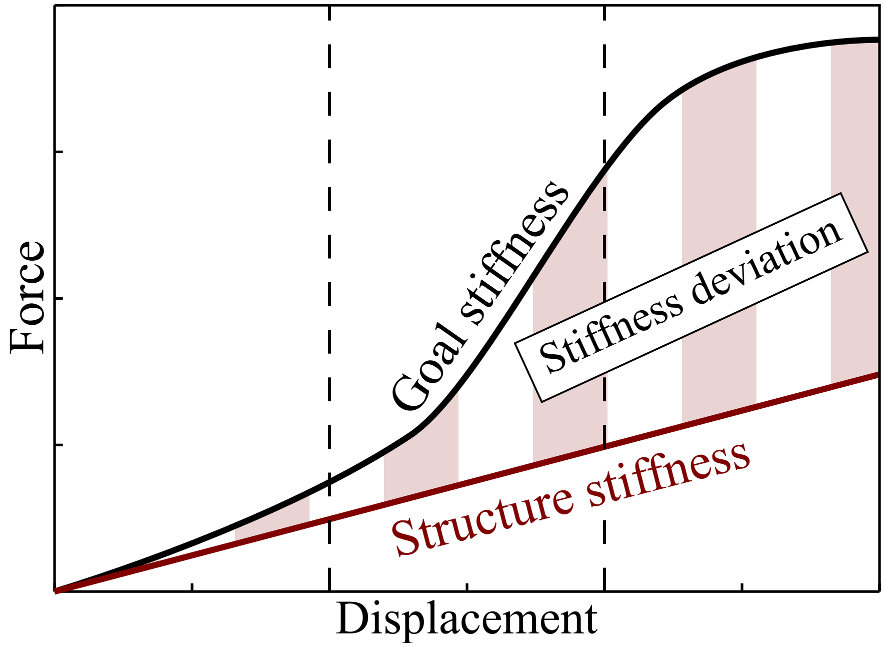
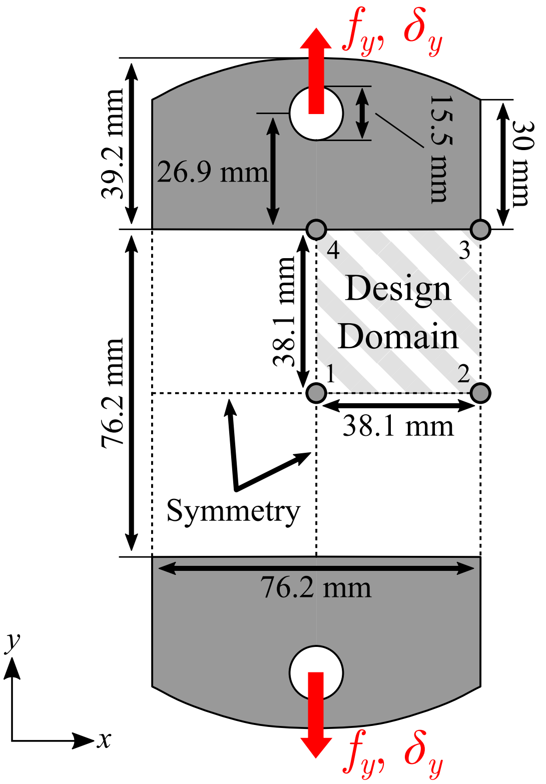


Figure 6.1: Example of the design problem considered in Section 6.1, where the goal is to minimize the deviation between a specified goal stiffness curve and the stiffness curve associated with a SPIDRS-generated topology.

in a wide range of applications including aeroelastic tuning and vibration control.

6.1.1 Problem Formulation

An illustration of the structural design domain for the tailorable stiffness design problems is shown in Figure 6.2. The overall domain consists of a square with a height and width of 78.1 mm, with symmetry boundary conditions applied to the x- and y-axes meaning that only one-quarter of this domain in considered in the design problem. The initial state of the graph is defined as $F = \{[1,4,3,2],[1,2,3,4]\}$; however, the edges associated with the initial graph are removed from the final graph topology. Generated structural members feature an assumed cross-section of $2.2\,\text{mm} \times 6.35\,\text{mm}$ (larger dimension out-of-plane) and consist of polypropylene with a specified elastic modulus of $E = 1.1\,\text{GPa}$, Poisson's ratio of $\nu = 0.39$, and a density of $\rho = 1,175\,\text{kg/m}^3$. Loads are transmitted to this domain via two connectors (top and bottom), which are designed to be compatible with a uniaxial load frame for future experimental testing and validation.



Polypropylene, 2.2 mm x 6.35 mm
Initial Graph: F={[1,4,3,2],[1,2,3,4]}

Figure 6.2: Initial graph and boundary conditions associated with the tailorable stiffness structure design study.

A summary of the multiobjective topology optimization problem to be considered is given in Table 6.1, and the problem is approached using the topology optimization framework introduced in Section 3.3. The goal of the problem is to design a structure capable of matching a specified nonlinear stiffness curve while also minimizing the mass of the structure. To obtain the stiffness curve of a given topology, an input displacement δ_y is applied to each design candidate (cf., Figure 6.2) in increments of 0.5 mm up to 5 mm, with the associated reaction force f_y recorded at every increment. The deviation between the goal stiffness curve and that of a given topology is

Table 6.1: Specifications for the multiobjective topology optimization problem of a tailorable stiffness structure using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Minimize:	mass, σ_{stiff}
by varying:	2 axiom characters,
	4 rule assignments (18 genes each),
subject to:	no constraints
NSGA-II Parameters [68]	
60 members for 400 generations,	
P_{cross} = 0.9, η_{cross} = 50,	
P_{mut} = 0.1, η_{mut} = 20	

then calculated using the root-mean- square error given as

$$\sigma_{stiff} = \sqrt{\frac{\sum_{i=1}^{10} \left(f_{goal,i} - f_{y,i}\right)^2}{10}}, \tag{6.1}$$

where $f_{goal,i}$ and $f_{y,i}$ are the goal reaction force and predicted reaction force for the structure at the ith loading increment, respectively. Structural analysis accounts for buckling and large rotations of members using implicit dynamic analysis and small loading increments.

6.1.2 Results

The first tailorable stiffness design problem considers a goal curve where the stiffness rate of change *increases* as the input displacement applied to the structure increases. This goal curve is defined by the equation

$$f_y(\delta_y) = 5.65\delta_y^3 - 17\delta_y^2 + 23\delta_y, \tag{6.2}$$

where the reaction force at $\delta_y = 5\,\text{mm}$ is based on that of a structure that consists only of straight, vertical members. The optimization problem detailed in Table 6.1 and illustrated in Figure 6.2 is considered using a population size of 60, 400 generations, and two levels of parameterized L-System recursion. A comparison of the stiffness curves associated with both the goal and the best performing SPIDRS-generated topology in terms of stiffness deviation in shown in Figure 6.3.

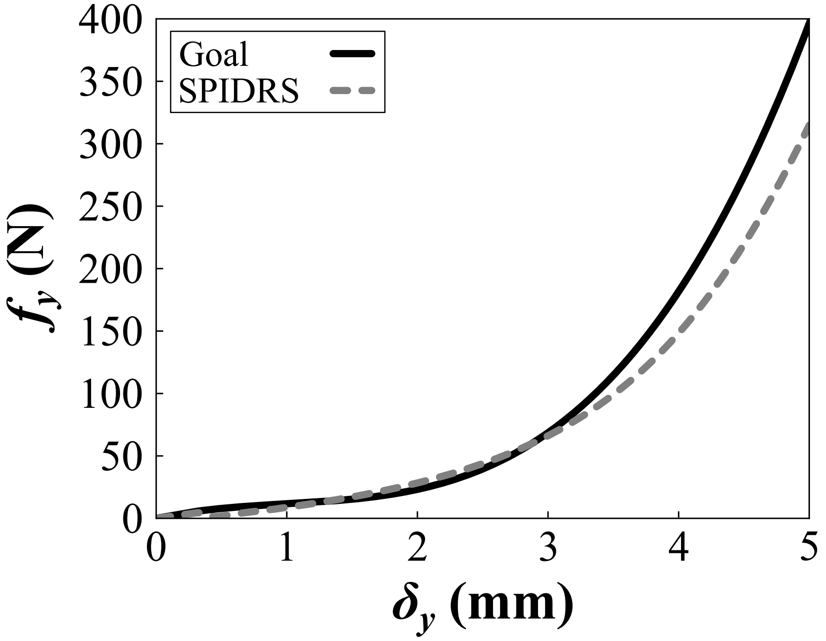
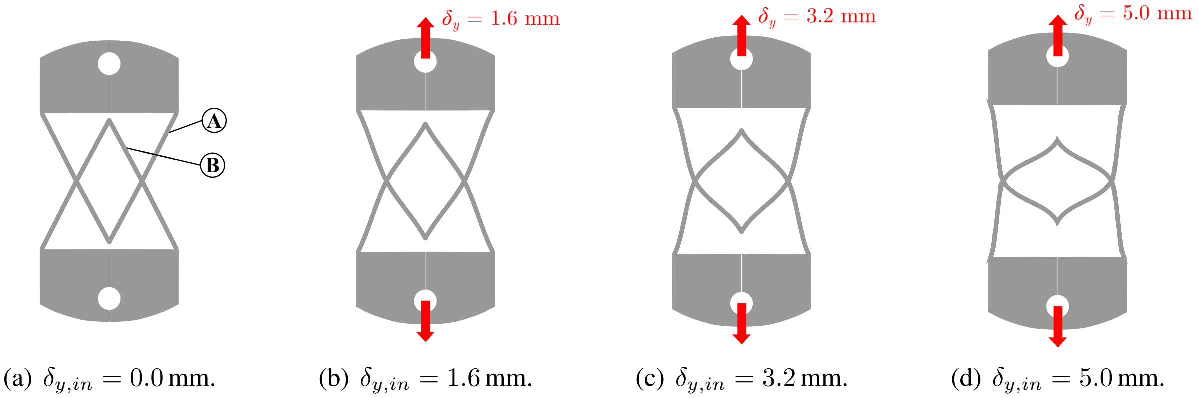


Figure 6.3: Comparison of the goal stiffness curve defined in Equation 6.2 to that associated with the best-performing SPIDRS-generated topology. Despite a slight deviation at higher magnitudes of input displacement, in general the two curves are in good agreement.

Despite a slight deviation for input displacements of $\delta_{y,in} \geq 4\,\mathrm{mm}$, in general the two curves are in good agreement. Figure 6.4 illustrates the deformation of this best-performing topology during loading. At the beginning of loading, the input displacement applied to the structure is transmitted through the A members and begins to buckle the B members as they are pulled outward (cf., Figure 6.4a-b). As loading continues, the A members begin to orient themselves parallel to the direction of the applied displacement, increasing the rate of change of the stiffness of the structure as the load case becomes more and more similar to that of a column subjected to uniaxial tension. However, note that at the end of loading (Figure 6.4d) the A members are not completely vertical, which would account for the higher magnitude of deviation between the stiffness of this topology and that of goal curve.

The second tailorable stiffness design problem considers a goal curve where the stiffness rate of change *decreases* as the input displacement applied to the structure increases. This goal curve



(a) $\delta_{y,in} = 0.0\,\text{mm}$. (b) $\delta_{y,in} = 1.6\,\text{mm}$. (c) $\delta_{y,in} = 3.2\,\text{mm}$. (d) $\delta_{y,in} = 5.0\,\text{mm}$.

Figure 6.4: SPIDRS-generated topology whose stiffness curve best matches that goal curve defined by Equation 6.2.

is defined by the equation

$$f_y(\delta_y) = 238.7e^{0.1104\delta_y} - 267.2e^{-3.46\delta_y}, \tag{6.3}$$

and the optimization problem once again considers a population size of 60, 400 generations, and two levels of parameterized L-System recursion. A comparison between the goal curve defined in Equation 6.3 and the SPIDRS-generated topology found to be best capable of matching that curve is shown in Figure 6.5. Despite having the shape general shape, there are significant differences between the two curves, especially at lower magnitudes of $\delta_{y,in}$. Figure 6.6 illustrates the deformation of the SPIDRS-generated topology during loading. As loading begins, the applied input displacement is transmitted through the A members to the B members by way of a compressive force as the A members attempt to orient themselves parallel to the direction of the input displacement (cf., Figure 6.6a). As the B members resist this compressive force the rate of change of the stiffness of the structure is relatively high (though still below the desired rate of change seen in the goal curve). However, as the B members begin to buckle (cf., Figure 6.6b-d) there is less resistance to the input displacement and the rate of change of the stiffness of the structure decreases. It should be noted in Figure 6.6a there is no gap between the B members due to the thickness as-

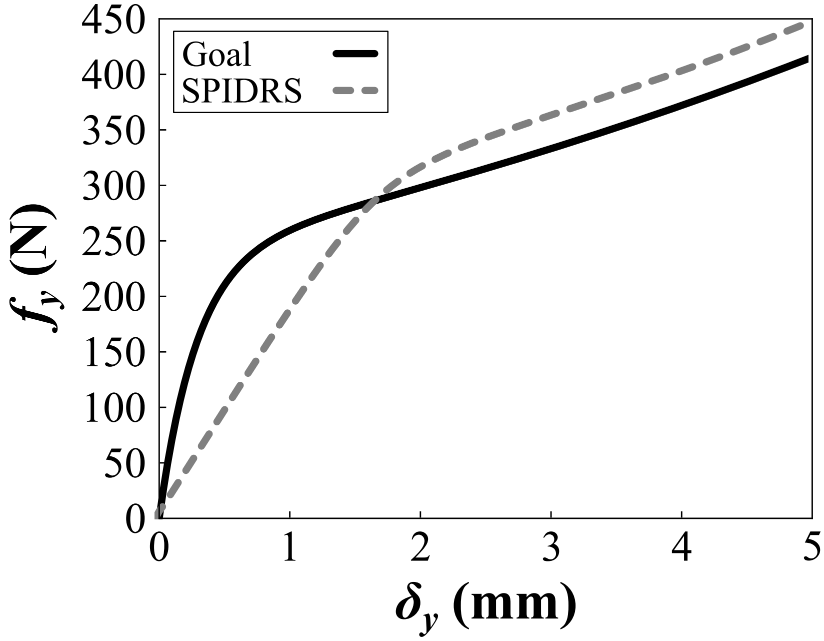
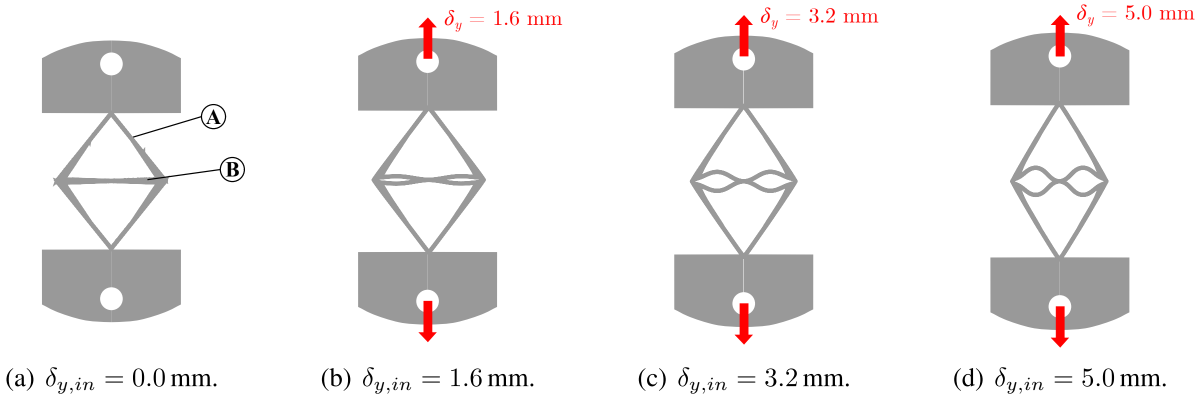


Figure 6.5: Comparison of the goal stiffness curve defined in Equation 6.3 to that associated with the best-performing SPIDRS-generated topology. While sharing similar shapes, there is noticeable deviation between the two curves, especially at lower magnitudes of input displacement.

sociated with each beam element, and thus in actuality these members would be unable to buckle, drastically altering the response of the structure. This is because the material overlap constraint detailed in Appendix D only applies to the topology within design domain and does not consider symmetry boundary conditions, which would need to be addressed in future studies.

It should be noted that these design problems assume a constant cross-sectional area for all generated structural members. However, preliminary studies have begun to consider varying the thickness of each member by treating the material assignment parameter in the SPIDRS graphical operations as a thickness parameter, which results in even more complex modules as shown in Figure 6.7. This could enable a variant of sizing optimization to be conducted using only a genetic algorithm and allow for the generation of better performing structures.



(a) $\delta_{y,in} = 0.0\,\text{mm}$.
(b) $\delta_{y,in} = 1.6\,\text{mm}$.
(c) $\delta_{y,in} = 3.2\,\text{mm}$.
(d) $\delta_{y,in} = 5.0\,\text{mm}$.

Figure 6.6: SPIDRS-generated topology whose stiffness curve best matches that goal curve defined by Equation 6.3.

6.2 Design and Optimization of Thermomechanical Structures

Many engineering design problems require the consideration of loads from multiple physical fields (i.e., structural, thermal, acoustic, aerodynamic, etc.). The ability to accurately consider these

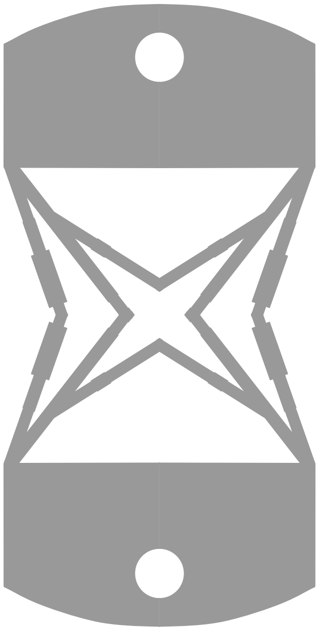


Figure 6.7: Example of a SPIDRS-generated structural module which features varying member thicknesses determined by the algorithm in a manner consistent with material assignments.

fields in a topology optimization framework is an obvious benefit in the context of preliminary design, where generating a better understanding of the design space can rapidly expedite the design process. In topology optimization, thermomechanical problems are commonly addressed due to the ease with which one can consider both structural and thermal responses using the same analysis tools [136, 137, 138, 139]. This chapter demonstrates the use of the proposed parameterized L-System and SPIDRS interpretation algorithm on a simple thermomechanical design problem.

6.2.1 Problem Formulation

Consider the periodic structure shown in Figure 6.8, which consists of small substructures with heights and widths of 100 mm each. The initial structure and subsequently generated structural members feature an assumed square cross-section of 1 mm × 1 mm and consist of aluminum with a specified elastic modulus of $E = 68$ GPa, Poisson's ratio of $\nu = 0.36$, density of $\rho = 2,700$ kg/m^3, specific heat of $c = 900$ J/kg $\cdot$ K, and thermal conductivity of $k_{thermal} = 210$ W/m $\cdot$ K. The initial state of the graph is defined as $F = \{[1,4,3,2],[1,2,3,4]\}$, with geometric symmetry boundary conditions placed on the planes associated with the half-edges e_{12} and e_{34}. Note that structural members along these axes of symmetry are removed from the final graph topology.

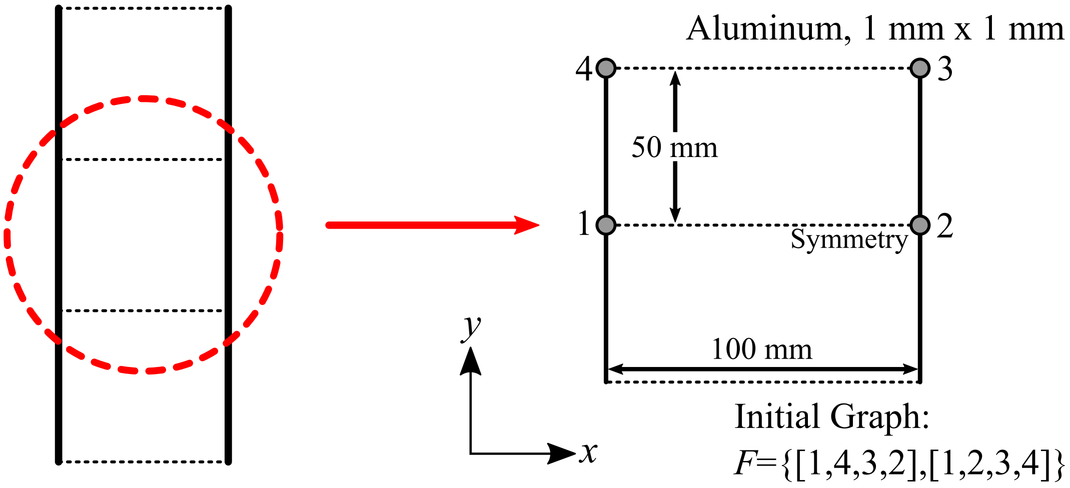


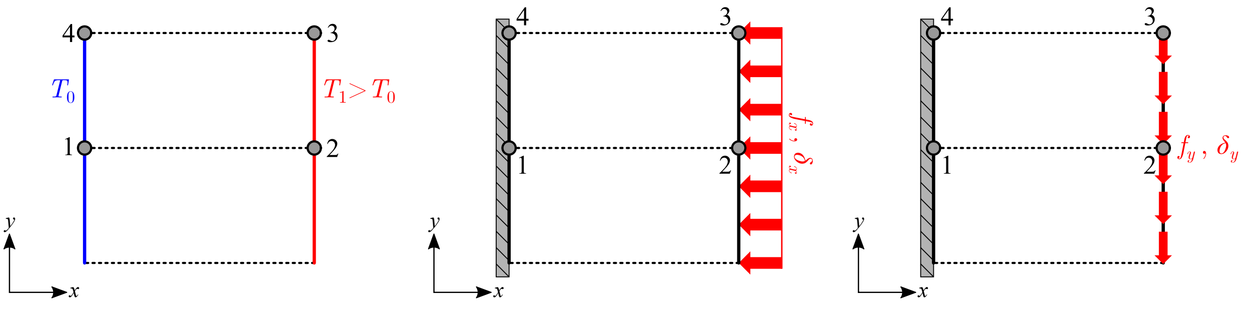
Figure 6.8: Initial graph and boundary conditions associated with the structurally-stiff insulator design problem.

Table 6.2: Specifications for the multiobjective topology optimization problem of a structurally stiff insulator using a graph-based interpretation of L-System encodings.

Design Problem Statement	
Minimize (Maximize):	$\hat{k}_{thermal}$ ($\hat{K}_{axial}$, $\hat{K}_{shear}$)
by varying:	2 axiom characters,
	4 rule assignments (18 genes each),
subject to:	$\sigma_{Mises}^{max} \leq 276\,\text{MPa}$
NSGA-II Parameters [68]	
100 members for 500 generations,	
$P_{cross} = 0.9$, $\eta_{cross} = 50$,	
$P_{mut} = 0.1$, $\eta_{mut} = 20$	

A summary of the multiobjective, multiphysical topology optimization problem to be considered is given in Table 6.2. The goal of the problem is to design a structure such that it is capable of acting as an insulator (i.e., minimizing thermal conductivity) while also maximizing its resistance to both axial and shear loads. Thus, three individual loading conditions are considered as shown in Figure 6.9. First, starting from an initial temperature of $T_0 = 298\,\text{K}$, a temperature of $T_1 = 318\,\text{K}$ is applied to the right side of the structure while the left side is held constant at T_0 (Figure 6.9a). The thermal conductivity of the structure is then calculated as

$$k_{thermal} = \frac{q_{thermal} L}{T_1 - T_0}, \tag{6.4}$$



(a) Thermal conductivity test. (b) Axial stiffness test. (c) Shear stiffness test.

Figure 6.9: Illustration of the three load cases for the structurally-stiff insulator design problem.

where $q_{thermal}$ is the heat flux, or rate per unit area at which heat flows through the structure, and L is the distance between the two walls of the structure. Next, a fully-fixed boundary condition (i.e., $u_x = u_y = 0$) is applied to the left side of the structure and an applied distributed force of $f_x = -0.001$ N/mm is applied to node 2 (Figure 6.9b). The resulting output displacement δ_x is then measured to provide an indication of the axial stiffness of the structure. Finally, the fully-fixed boundary condition is retained and an applied distributed force of $f_y = -0.001$ N/mm is applied to node 2 (Figure 6.9c), with the resulting output displacement δ_y providing an indication of the shear stiffness of the structure. Each load case is considered using a static, fully linear FEA analysis step.

6.2.2 Results

The optimization problem detailed in Table 6.2 and illustrated in Figure 6.9 is considered using the topology optimization framework discussed in Section 3.3 for 500 generations, a population

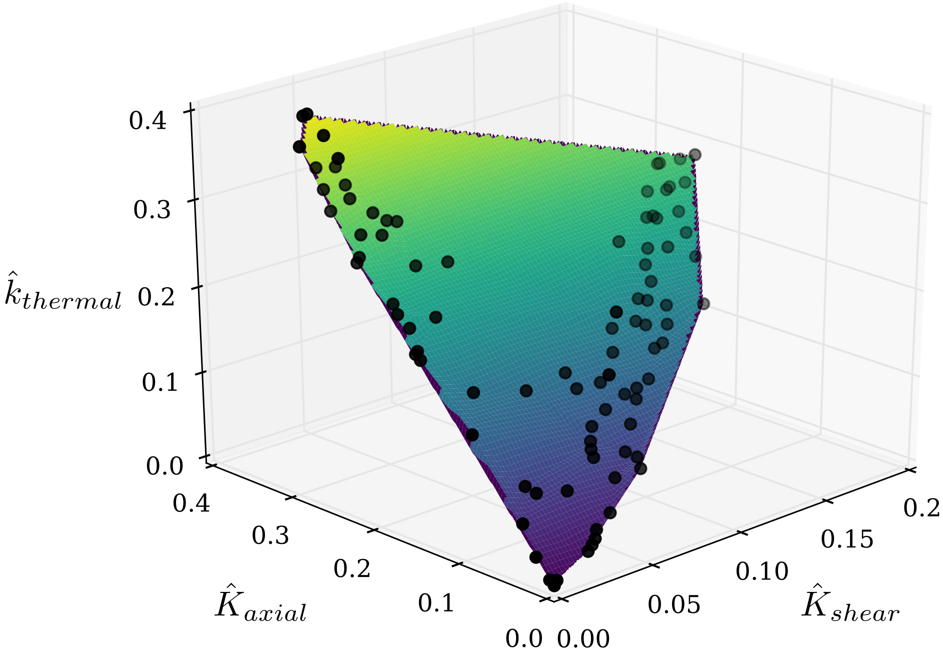
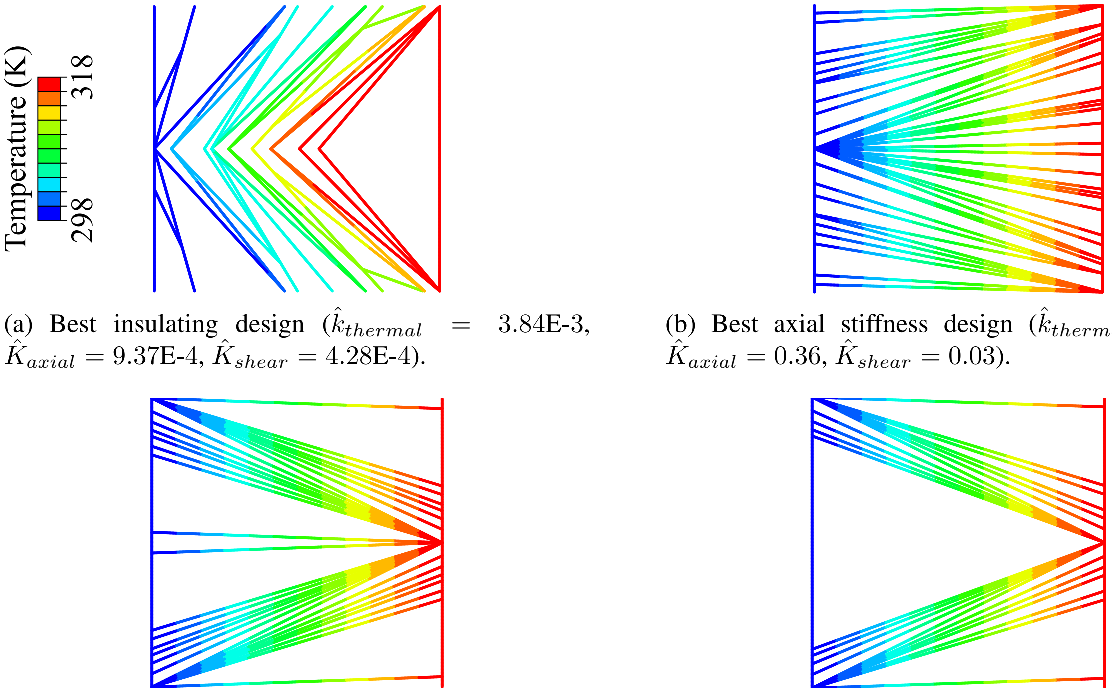


Figure 6.10: Three-objective Pareto frontier associated with the structurally-stiff insulator design problem detailed in Table 6.2 and illustrated in Figures 6.8-6.9.

size of 100 individuals, and three parameterized L-System recursions. For each design, the resulting thermal conductivity, axial stiffness, and shear stiffness values are normalized by those associated with a completely solid aluminum structure ($k_{thermal} = -2.1\,\mathrm{W/m \cdot K}$, $K_{axial} = 705\,\mathrm{GPa}$, and $K_{shear} = 96.8\,\mathrm{GPa}$). The resulting Pareto frontier is shown in Figure 6.10. The goal of minimizing the thermal conductivity of the structure favors designs with minimal load paths that are as long as possible (which will result in low structural stiffness), while the two stiffness objectives to be maximized call for the creation of numerous load paths (which will increase the thermal conductivity of the design). These trends are observable in the topologies shown in Figure 6.11, which also depict the temperature contours calculated when analyzing the load case illustrated in Figure 6.9a. Considering the best insulating design shown in Figure 6.11a, structural members are constructed



(a) Best insulating design ($\hat{k}_{thermal} = 3.84\text{E-}3$, $\hat{K}_{axial} = 9.37\text{E-}4$, $\hat{K}_{shear} = 4.28\text{E-}4$).

(b) Best axial stiffness design ($\hat{k}_{thermal} = 0.39$, $\hat{K}_{axial} = 0.36$, $\hat{K}_{shear} = 0.03$).

(c) Best shear stiffness design ($\hat{k}_{thermal} = 0.28$, $\hat{K}_{axial} = 0.24$, $\hat{K}_{shear} = 0.20$).

(d) Design closest to the utopia point ($\hat{k}_{thermal} = 0.21$, $\hat{K}_{axial} = 0.17$, $\hat{K}_{shear} = 0.16$).

Figure 6.11: Examples of SPIDRS-generated topologies along the Pareto frontier associated with the structurally-stiff insulator design problem, shown here with temperature contours calculated when analyzing the load case illustrated in Figure 6.9a.

in a zigzagged pattern, making the path between the two walls of the structure much longer than those seen in other designs and greatly reducing the thermal conductivity of the part. Contrast this with the topologies of the best axial stiffness design, best shear stiffness design, and design closest to the utopia point (Figures 6.11b-d), which feature multiple load paths for heat to flow through and thus an increased thermal conductivity. However, these multiple load paths serve to increase the stiffness of the these topologies when subjected to axial and shear loads.

6.3 Design and Optimization of Airfoils in Supersonic Flow

While aircraft wing morphing has garnered increased attention in recent years, the idea of modifying the shape or geometry of the wing dates back to the first aircraft. Inspired by the ability of birds to rapidly change the shape of their wings for maneuverability in ever-changing flight conditions, the Wright Flyer was equipped with cables that could increase or decrease wing twist to obtain roll control. The need to carry larger payloads and withstand greater aeroelastic forces while traveling at higher cruise speeds led to the design of rigid aircraft structures that are in use today [140]. These structures rely on discrete control surfaces to alter the wing geometry to allow for increased aerodynamic performance over a fixed range of flight conditions. Outside of this range, however, these control surfaces can negatively influence aerodynamic performance and lead to reduced overall efficiency [141]. Furthermore, the types of geometry changes the wing can undergo are fairly limited when compared to those possible using an inherently deformable structure.

Ideally, the actuation system responsible for inducing geometric changes should be integrated directly into the wing itself, thereby reducing the complexity of the overall structure. This is a challenging problem, as the wing must be capable of withstanding a wide variety of aeroelastic loading conditions while also remaining "compliant" enough to undergo a wide variety of shape changes based on the mission profile of the aircraft. Recent advancements in the field of multifunctional materials have provided an opportunity to explore and develop innovative actuation systems [142], such as those utilizing shape memory alloys (SMAs) [143, 144] and piezoelectric materials [145, 146]. These material systems are driven by an external physical field, such as temperature changes and applied electric field in the two examples above. Integrating these systems into an already challenging aeroelastic problem requires multidisciplinary thinking and novel design methods capable of providing inspiration for innovative aircraft systems.

This work seeks to extend that of Hodson et al., who used the map L-System to develop the structural topology of a morphing airfoil subjected to supersonic flow [147]. Here, the map L-System will be substituted for the parameterized L-System interpreted using the graph-based SPIDRS algorithm introduced in Chapter 3. It will be shown here that the graph-based L-System interpretation approach is also a viable and promising option in the design of adaptive structures, where its ability to efficiently generate structures with integrated multifunctional materials can provide an intuition for potential actuation systems in the next generation of aircraft.

6.3.1 Optimization Framework

6.3.1.1 Evolutionary Design Process

Generating optimal structural topologies using the parameterized L-System and SPIDRS interpreter requires an optimization approach capable of operating with discrete design variables over a discontinuous design space. This problem utilizes an implementation of the well-known NSGA-II genetic algorithm [68] available in the open source Python package *Inspyred* [148] as shown in Figure 6.12. The process begins with NSGA-II generating a random population of design candidates, with each candidate defined by a genome, or list of values, that determines the axiom, production rules, and formal parameters needed to uniquely define the structural topology of the candidate (cf., Section 3.3). After the genome is interpreted using the coupled parameterized L-System/SPIDRS algorithm, basic checks are performed to ensure the validity of the structural topology, with invalid topologies penalized via assignment of an artificially poor fitness value. If a topology is deemed valid, it is passed to an aeroelastic solver which evaluates both the structural response and aerodynamic performance of the design candidate and returns a design fitness based on the problem formulation (cf., Section 6.3.1.2). Once all fitness values for each candidate in the population have been computed, the genetic algorithm uses these values and evolutionary tech-

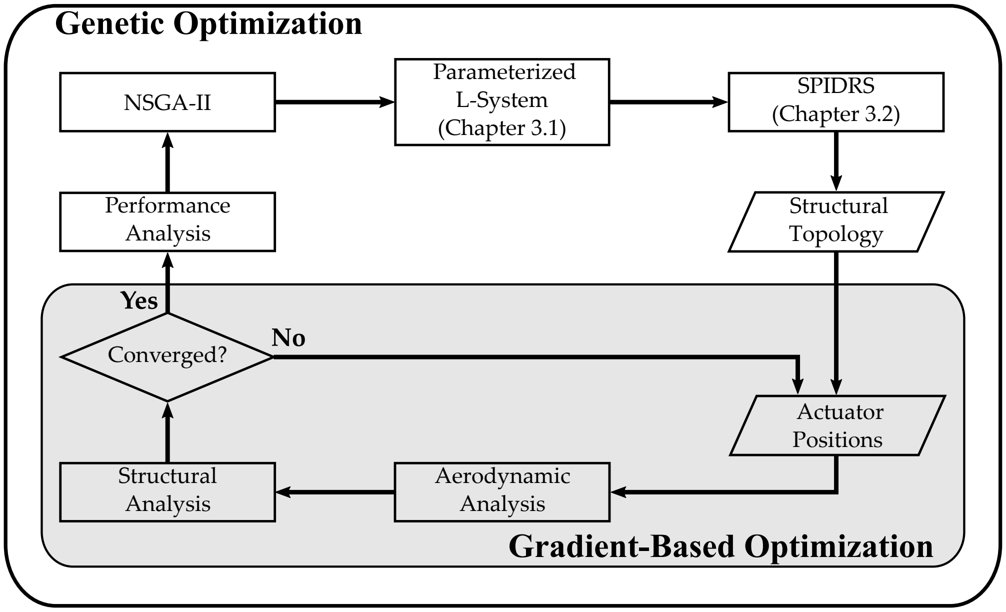


Figure 6.12: Overview of the optimization framework for the morphing supersonic airfoil problem. Topology generation is driven by NSGA-II, while gradient-based optimization maximizes the lift-to-drag ratio of a given design candidate by changing the positioning of actuators.

niques such as mating, crossover, and mutation to generate a new generation of design candidates. This cycle is then repeated until one of the genetic algorithm's stopping criteria is met.

Upon successful completion of the evolutionary design optimization process, the final population will contain candidate designs that demonstrate improved performance when compared to those in the initial population. For simple problems with sufficiently small design spaces, this final population will contain the most optimized design possible for the given problem and design space. However, the design spaces associated with adaptive structures problems, particularly when considering multiphysical problems such as the one considered herein, are so large and complex that it becomes impossible to fully explore the design space given current computing capabilities. Thus, the goal of this work is not to identify a specific deterministic design, but rather to explore the design space for non-intuitive solutions, identify trends and commonalities among well-performing designs, and gain insights into specific solutions that conventional design methods may not consider.

6.3.1.2 Fitness Evaluation Process

The fitness evaluation process itself uses a gradient-based optimization algorithm to determine the maximum achievable lift-to-drag ratio by changing the positions of actuators in a given design candidate subject to defined constraints (cf., Figure 6.12). The gradient-based optimizer drives the design vector (i.e., the list of current actuator positions) towards optimal values by obtaining function and derivative values (via direct analytical derivatives) from the aeroelastic solver.

The aeroelastic solver is an iterative process that couples low-fidelity aerodynamic and structural analysis tools. Aerodynamic analysis is performed using third-order piston theory (see [149]) to predict aerodynamic loads on the airfoil skin, which are then provided to a structural analysis tool. This tool utilizes elementary beam theory (see [128]) to predict deformations of the structure based on both the actuator positions and computed aerodynamic loads as well as the typical section method (see [150]) to determine rigid-body rotation of the airfoil. The deformed and rotated coordinates of the airfoil are then passed back to the aerodynamic solver, which updates the predicted aerodynamic loads based on the deformed shape of the structure. This process continues iteratively until changes in the aerodynamic loads, skin deformations, and rigid-body rotation of the airfoil fall below user-specified thresholds. The resulting lift-to-drag ratio and corresponding design vector are returned to the gradient-based optimizer, which then calculates a new design vector. This process is repeated until either an optimized solution is obtained or the gradient-based optimizer determines that no solution capable of satisfying the prescribed constraints can be found. If an optimized solution is found, the resulting lift-to-drag ratio is pass back to the genetic algorithm (cf., Section 6.3.1.1). Topologies for which there are no suitable solutions are assigned an artificially poor lift-to-drag ratio so that the candidate will be eliminated from future generations and more satisfactory designs may emerge.

6.3.2 Problem Formulation

Consider a supersonic aircraft that operates primarily at a specified cruise condition but still requires the ability to perform brief transient maneuvers such as climb, descent, and roll. The

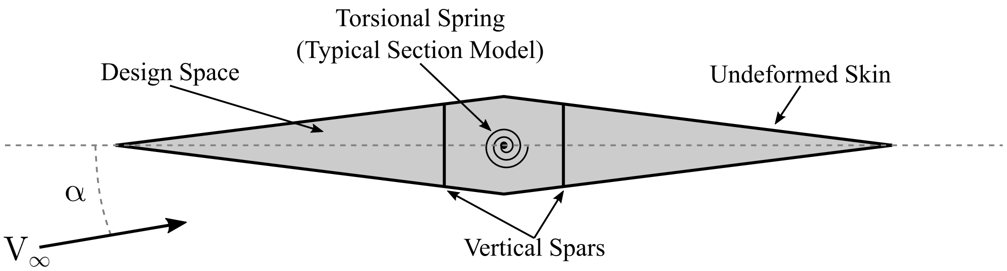


Figure 6.13: Illustration of the design problem for a supersonic morphing airfoil.

aircraft should be designed for maximum aerodynamic efficiency under cruise conditions while still being able to meet specific minimum maneuverability requirements. Here, the problem seeks to solve a simplified abstraction of this design problem by determining the topological layout of a 2-D supersonic airfoil that includes a skin, vertical spars, and internal stiffeners, and relies on linear internal actuators rather than rigid control surfaces to actively control deformations of the airfoil skin under aerodynamic loads to achieve efficient operation under cruise conditions and meet minimum maneuverability requirements.

A cross section of the design problem is shown in Figure 6.13, and the geometric properties of the airfoil and freestream properties are shown in Table 6.3. A flexible skin in the shape of a conventional diamond airfoil is supported by two vertical spars that are fixed in displacement at their midpoints but are otherwise allowed to deform and/or rotate. Both skin and spars are composed of uniform beams with material properties and cross-sectional dimensions as indicated in Table 6.4; the cross-sectional width of these members represents the distance between the ribs in an abstracted 3-D wing. A rotational spring located at the center of the airfoil with a torsional spring constant $\kappa = 10^6$ N/rad is used to represent the torsional stiffness of a 3-D wing. The design space is the internal area of the airfoil, where stiffeners and actuators can be placed in any configuration and with any number of connections between themselves, the outer skin, and the vertical spars. The stiffeners and actuators are also composed of rectangular cross-sections (Table 6.4), though the cross-sectional widths are not tied to the spacing between ribs in the abstracted 3-D wing but to the thickness of an individual rib.

Table 6.3: Airfoil geometry and freestream properties.

Airfoil Geometric Properties	
Chord (m)	2.0
Maximum Thickness-to-Chord Ratio	2.5%
Angle of attack	2.0°
Freestream Properties	
Mach Number	2.0
Operating Pressure (kPa)	12.1
Ratio of Specific Heats	1.4

Table 6.4: Section and material properties of structural components.

Property	**Skin**	**Spars**	**Stiffeners**	**Actuators**
Width (cm)	10	10	2	2
Thickness (cm)	0.5	0.5	2	2
Density (kg/m^3)	2,700	8,000	2,700	6,450
Elastic Modulus (GPa)	69	200	69	69

The design objective is to maximize the lift-to-drag ratio of the airfoil at cruise subject to specific lift and pitching moment M_{pitch} requirements. The design is also subject to constraints for minimum and maximum achievable lift (L_{min} and L_{max}, respectively) to meet the maneuverability requirement as well as limitations on the actuation force F_{act}, actuation strain ϵ_{act}, and actuator stroke length u_{act} at every airfoil configuration. The global optimization problem governed by the evolutionary design process discussed in Section 6.3.1.1 reads

$$\begin{aligned} \max \quad & (L/D)_{cruise} \\ \text{s.t.} \quad & L_{cruise} = W \\ & L_{min} \leq L_{min,target} \\ & L_{max} \geq L_{max,target} \\ & M_{pitch} \leq 300\,\mathrm{N} \cdot \mathrm{m} \\ & |F_{act}| \leq 1000\,\mathrm{N} \\ & |\epsilon_{act}| \leq 10\% \\ & |u_{act}| \leq 2\,\mathrm{cm}. \end{aligned}$$

Evaluation of the aerodynamic performance at each airfoil configuration (L_{cruise}, L_{min}, and L_{max}) are governed by the gradient-based optimizer and aeroelastic solver discussed in Section 6.3.1.2. These aerodynamic objectives are optimized by determining the number, placement, and properties (i.e., force, strain, etc.) of actuators in the SPIDRS-generated airfoil. In this work, the number of actuators is limited to between two and eight, and there is no constraint on the total amount of actuation energy to be used. At cruise, each airfoil is required to generate enough lift to match the weight of the aircraft W, which is given by

$$W = nW_{airfoil} + W_{payload}, \tag{6.5}$$

where n is the ratio of the weight of the abstracted 3-D wing relative to the weight of the 2-D airfoil $W_{airfoil}$ (calculated using the properties shown in Table 6.4) and $W_{payload}$ is the weight of everything not included in the weight of the wing (e.g., fuselage, crew, fuel, ordinance, etc.). In this work, it is assumed that $n = 20$ and $W_{payload} = 3,140\,\mathrm{N}$. Minimum and maximum achievable lift configurations for each airfoil must also meet certain target values to demonstrate maximum adaptability across a flight envelope. These are given by

$$L_{min,target} = (1 - c_a)L_{cruise} \tag{6.6}$$

and

$$L_{max,target} = (1 + c_a)L_{cruise}, \tag{6.7}$$

where c_a is a coefficient used to specify how adaptable the airfoil should be. A low specified value of c_a results in little difference in achievable lift between the three considered airfoil configurations, while a high specified value of c_a results in larger differences and therefore increased adaptability.

6.3.3 Results

The design problem introduced in Section 6.3.2 is analyzed over 1,000 generations using a population size of 1,000 individuals. The results of the optimization are shown in Figure 6.14. The striations prior to the start of generation 150 are due to changes in the coefficient c_a, which initiated at 5% and incrementally increased every 25 generations until stopping at 50%. This process is used to filter out structural configurations that demonstrated an inability to actuate. Between 150 and 1,000 generations, the optimization slowly begins to converge towards a majority of solutions exhibiting a maximum $(L/D)_{cruise}$ between approximately 18.6 and 19.1. A group of exceptional designs are able to achieve ratios over 19.1, including one with a maximum of over 19.6. The four best performing designs are selected to be examined in greater detail in an attempt to discern any noticeable trends or commonalities.

Topologies of the four best performing configurations under the constraints specified in Section 6.3.2 are shown in Figure 6.15, along with the locations of the actuators determined by the gradient-based optimization. It is interesting to note that, despite being able to employ between two and eight actuators, the top performing designs all utilize six actuators. This is likely because six actuators represents a balance between actuation capability, which impacts the adaptability of the airfoil, and the weight of the airfoil, which is penalized by requiring a greater amount of lift. Also, note that for the most part actuators are concentrated in between the two spars of each design.

This would indicate that generating deformations and/or rotations in this part of the airfoil has the biggest effect on aerodynamic performance for supersonic flight.

The deformed topologies for the four best performing designs during cruise, minimum lift, and maximum lift configurations are shown in Figures 6.16-6.18. The color of the actuator denotes both the magnitude of the actuation force and whether the actuator is expanding or contracting. The first noticeable trend is that each design features an actuator (actuator 1 in each configuration, see Figure 6.15) towards the leading edge of the airfoil that actuates fully during cruise (Figure 6.16). This has the effect of resisting aerodynamic loads on the leading portion of the airfoil, leading to a camber line that remains relatively flat; this is expected, as a perfectly straight camber line represents the theoretical maximum L/D for supersonic flow. Minimizing the required actuation energy during cruise would likely result in more diverse actuation systems and may be considered

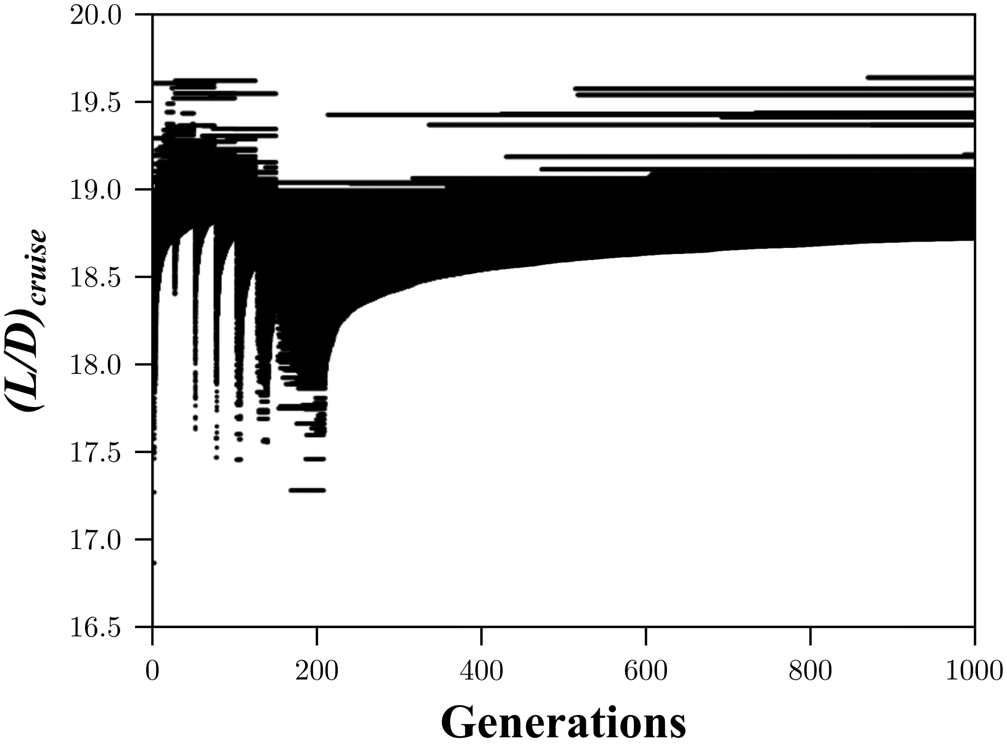
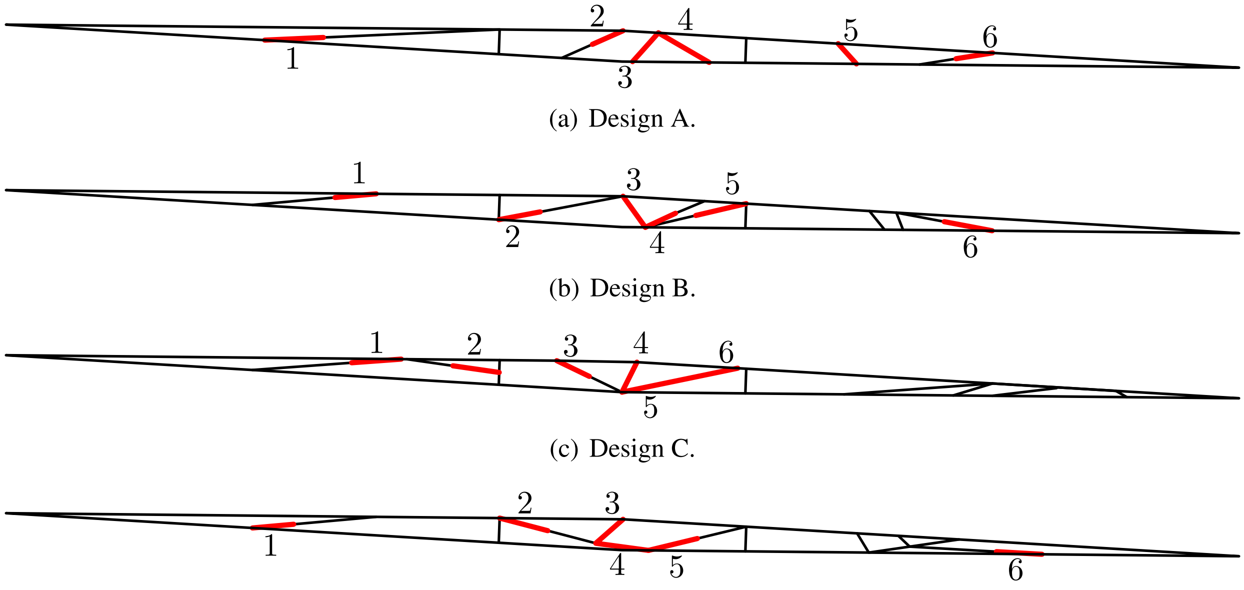


Figure 6.14: Evolution of $(L/D)_{cruise}$ for the design problem described in Section 6.3.2 when considering a population size of 1,000 over 1,000 generations.



(a) Design A.

(b) Design B.

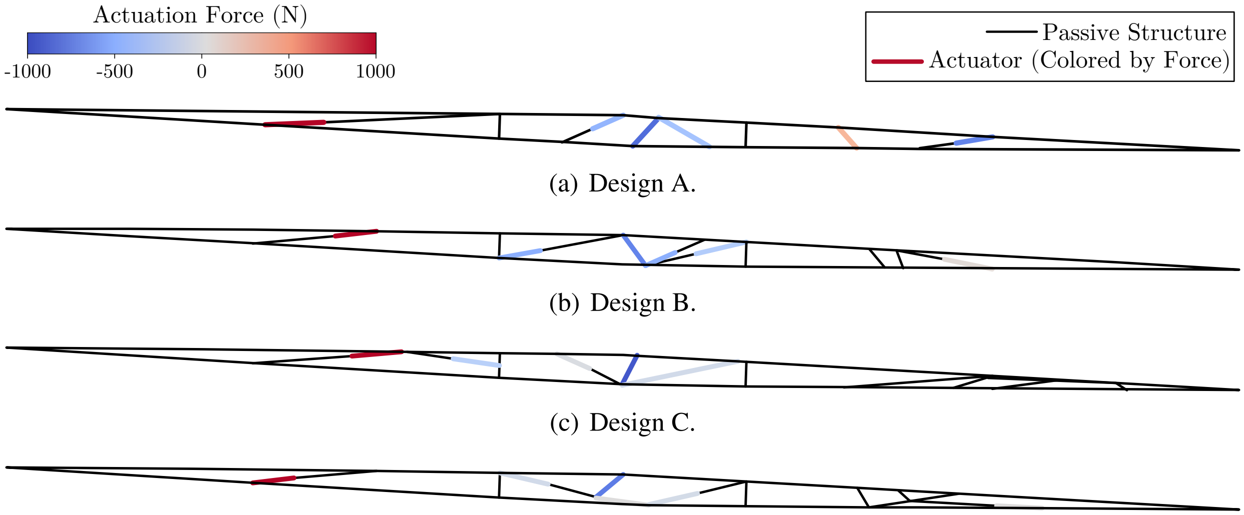
(c) Design C.

(d) Design D.

Figure 6.15: Topologies and actuator locations for the four best performing airfoil designs.

in future work. These actuators remain in use in the minimum and maximum lift configurations, but they seem to have far less impact. A second commonality between all designs is the existence of an actuator that acts primarily vertically (actuators 3, 3, 4, and 3 in designs A, B, C, and D, respectively). During cruise, these actuators are all contacting, which decreases the thickness of the airfoil and leads to improved L/D.

Another observable trend concerns the behavior of the actuators in between the spars in each design during all three aerodynamic configurations. All actuators that have a positive slope expand in the maximum lift configuration and contract in the minimum lift configuration. Alternatively, all actuators that have a negative slope contract in the maximum lift configuration and expand in the minimum lift configuration. This combined effect leads to the spars rotating depending on the desired aerodynamic performance. Rotating the spars clockwise leads to an increased angle of attack on the leading and trailing edges, which in turn increases the amount of lift generated by the airfoil. Rotating the spars counter-clockwise decreases the angle of attack and therefore decreases the amount of lift generated by the airfoil. It is also interesting to note that in each



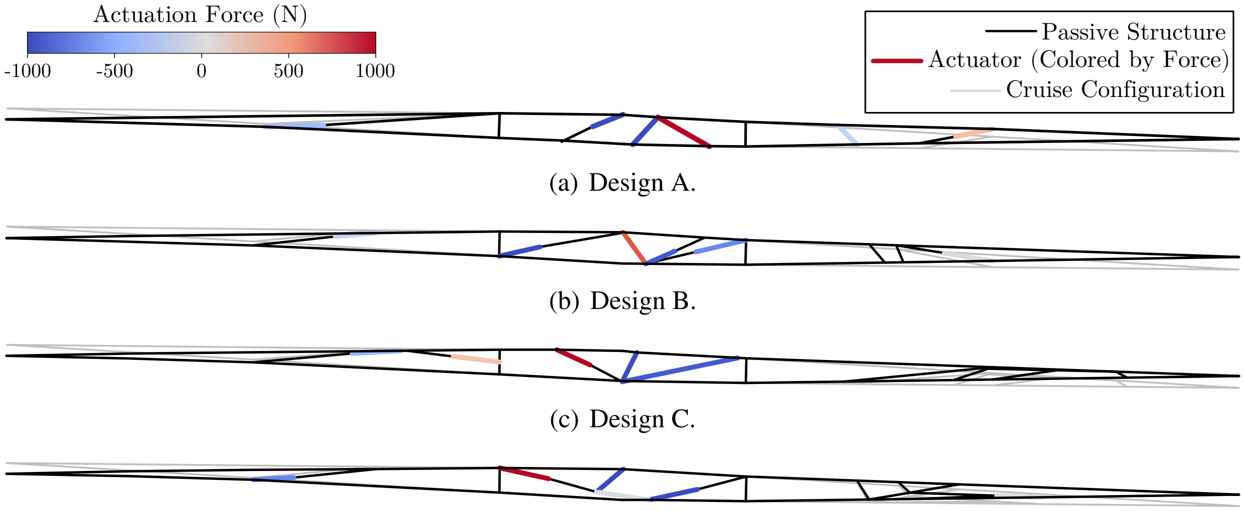
(a) Design A.

(b) Design B.

(c) Design C.

(d) Design D.

Figure 6.16: Deformed topologies during cruise for the four best performing airfoil designs. The color bar indicates the magnitude of actuation force in each actuator.



(a) Design A.

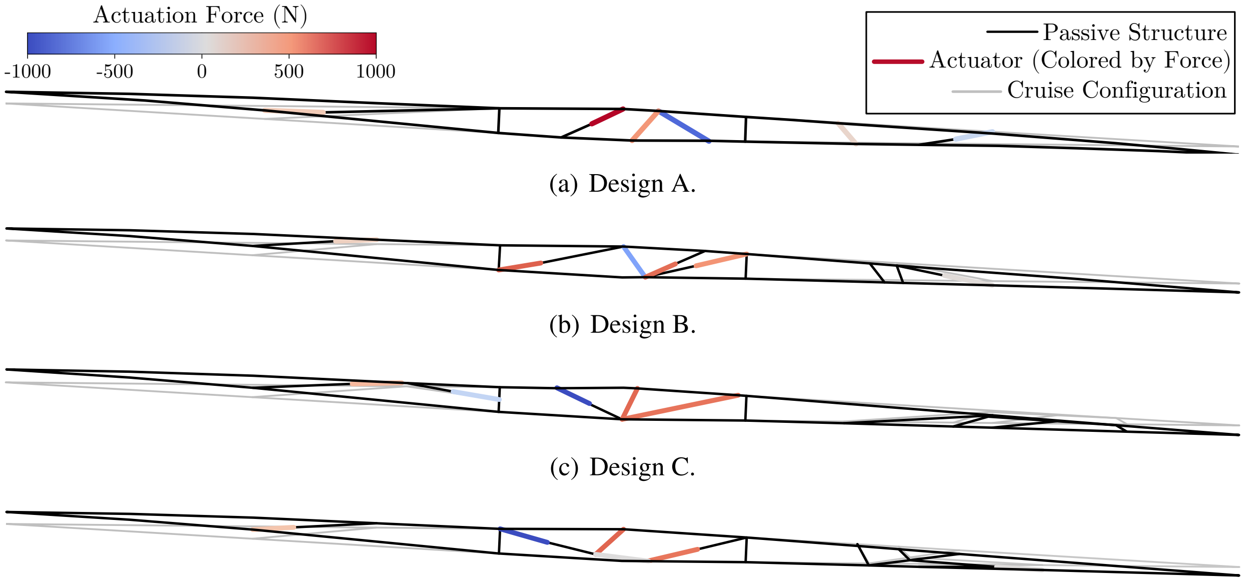
(b) Design B.

(c) Design C.

(d) Design D.

Figure 6.17: Deformed topologies for a minimum lift configuration for the four best performing airfoil design. The color bar indicates the magnitude of actuation force in each actuator.

configuration the forward and aft spars are rotated in the same direction and by approximately the same amount. This keeps the leading and trailing edges relatively parallel and helps maintain a



(a) Design A.

(b) Design B.

(c) Design C.

(d) Design D.

Figure 6.18: Deformed topologies for a maximum lift configuration for the four best performing airfoil designs. The color bar indicates the magnitude of actuation force in each actuator.

reasonable pitching moment ($< 300\,\mathrm{N \cdot m}$). However, during cruise all actuators between the spars are contracting, and the spars stay relatively rotation-free.

Figure 6.19 shows a comparison of $(L/D)_{cruise}$ and the average absolute force per actuator for the four designs, which can be directly linked to the amount of energy required by the actuation system. During cruise, the average absolute force needed for designs C and D are far below those needed for designs A and B. In fact, despite having an $(L/D)_{cruise}$ value within approximately 1.5% of design A, design D requires almost half the average absolute force. Inspection shows that this is due to the fact that design D features two “dominant” actuators (actuators 1 and 3) that drive the majority of the necessary shape change during cruise, while design A appears to utilize all six actuators in its cruise configuration. Design D appears to use it’s other actuators in the minimum and maximum lift configurations, and the average absolute forces are understandably higher. However, given that the aircraft would only be in these configurations for short periods and would operate primarily at cruise conditions, future work could look to include minimizing actuation energy at cruise to the optimization problem.

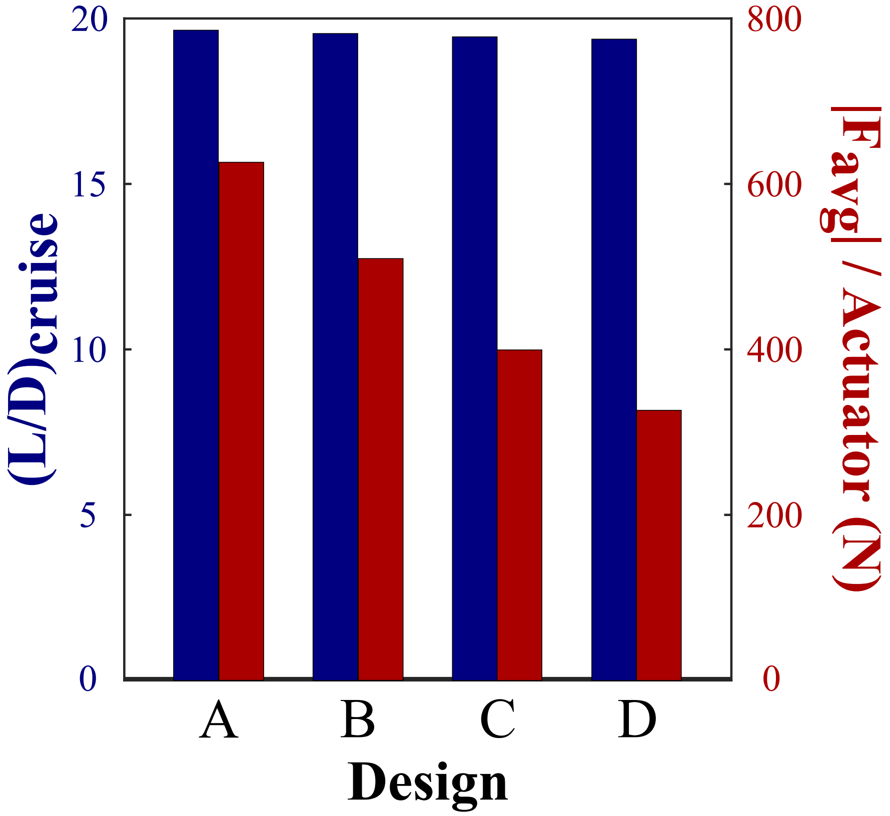


Figure 6.19: Comparison of the lift-to-drag ratio at cruise and average actuation force per actuator required for the four best performing airfoil topologies. Despite having roughly the same lift-to-drag ratio as design A, design D requires almost half the average absolute actuation force.

7. CONCLUSIONS

To conclude this dissertation, the key points presented in each of the previous chapters are summarized, and future research tasks pertaining to each are suggested.

7.1 Two-Dimensional L-System Topology Optimization Using Geometry-Based Interpretation

Chapter 2 described the development of a topology optimization framework that uses a geometry-based interpretation of L-System encodings. The L-System takes 52 independent design variables and encodes them into a complex string of characters using an axiom, production rules, and n numbers of recursion. However, this string of characters has no spatial significance until interpreted by some algorithm. Here, a geometry-based interpreter known as turtle graphics was employed, which translates each character of the L-System encoding into translational or rotational instructions for a line segment creating-agent. This results in the creation of fractal, branch-like structural topologies. The L-System description and turtle graphics interpreter was then coupled with an FEA package and genetic algorithm for the purpose of driving populations of structures towards designs capable of simultaneously satisfying multiple design goals associated with multiple functions.

To assess the ability of the proposed L-System topology optimization framework to find effective configurations for multifunctional design problems, three design problems of increasing complexity were considered. The first problem focused on a simple cantilevered frame problem with the goal of maximizing structural stiffness while minimizing mass. Comparisons to results from conventional topology optimization methods as well as a family of Michell trusses, the mathematically proven optimal result, revealed that, despite analyzing 100,000 topologies, designs developed using the L-System framework demonstrate a relatively poor performance. Furthermore, when given explicit L-System instructions that resulted in an identical topology to that of a three-bar Michell truss, the L-System was still outperformed. It was determined that the inability of this L-System formulation to match the performance of a traditional Michell truss was due to the restrictions placed upon possible segment length parameters and segment turn angles. Given a finite number of discrete lengths a given segment could have and a finite number of angles a given segment could turn through, it was incredibly difficult for the framework to generate structures that featured a distributed stress field when loaded, which can be directly correlated to structural stiffness.

Despite the aforementioned limitations of the turtle graphics algorithm, it did show promise in the design of compliant mechanisms. Specifically, the Pareto frontier generated by the L-System framework compared well against the ideal solution for a tensile inverter (with a goal of maximizing both the displacement and force inversion behavior of a mechanism), and designs associated with this frontier featured the same load path geometries found in solutions obtained using a SIMP implementation. The framework also performed well when designing a kinematic rectifier, a non-intuitive problem in which the goal was to maximize the response of the mechanism in a single direction regardless of the direction of an input displacement. Here, turtle graphics demonstrated an ability to create a truly multifunctional structure, as different structural members provided different functionality depending on the direction of input to achieve the desired mechanism response.

While proving effective in evaluating the response of a large number of designs in a relatively short amount of time, the framework neglects critical aspects during modeling such as overlapping material, precise calculation of inter-segment connections, and the consideration of self-contact. To assess whether the proposed framework is capable of producing physically realizable designs, select topologies from the three design problems compared to both high-fidelity FEA models considering more realistic continuum responses and experimentally characterized prototypes for the purposes of validation. For each topology considered, the response predicted by the continuum-based FEA model compared favorably with the beam-based FEA model used by the optimization framework. Furthermore, experimentally characterized prototypes demonstrated an error of less than 18% of the maximum magnitude of the response for each test case. These results indicated that, despite the low-fidelity FEA approach, the proposed optimization framework is capable of determining realistic solutions to structural topology optimization problems.

In spite of its successes in the design of compliant mechanisms, geometry-based interpretations of L-System encodings were shown to be suboptimal. Structure creation and angle changes were a function of just two variables, severely limiting the design space. Additionally, its reliance upon a knowledge of the geometry of the structure at all times proved to be computationally inefficient compared to other potential methods. Furthermore, the creation of load paths, formation of void regions, and material changes, all of which form the very essence of topological description, occurred coincidentally when line segments overlapped or crossed. Thus, while perhaps having potential in design problems involving fluid flow or heat transfer, the use of the turtle graphics algorithm for the interpretation of L-System encodings proved to be unfit for purpose in structural topology optimization.

7.2 Two-Dimensional L-System Topology Optimization Using Graph-Based Interpretation

Having demonstrated a need for an interpretation algorithm better suited to structural topology optimization, Chapter 3 introduced a novel graph-based method inspired by referred to as Spatial Interpretation for the Development of Reconfigurable Structures (SPIDRS). By using the idea of connectivity found in graph theory, this interpreter describes the topology as a function of the nodes, edges, and faces of a planar graph. After translating a character of a parameterized L-System encoding, which provides for greater design freedom than the L-System utilized in Chapter 2, SPIDRS induces topological changes by simply modifying the connectivity of the graph based on the graphical operations associated with that character. These graphical operations were developed to ensure that movement, structure creation, and material assignments remain deliberate and natural.

The ability of the SPIDRS algorithm to effectively explore the design space and develop optimal structural topologies was assessed using two design problems. While outperforming the geometry-based L-System interpretation method, cantilevered frame designs generated by SPIDRS were still significantly outperformed by the Michell truss. However, it should be noted that several SPIDRS-generated designs did contain topological features similar to a Michell truss. Furthermore, SPIDRS did compare favorably with a SIMP implementation in regards to the performance of designs with a relatively low normalized mass value. SPIDRS was also shown to be adept at considering the design of high-efficiency tensile inverters, many of which once again featured load path geometries identical to that obtained using a SIMP implementation.

The graph-based foundation of SPIDRS also allows for the demonstration of several extensions that increase design flexibility. The first is the include the shape of the initial graph in the optimization (i.e., shape optimization), which permits the algorithm to explore configurations that further optimize the performance of a given topology. The second is the ability to assign materials to the face of a graph, enabling the consideration of large continuum domains in addition to the slender beam members generated by SPIDRS. This operation could serve as a bridge between traditional density-based topology optimization methods and the proposed L-System method, combining both solid and branched structural members to achieve optimal performance. These two extensions were both demonstrated using tensile inverter design studies, where the inclusion of shape optimization both improved SPIDRS-generated mechanism performance and provided insight into the trends behind well-performing designs and the inclusion of face material assignments resulted in much higher attainable magnitudes of both force and displacement inversion behavior.

Returning to the originally introduced SPIDRS algorithm, several key components were considered and modified to further optimize the performance of resulting designs. It was observed during the cantilevered frame design study that SPIDRS-generated designs featured dense concentrations of topological modifications in specific portions of the graph rather than the desired relatively even distribution. To alleviate this, the SPIDRS algorithm was modified to consider both the external face of the graph for added freedom of movement and graphical operations that are parameterized by the current geometry of the graph. Additionally, a study was conducted to determine the genetic algorithm parameters that result in better-performing and more diverse topologies. After implementing these changes, SPIDRS was once again applied to the cantilevered frame design problem, this time showing good comparison with the performance of a Michell truss across a range of normalized mass values. Considering the tensile inverter design problem, the frontier of solutions generated by SPIDRS outperforms that associated with the turtle graphics algorithm and closely agrees with the ideal solution. Finally, SPIDRS demonstrated an effectiveness in identifying multifunctional solutions when applied to two versions of the kinematic rectifier design problem.

Given the novelty of this graph-based approach, there are various avenues that could be pursued in future research. To this point, the edges of a given graph generated by SPIDRS have been assumed to be represented by straight lines. Recalling that a graph simply represents the connectivity between nodes, there is no reason why edges cannot be defined/created using a specified curvature parameter. In addition to further allowing for performance improvements, this would also enable the use of SPIDRS in applications where the structural design domain is curved. However, the consideration of curved members may begin to violate the half-edge data structure, as faces would no longer be guaranteed to be convex. Additionally, the ability of SPIDRS to assign material to a face in the graph, allowing for the consideration of both continuum and branched domains, is a powerful concept that has not fully been explored. This would allow for L-System topology optimization to be employed in applications where branched topologies alone may be unsuitable.

Another exciting potential direction for this research is the idea of developing a “smart” SPIDRS algorithm using machine learning or some other mechanically-informed decision criteria. SPIDRS currently induces topological modifications based on its interpretation of an L-System encoding that is unique to a set of design variables generated using a genetic algorithm. However, future studies could utilize machine learning to recognize patterns in well-performing designs or include some internal optimization loop based on the predicted mechanics of a design to allow the agent to choose its next command(s) based on which operation(s) would most improve the performance of the current topology. Furthermore, machine learning algorithms could potentially be trained using data from previous SPIDRS-generated designs to predict the performance of arbitrary sets of design variables. This would allow for computationally expensive analyses to be replaced by a simple function call to a trained machine learning algorithm, greatly expediting the optimization process.

7.3 Three-Dimensional L-System Topology Optimization Using Graph-Based Interpretation

Chapter 4 explored the ability of extending the parameterized L-System/SPIDRS framework to consider 3-D structural design domains, an area that conventional topology optimization methodologies are capable in but had not been demonstrated using an L-System. It demonstrated that this extension required no additional design variables, only the addition of two characters to the alphabet associated with the parameterized L-System. Furthermore, it was shown that both the half-edge data structure and SPIDRS graphical operations remain valid when considering a 3-D graph. Next, inspired by graph theory and the definition of a face in a graph, a Create Interfacial Edge graphical operation that allows for the creation of edges between faces defined to be in different planes in the structural design space was discussed. While the implementation of this graphical operation has several minor consequences, namely that the graph operated on by SPIDRS can no longer be guaranteed to be planar, this results in the elegant formation of 3-D topologies.

The first demonstrations of this 3-D L-System topology optimization framework focused on cantilevered frames subjected to various load cases with the goal of maximizing stiffness (either transverse or torsional) while minimizing mass. For a frame subjected to transverse loading, the Pareto frontiers generated by 3-D SPIDRS using two different levels of L-System recursion compare favorably with those generated using two different discretizations in a 3-D SIMP implementation. Furthermore, a comparison of the computation times associated with the two methods shows that SPIDRS was capable of generating a 100-design Pareto frontier in approximately 2.5% of the computation time required for a SIMP implementation with a fine discretization. SPIDRS-generated designs also showed promise when considering a cantilevered frame subjected to torsional loading, with one topology demonstrating symmetry across a plane of the design space despite a lack of symmetry boundary conditions being specified.

A series of single-material 3-D compliant mechanism design problems were then considered, including a tensile inverter, elevator, and cruncher, with the goal of maximizing both the desired displacement and desired force responses as dictated by the problem. For each problem, Pareto frontiers generated by 3-D SPIDRS compared well against the derived ideal solutions, and designs closest to exhibiting a one-to-one displacement and force response feature load paths with geometries similar to designs generated using a conventional topology optimization approach.

Despite the strides made in this work in developing a 3-D L-System topology optimization method, several challenges remain in addition to those described for 2-D problems. Due to the definition of the parameterized L-System and how the Create Interfacial Edge operation is implemented, edges created between faces associated with different planes are assigned a material *a priori*. Thus, for demonstration purposes the designs studies discussed in Chapter 4 considered only single-material structures. Future work should investigate methods that enable the assignment of multiple materials, whether that be through simply employing the Change Material operation to reassign the material of preexisting structural members or modifying the way the L-System encoding is interpreted. Complications also arose from combinations of symmetry boundary conditions and portions of the initial graph forming direct load paths between input and output points, which may have influenced the resulting topologies generated by the algorithm. Future studies must identify a methodology for avoiding this influence, such as treating the entire initial graph similarly to "construction lines".

7.4 Sizing Optimization of Graph-Based L-System-Generated Topologies

Extending upon Chapters 3-4, Chapter 5 considered the implementation of a sizing optimization scheme within the proposed optimization framework to determine the optimal thicknesses of structural members in SPIDRS generated topologies. This allows both for the elimination of unnecessary structural members that form indirect or auxiliary load paths and the redistribution of material to critical members to further optimize the structural performance of a given topology. The proposed sizing optimization process closely resembles the procedure used by the GSM, where the cross-sectional area of predefined, discrete structural members are varied using sensitivity analysis. The need for sensitivity calculations precluded the use of the Abaqus FEA suite, so a linear frame FEA solver written in Python was developed.

The implementation of a sizing optimization scheme within the overall topology optimization framework required the development of a hybrid genetic/gradient-based optimization approach. The genetic topology optimization framework remained consistent with previous chapters in generating sets of design variables that were encoded and then interpreted to create a structural topology, which was then evaluated and optimized using a gradient-based sizing optimizer. This, however, was problematic, as gradient-based methods seek to optimize a single scalar value, while the problems to be considered by the framework were almost exclusively multiobjective. To alleviate this issue, an ε-constraint method was employed to decompose the multiobjective optimization problem into a series of single-objective constrained optimization problems, with the optimal solution to be returned to the genetic algorithm selected using a user-specified decision criteria.

The proposed hybrid optimization framework was first applied to the 2-D cantilevered truss design problem from Chapter 3, with SPIDRS-generated, size-optimized topologies matching the performances of a family of Michell trusses and exceeding the performances of designs obtained using SIMP and GSM implementations. The effect of where in the optimization framework the sizing optimization scheme is implemented was also explored by comparing the performances of topologies with in situ sizing optimization against those of topologies that were considered using only topology optimization, then sized during post processing. Interestingly, a majority of the designs sized only during post processing were able to match the performances of designs that were sized during the entire optimization process. Additionally, the post process sizing optimization technique resulted in a 89% decrease in computation time, though it should be noted that this analysis was able to make use of parallelization techniques, whereas the analysis that featured in situ sizing optimization was not. Nevertheless, this indicated that applying sizing optimization only to the final topology optimization Pareto frontier should be recognized as a computationally efficient alternative to the hybrid optimization approach. Next, the hybrid optimization framework was applied to a 2-D linear tensile inverter problem, with the associated Pareto frontier agreeing well with the ideal solution. Designs generated by the framework feature structural members in tension that were sized such that they could be replaced by a string with a comparable tensile modulus, a result that would not be possible using a SIMP implementation unless the design domain was very finely discretized.

The hybrid optimization framework was then extended to consider the 3-D cantilevered truss design problem from Chapter 4, with SPIDRS-generated, size-optimized topologies exceeding the performances of designs obtained using a SIMP implementation with a fine discretization. The same trends with regards to where in the framework the sizing optimization scheme is implemented for 2-D topologies were also observed here, with a majority of designs that were sized only during post processing comparing well with designs generated by the hybrid optimization scheme while also demonstrating a 94% reduction in required computational time, though the same caveats apply. An additional 3-D cantilevered frame problem was also considered, which removed any symmetry boundary conditions and oriented the applied force 30° off-axis. The resulting Pareto frontier demonstrated the capability of the hybrid optimization framework to produce elegant solutions to 3-D structural design problems without relying on symmetry boundary conditions to easily produce critical load paths.

Future work in this area should focus primarily upon developing a parallelization approach for the hybrid optimization framework. Each topology in a given generation was topologically optimized and sized individually, as opposed to the original topology optimization framework that analyzed entire populations simultaneously. Thus, computation time was a key issue noted when discussing the results generated by the coupled topology/sizing optimization method.

7.5 Applications of Graph-Based L-System Topology Optimization

The final chapter of this work briefly explored several applications of graph-based L-System topology optimization. First, the design of a tailorable stiffness structure was considered with the goal of minimizing the deviation between a specified nonlinear stiffness response and the stiffness response of the structure. In this brief study, two different stiffness responses were specified: one in which the stiffness rate of change increases as the input displacement increases, and one in which the stiffness rate of change decreases as the input displacement increases. For the increasing stiffness rate case, SPIDRS was able to reasonably match the desired stiffness response across the range of loading, with the best design featuring a configuration that deformed such that the tensile load was carried by near-vertical columns. However, for the decreasing stiffness rate case there was noticeable deviation between the best SPIDRS-generated design and the goal response, perhaps due to the amount of loading required to buckle critical structural members and reduce the resistance to the applied load. A need to consider symmetry boundary conditions in the topological equivalency constraint discussed in Appendix D was also observed.

Next, to demonstrate the effectiveness of the proposed framework in considering loads from multiple physical fields, a thermomechanical design problem was discussed. The goal of this problem was to design a unit cell of a periodic structure that is capable of minimizing thermal conductivity while also maximizing both the axial and shear stiffness of the structure. These are conflicting objectives, as the two stiffness objectives to be maximized favor numerous load paths, while the goal of minimizing thermal conductivity calls for designs with minimal load paths that are as long as possible. These trends were observed in SPIDRS-generated designs, where the best insulating design featured structural members constructed in a zigzagged pattern to increase the the length of the load path between the two walls. However, the best axial and shear stiffness designs featured numerous load paths which increased the structure's resistance to loading but also made the structure increasingly conductive. Thus, while simple, this example illustrated the potential of the proposed framework in multiphysical problems.

Finally, a multiphysical design problem was presented with the goal of developing an actuation system capable of morphing an airfoil in supersonic flow. Specifically, SPIDRS was used to generate the topological layout of a 2-D supersonic airfoil that includes a skin, vertical spars, internal stiffeners, and linear internal actuators to maximize the lift-to-drag ratio under cruise conditions while meeting minimum maneuverability requirements. After conducting the optimization process, the four best performing SPIDRS generated configurations were examined in greater detail. Each of these designs featured actuation systems that flatten the camber line during cruise (where a perfectly flat camber line represents the theoretical maximum in supersonic flow) and rotated the spars to decrease/increase the angle of attack and achieve the desired aerodynamic performance during minimum/maximum lift maneuvers. The effect of considering the amount of actuation energy required by a given design to the optimization problem was also discussed.

This is perhaps the largest area of potential future research for the proposed L-System topology optimization framework, as Chapter 6 provided only a brief discussion of preliminary studies into how SPIDRS can influence the design of multifunctional structures. With regards to the tailorable stiffness structure problem discussed, current and future work is focusing on more complex stiffness responses to match, both in tensile and compressive loading, with hopes of beginning to consider non-monotonic stiffness curves, as well as experimental validation of SPIDRS-generated designs. These studies could also consider varying the thickness of structural members by treating the material assignment parameter in the SPIDRS graphical operations as a thickness parameter, further expanding the design space and enabling the generation of better performing structures.

Future work in the area of multifunctional structures design and optimization will require the implementation of accurate multiphysical models and a capability to simulate the behavior of complex smart materials. Studies continuing to analyze the ability of SPIDRS to design a actuation system for a morphing supersonic airfoil will begin to consider transient problems such as matching specified flight paths or aerodynamic performance over a length of time. Additionally, ongoing research is and will continue to use SPIDRS to design actuation systems in rotorblades capable of morphing between multiple different configurations depending on flight conditions. SPIDRS is also being utilized to design fold patterns in origami structures with a view towards developing reconfigurable antennas. These are just a few of the myriad of potential applications that SPIDRS could be utilized in.

Furthermore, future research on potential applications for SPIDRS should not just be limited to structural topology. Recall that SPIDRS generates a graph, which is then converted into a structure by representing edges as frame elements. However, this graph could represent anything from an electrical circuit to the connectivity between nodes in hidden layers or a neural network. It is this flexibility that makes the proposed framework such a powerful and promising design tool.

REFERENCES

[1] A. Newton and H. Gadow, *A Dictionary of Birds*. A. and C. Black, 1896.

[2] M. P. Bendsøe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.

[3] J. D. Deaton and R. V. Grandhi, "A survey of structural and multidisciplinary continuum topology optimization: post 2000," *Structural and Multidisciplinary Optimization*, vol. 49, pp. 1–38, Jan. 2014.

[4] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural Optimization*, vol. 1, pp. 193–202, Dec. 1989.

[5] H. A. Eshenauer and N. Olhoff, *Topology optimization of continuum structures: a review*, vol. 54. American Society of Mechanical Engineers, 2001.

[6] M. H. Kobayashi, "On a biologically inspired topology optimization method," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 3, pp. 787–802, 2010.

[7] B. Bourdin, "Filters in topology optimization," *International Journal for Numerical Methods in Engineering*, vol. 50, pp. 2143–2158, Mar. 2001.

[8] J. K. Guest, J. H. Prévost, and T. Belytschko, "Achieving minimum length scale in topology optimization using nodal design variables and projection functions," *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 238–254, Sept. 2004.

[9] O. Sigmund, "Morphology-based black and white filters for topology optimization," *Structural and Multidisciplinary Optimization*, vol. 33, pp. 401–424, Feb. 2007.

[10] J. K. Guest, A. Asadpoure, and S.-H. Ha, "Eliminating beta-continuation from Heaviside projection and density filter algorithms," *Structural and Multidisciplinary Optimization*, vol. 44, pp. 443–453, Oct. 2011.

[11] A. Kawamoto, T. Matsumori, S. Yamasaki, T. Nomura, T. Kondoh, and S. Nishiwaki, "Heaviside projection based topology optimization by a PDE-filtered scalar function," *Structural and Multidisciplinary Optimization*, vol. 44, pp. 19–24, July 2011.

[12] O. Sigmund, "On the design of compliant mechanisms using topology optimization," *Mechanics of Structures and Machines*, vol. 25, pp. 493–524, Jan. 1997.

[13] O. Sigmund and J. Petersson, "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Structural Optimization*, vol. 16, pp. 68–75, Aug. 1998.

[14] B. S. Lazarov and O. Sigmund, "Filters in topology optimization based on Helmholtz-type differential equations," *International Journal for Numerical Methods in Engineering*, vol. 86, pp. 765–781, May 2011.

[15] J. K. Guest, "Imposing maximum length scale in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 37, pp. 463–473, Feb. 2009.

[16] B. S. Lazarov and F. Wang, "Maximum length scale in density based topology optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 318, pp. 826–844, May 2017.

[17] F. Wang, B. S. Lazarov, and O. Sigmund, "On projection methods, convergence and robust formulations in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 43, pp. 767–784, June 2011.

[18] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, "Efficient topology optimization in MATLAB using 88 lines of code," *Structural and Multidisciplinary Optimization*, vol. 43, pp. 1–16, Jan. 2011.

[19] J. Sethian and A. Wiegmann, "Structural boundary design via level set and immersed interface methods," *Journal of Computational Physics*, vol. 163, pp. 489–528, Sept. 2000.

[20] M. Y. Wang, X. Wang, and D. Guo, “A level set method for structural topology optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 227–246, Jan. 2003.

[21] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, “Level-set methods for structural topology optimization: a review,” *Structural and Multidisciplinary Optimization*, vol. 48, pp. 437–472, Sept. 2013.

[22] G. Allaire, F. Jouve, and A.-M. Toader, “Structural optimization using sensitivity analysis and a level-set method,” *Journal of Computational Physics*, vol. 194, pp. 363–393, Feb. 2004.

[23] S. Amstutz and H. Andrä, “A new algorithm for topology optimization using a level-set method,” *Journal of Computational Physics*, vol. 216, pp. 573–588, Aug. 2006.

[24] X. Xing, P. Wei, and M. Y. Wang, “A finite element-based level set method for structural optimization,” *International Journal for Numerical Methods in Engineering*, pp. n/a–n/a, 2009.

[25] N. van Dijk, M. Langelaar, and F. Keulen, “Explicit level-set-based topology optimization using an exact Heaviside function and consistent sensitivity analysis,” *International Journal for Numerical Methods in Engineering*, vol. 91, pp. 67–97, July 2012.

[26] M. de Ruiter and F. van Keulen, “Topology optimization using a topology description function,” *Structural and Multidisciplinary Optimization*, vol. 26, pp. 406–416, Apr. 2004.

[27] S. Wang and M. Y. Wang, “Radial basis functions and level set method for structural topology optimization,” *International Journal for Numerical Methods in Engineering*, vol. 65, pp. 2060–2090, Mar. 2006.

[28] Z. Luo, L. Tong, M. Y. Wang, and S. Wang, “Shape and topology optimization of compliant mechanisms using a parameterization level set method,” *Journal of Computational Physics*, vol. 227, pp. 680–705, Nov. 2007.

[29] S. Kreissl, G. Pingen, and K. Maute, “An explicit level set approach for generalized shape optimization of fluids with the lattice Boltzmann method,” *International Journal for Numerical Methods in Fluids*, vol. 65, pp. 496–519, Feb. 2011.

[30] A. A. Gomes and A. Suleman, “Application of spectral level set methodology in topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 31, pp. 430–443, June 2006.

[31] S.-H. Ha and S. Cho, “Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh,” *Computers & Structures*, vol. 86, pp. 1447–1455, July 2008.

[32] G. Allaire, C. Dapogny, and P. Frey, “Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh,” *Comptes Rendus Mathematique*, vol. 349, pp. 999–1003, Sept. 2011.

[33] Z. Luo, M. Y. Wang, S. Wang, and P. Wei, “A level set-based parameterization method for structural shape and topology optimization,” *International Journal for Numerical Methods in Engineering*, vol. 76, pp. 1–26, Oct. 2008.

[34] B. Zhu, X. Zhang, and S. Fatikow, “Filter the shape sensitivity in level set-based topology optimization methods,” *Structural and Multidisciplinary Optimization*, vol. 51, pp. 1035–1049, May 2015.

[35] Luo Z., Zhang N., Gao W., and Ma H., “Structural shape and topology optimization using a meshless Galerkin level set method,” *International Journal for Numerical Methods in Engineering*, vol. 90, pp. 369–389, Mar. 2012.

[36] Y. M. Xie and G. P. Steven, “Basic evolutionary structural optimization,” in *Evolutionary Structural Optimization*, pp. 12–29, Springer, London, 1997.

[37] Q. Liang, Y. Xie, and G. Steven, “Optimal topology selection of continuum structures with displacement constraints,” *Computers & Structures*, vol. 77, pp. 635–644, Aug. 2000.

[38] O. Querin, G. Steven, and Y. Xie, “Evolutionary structural optimisation (ESO) using a bidirectional algorithm,” *Engineering Computations*, vol. 15, pp. 1031–1048, Dec. 1998.

[39] O. M. Querin, V. Young, G. P. Steven, and Y. M. Xie, “Computational effiency and validation of bi-directional evolutionary structural optimisation,” *Comput. Methods Appl. Mech. Engrg.*, p. 15, 2000.

[40] M. Zhou and G. Rozvany, “On the validity of ESO type methods in topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 21, pp. 80–83, Mar. 2001.

[41] G. I. N. Rozvany, “A critical review of established methods of structural topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 37, pp. 217–237, Jan. 2009.

[42] A. Ben-Tal and M. P. Bendsøe, “A new method for optimal truss topology design,” *SIAM Journal on Optimization*, vol. 3, pp. 322–358, May 1993.

[43] M. P. Bendsøe, A. Ben-Tal, and J. Zowe, “Optimization methods for truss geometry and topology design,” *Structural Optimization*, vol. 7, pp. 141–159, Apr. 1994.

[44] J. Lee, C. Mueller, and C. Fivet, “Automatic generation of diverse equilibrium structures through shape grammars and graphic statics,” *International Journal of Space Structures*, vol. 31, pp. 147–164, June 2016.

[45] T. Hagishita and M. Ohsaki, “Topology optimization of trusses by growing ground structure method,” *Structural and Multidisciplinary Optimization*, vol. 37, pp. 377–393, Jan. 2009.

[46] T.-Y. Chen and S.-C. Wu, “Multiobjective optimal topology design of structures,” *Computational Mechanics*, vol. 21, pp. 483–492, June 1998.

[47] J. L. Cohon, *Multiobjective programming and planning*. Mineola, N.Y: Dover Publications, 2003.

[48] H. Tamaki, H. Kita, and S. Kobayashi, “Multi-objective optimization by genetic algorithms: a review,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 517–522, May 1996.

[49] O. Sigmund, “On the usefulness of non-gradient approaches in topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 43, pp. 589–596, May 2011.

[50] K. Deb, “An introduction to genetic algorithms,” *Sadhana*, vol. 24, pp. 293–315, Aug. 1999.

[51] I. Turevsky and K. Suresh, “Efficient generation of pareto-optimal topologies for compliance optimization,” *International Journal for Numerical Methods in Engineering*, vol. 87, pp. 1207–1228, Sept. 2011.

[52] B. Zhu, X. Zhang, and S. Fatikow, “A multi-objective method of hinge-free compliant mechanism optimization,” *Structural and Multidisciplinary Optimization*, vol. 49, pp. 431–440, Mar. 2014.

[53] J. Lin, Z. Luo, and L. Tong, “A new multi-objective programming scheme for topology optimization of compliant mechanisms,” *Structural and Multidisciplinary Optimization*, vol. 40, pp. 241–255, Jan. 2010.

[54] I. Das and J. E. Dennis, “A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems,” *Structural Optimization*, vol. 14, pp. 63–69, Aug. 1997.

[55] A. Messac, G. J. Sundararaj, R. V. Tappeta, and J. E. Renaud, “Ability of objective functions to generate points on nonconvex pareto frontiers,” *AIAA Journal*, vol. 38, pp. 1084–1091, June 2000.

[56] A. Messac and A. Ismail-Yahaya, “Required relationship between objective function and pareto frontier orders: Practical implications,” *AIAA Journal*, vol. 39, pp. 2168–2174, Nov. 2001.

[57] W. Hare, J. Nutini, and S. Tesfamariam, “A survey of non-gradient optimization methods in structural engineering,” *Advances in Engineering Software*, vol. 59, pp. 19–28, May 2013.

[58] D. Guirguis, K. Hamza, M. Aly, H. Hegazi, and K. Saitou, “Multi-objective topology optimization of multi-component continuum structures via a Kriging-interpolated level set approach,” *Structural and Multidisciplinary Optimization*, vol. 51, pp. 733–748, Mar. 2015.

[59] M. Yoshimura, K. Shimoyama, T. Misaka, and S. Obayashi, “Topology optimization using a kriging-assisted genetic algorithm with a novel level set representation approach,” pp. 3361–3376, Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece, 2016.

[60] X. Liu, W.-J. Yi, Q. Li, and P.-S. Shen, “Genetic evolutionary structural optimization,” *Journal of Constructional Steel Research*, vol. 64, pp. 305–311, Mar. 2008.

[61] Z. H. Zuo, Y. M. Xie, and X. Huang, “Combining genetic algorithms with BESO for topology optimization,” *Structural and Multidisciplinary Optimization*, vol. 38, pp. 511–523, June 2009.

[62] J. Prasad and A. R. Diaz, “Synthesis of bistable periodic structures using topology optimization and a genetic algorithm,” *Journal of Mechanical Design*, vol. 128, pp. 1298–1306, Dec. 2005.

[63] H. Shin, A. Todoroki, and Y. Hirano, “Elite-initial population for efficient topology optimization using multi-objective genetic algorithms,” *International Journal of Aeronautical and Space Sciences*, vol. 14, pp. 324–333, Dec. 2013.

[64] T. Li, X. Sun, Z. Lu, and Y. Wu, “A novel multiobjective optimization method based on sensitivity analysis,” *Mathematical Problems in Engineering*, vol. 2016, pp. 1–12, 2016.

[65] S. M. D. Institute, “Lightweight steel wheel,” tech. rep., 2013.

[66] A. Kaldate, D. Thurston, H. Emamipour, and M. Rood, “Engineering parameter selection for design optimization during preliminary design,” *Journal of Engineering Design*, vol. 17, pp. 291–310, Aug. 2006.

[67] P. Fantini, “Effective multiobjective MDO for conceptual design - an aircraft design perspective,” 2007.

[68] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist Non-dominated Sorting Genetic Algorithm for multi-objective optimization: NSGA-II,” in *Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18âĂŞ20,*

2000 Proceedings (M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, eds.), pp. 849–858, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000.

[69] B. Stanford, P. Beran, and M. Kobayashi, “Aeroelastic optimization of flapping wing venation: A cellular division approach,” *AIAA Journal*, vol. 50, pp. 938–951, Apr. 2012.

[70] B. Stanford, P. Beran, and M. Kobayashi, “Simultaneous topology optimization of membrane wings and their compliant flapping mechanisms,” *AIAA Journal*, vol. 51, no. 6, pp. 1431–1441, 2013.

[71] M. Kobayashi, R. Kolonay, G. Reich, A. LeBon, and H.-T. Pedro, “On a cellular division model for multi-disciplinary optimization,” American Institute of Aeronautics and Astronautics, Apr. 2010.

[72] R. Kolonay and M. Kobayashi, “Topology, shape, and sizing optimization of aircraft lifting surfaces using a cellular division method,” American Institute of Aeronautics and Astronautics, Sept. 2010.

[73] R. M. Kolonay and M. H. Kobayashi, “Optimization of aircraft lifting surfaces using a cellular division method,” *Journal of Aircraft*, vol. 52, no. 6, pp. 2051–2063, 2015.

[74] H.-T. C. Pedro and M. H. Kobayashi, “On a cellular division method for topology optimization,” *International Journal for Numerical Methods in Engineering*, vol. 88, no. 11, pp. 1175–1197, 2011.

[75] H. Abelson and A. A. DiSessa, *Turtle geometry: The computer as a medium for exploring mathematics*. MIT Press, 1986.

[76] A. Roth-Nebelsick, D. Uhl, V. Mosbrugger, and H. Kerp, “Evolution and function of leaf venation architecture: a review,” *Annals of Botany*, vol. 87, no. 5, pp. 553–566, 2001.

[77] J. Stark, J. Bonacum, J. Remsen, and R. DeSalle, “The evolution and development of dipteran wing veins: A systematic approach,” *Annual Review of Entomology*, vol. 44, pp. 97–129, Jan. 1999.

[78] E. R. Weibel, “Morphometrics of the lung,” *Handbook of Physiology. Respiration*, vol. 1, pp. 285–307, 1964.

[79] A. F. Miguel, “Lungs as a natural porous media: architecture, airflow characteristics and transport of suspended particles,” in *Heat and Mass Transfer in Porous Media*, pp. 115–137, Springer, 2012.

[80] T. F. Sherman, “On connecting large vessels to small. the meaning of Murray’s law.,” *The Journal of General Physiology*, vol. 78, no. 4, pp. 431–453, 1981.

[81] M. Labarbera, “The design of fluid transport systems: a comparative perspective,” in *Flow-Dependent Regulation of Vascular Function*, pp. 3–27, Springer, 1995.

[82] H. T. Pedro, M. H. Kobayashi, C. F. Coimbra, and A. K. Da Silva, “Effectiveness of complex design through an evolutionary approach,” *Journal of Thermophysics and Heat Transfer*, vol. 22, pp. 115–118, Jan. 2008.

[83] M. H. Kobayashi, H.-T. C. Pedro, C. F. Coimbra, and A. K. Da Silva, “Formal evolutionary development of low-entropy dendritic thermal systems,” *Journal of Thermophysics and Heat Transfer*, vol. 23, pp. 822–827, Oct. 2009.

[84] D. J. Hartl, G. W. Reich, and P. S. Beran, “Additive topological optimization of muscular-skeletal structures via genetic L-system programming,” in *24th AIAA/AHS Adaptive Structures Conference*, p. 1569, 2016.

[85] D. J. Hartl, B. R. Bielefeldt, G. W. Reich, and P. S. Beran, “Multi-fidelity analysis and experimental characterization of muscular-skeletal structures optimized via genetic programming,” in *25th AIAA/AHS Adaptive Structures Conference*, p. 1442, 2017.

[86] A. Lindenmayer, “Mathematical models for cellular interactions in development I. filaments with one-sided inputs,” *Journal of theoretical biology*, vol. 18, no. 3, pp. 280–299, 1968.

[87] P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.

[88] A. R. Smith, “Plants, fractals, and formal languages,” *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, pp. 1–10, 1984.

[89] P. Prusinkiewicz, “Graphical applications of L-systems,” in *Proceedings of graphics interface*, vol. 86, pp. 247–253, 1986.

[90] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan, “Development models of herbaceous plants for computer imagery purposes,” in *ACM SIGGRAPH Computer Graphics*, vol. 22, pp. 141–150, ACM, 1988.

[91] G. Ochoa, “On genetic algorithms and Lindenmayer systems,” in *Parallel Problem Solving from Nature-PPSN V*, pp. 335–344, Springer, 1998.

[92] J. McCormack, “Grammar based music composition,” *Complex systems*, vol. 96, pp. 321–336, 1996.

[93] S. Manousakis, “Musical L-systems,” *Koninklijk Conservatorium, The Hague (Master Thesis)*, 2006.

[94] N. Chomsky, “On certain formal properties of grammars,” *Information and Control*, vol. 2, pp. 137–167, June 1959.

[95] F.-A. Fortin, F.-M. D. Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, “DEAP: Evolutionary Algorithms Made Easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, July 2012.

[96] D. S. of America Corp., “Abaqus analysis user’s guide,” 2007.

[97] M. Kobayashi, H. C. Pedro, R. Kolonay, and G. Reich, “On a cellular division method for aircraft structural design,” *Aeronautical Journal*, vol. 113, no. 1150, pp. 821–831, 2009.

[98] L. L. Howell, *Compliant mechanisms*. John Wiley & Sons, 2001.

[99] Z. Li, L. Zuo, J. Kuang, and G. Luhrs, “Energy-harvesting shock absorber with a mechanical motion rectifier,” *Smart Materials and Structures*, vol. 22, p. 025008, Feb. 2013.

[100] G. I. N. Rozvany, M. P. Bendsøe, and U. Kirsch, “Layout optimization of structures,” *Applied Mechanics Reviews*, vol. 48, pp. 41–119, Feb. 1995.

[101] A. G. M. Michell, “LVIII. the limits of economy of material in frame-structures,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 8, no. 47, pp. 589–597, 1904.

[102] M. Otomori, T. Yamada, K. Izui, and S. Nishiwaki, “MATLAB code for a level set-based topology optimization method using a reaction diffusion equation,” *Structural and Multidisciplinary Optimization*, vol. 51, pp. 1159–1172, May 2015.

[103] Q. Li, G. P. Steven, and Y. M. Xie, “On equivalence between stress criterion and stiffness criterion in evolutionary structural optimization,” *Structural optimization*, vol. 18, pp. 67–73, Aug. 1999.

[104] D. Tcherniak and O. Sigmund, “A web-based topology optimization program,” *Structural and multidisciplinary optimization*, vol. 22, no. 3, pp. 179–187, 2001.

[105] B. Collette, D. Falck, and Y. v. Havre, *FreeCAD: solid modeling with the power of Python*. Packt open source - community experience distilled, Birmingham Mumbai: Packt Publishing, 2012.

[106] M. W. Barclift and C. B. Williams, “Examining variability in the mechanical properties of parts manufactured via polyjet direct 3d printing,” in *International Solid Freeform Fabrication Symposium*, pp. 6–8, University of Texas at Austin Austin, Texas, 2012.

[107] D. Inoyama, B. P. Sanders, and J. J. Joo, “Topology optimization approach for the determination of the multiple-configuration morphing wing structure,” *Journal of Aircraft*, vol. 45, pp. 1853–1863, Nov. 2008.

[108] E. K. Lloyd, J. A. Bondy, and U. S. R. Murty, “Graph theory with applications,” *The Mathematical Gazette*, vol. 62, p. 63, Mar. 1978.

[109] R. Diestel, “Graph theory,” *Graduate Texts in Mathematics*, vol. 173, no. 451, 2010.

[110] A. Lindenmayer, "Adding continuous components to L-systems," in *L Systems*, Lecture Notes in Computer Science, pp. 53–68, Springer, Berlin, Heidelberg, 1974.

[111] D. Muller and F. Preparata, "Finding the intersection of two convex polyhedra," *Theoretical Computer Science*, vol. 7, no. 2, pp. 217–236, 1978.

[112] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, "Computational geometry," in *Computational Geometry: Algorithms and Applications* (M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, eds.), pp. 1–17, Berlin, Heidelberg: Springer, 1997.

[113] W. A. Sutherland, *Introduction to metric and topological spaces*. Oxford University Press, 2009.

[114] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, Reading, Massachusetts, 1989.

[115] K. A. De Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," in *Parallel Problem Solving from Nature* (G. Goos, J. Hartmanis, D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. SeegmÃijller, J. Stoer, N. Wirth, H.-P. Schwefel, and R. Männer, eds.), vol. 496, pp. 38–47, Berlin, Heidelberg: Springer Berlin Heidelberg, 1991.

[116] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 656–667, Apr. 1994.

[117] G. Taguchi and S. Konishi, *Orthogonal arrays and linear graphs*. American Supplier Institute, Inc., 1986.

[118] A. Díaz and R. Lipton, "Optimal material layout for 3D elastic structures," *Structural optimization*, vol. 13, pp. 60–64, Feb. 1997.

[119] K. Liu and A. Tovar, "An efficient 3D topology optimization code written in MATLAB," *Structural and Multidisciplinary Optimization*, vol. 50, pp. 1175–1196, Dec. 2014.

[120] Z. H. Zuo and Y. M. Xie, “A simple and compact Python code for complex 3D topology optimization,” *Advances in Engineering Software*, vol. 85, pp. 1–11, July 2015.

[121] P. D. Dunning, B. K. Stanford, and H. A. Kim, “Coupled aerostructural topology optimization using a level set method for 3D aircraft wings,” *Structural and Multidisciplinary Optimization*, vol. 51, pp. 1113–1132, May 2015.

[122] J. Liu and Y.-S. Ma, “3D level-set topology optimization: a machining feature-based approach,” *Structural and Multidisciplinary Optimization*, vol. 52, pp. 563–582, Sept. 2015.

[123] S. Deng and K. Suresh, “Multi-constrained 3D topology optimization via augmented topological level-set,” *Computers & Structures*, vol. 170, pp. 1–12, July 2016.

[124] D. S. of America Corp., “Tosca structure user’s guide,” 2014.

[125] R. Ansola, E. Veguería, A. Maturana, and J. Canales, “3D compliant mechanisms synthesis by a finite element addition procedure,” *Finite Elements in Analysis and Design*, vol. 46, pp. 760–769, Sept. 2010.

[126] K. K. Choi and N.-H. Kim, *Structural sensitivity analysis and optimization 1: linear systems*. Springer Science & Business Media, 2006.

[127] A. T. Gaynor, *Topology Optimization Algorithms for Additive Manufacturing*. PhD thesis, The Johns Hopkins University, 2015.

[128] R. D. Cook, D. S. Malkus, and M. E. Plesha, *Concepts and applications of finite element analysis*. New York: Wiley, 3rd ed ed., 1989.

[129] D. Kincaid, D. R. Kincaid, and E. W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*. American Mathematical Soc., 2009. Google-Books-ID: kPDtAp3UZtIC.

[130] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer Science & Business Media, Dec. 2012. Google-Books-ID: bnzjBwAAQBAJ.

[131] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, Dec. 2006. Google-Books-ID: VbHYoSyelFcC.

[132] S. P. Timoshenko and J. M. Gere, *Theory of Elastic Stability*. Courier Corporation, June 2009. Google-Books-ID: rAk_AwAAQBAJ.

[133] K. Tai and J. Prasad, "Target-matching test problem for multiobjective topology optimization using genetic algorithms," *Structural and Multidisciplinary Optimization*, vol. 34, pp. 333–345, Oct. 2007.

[134] N. F. Wang and K. Tai, "Design of grip-and-move manipulators using symmetric path generating compliant mechanisms," *Journal of Mechanical Design*, vol. 130, Nov. 2008.

[135] D. Sharma, K. Deb, and N. N. Kishore, "Towards generating diverse topologies of path tracing compliant mechanisms using a local search based multi-objective genetic algorithm procedure," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 2004–2011, June 2008. ISSN: 1941-0026.

[136] J. D. Deaton and R. V. Grandhi, "Stiffening of restrained thermal structures via topology optimization," *Structural and Multidisciplinary Optimization*, vol. 48, pp. 731–745, Oct. 2013.

[137] J. D. Deaton and R. V. Grandhi, "Stress-based design of thermal structures via topology optimization," *Structural and Multidisciplinary Optimization*, vol. 53, pp. 253–270, Feb. 2016.

[138] S. Deng and K. Suresh, "Stress constrained thermo-elastic topology optimization with varying temperature fields via augmented topological sensitivity based level-set," *Structural and Multidisciplinary Optimization*, vol. 56, pp. 1413–1427, Dec. 2017.

[139] S. Deng and K. Suresh, "Topology optimization under thermo-elastic buckling," *Structural and Multidisciplinary Optimization*, vol. 55, pp. 1759–1772, May 2017.

[140] R. M. Ajaj, C. S. Beaverstock, and M. I. Friswell, "Morphing aircraft: The need for a new design philosophy," *Aerospace Science and Technology*, vol. 49, pp. 154–166, Feb. 2016.

[141] S. Barbarino, O. Bilgen, R. M. Ajaj, M. I. Friswell, and D. J. Inman, “A review of morphing aircraft,” *Journal of Intelligent Material Systems and Structures*, vol. 22, pp. 823–877, June 2011.

[142] J. Sun, Q. Guan, Y. Liu, and J. Leng, “Morphing aircraft based on smart materials and structures: A state-of-the-art review,” *Journal of Intelligent Material Systems and Structures*, vol. 27, pp. 2289–2312, Oct. 2016.

[143] V. P. Galantai, A. Y. N. Sofla, S. A. Meguid, K. T. Tan, and W. K. Yeo, “Bio-inspired wing morphing for unmanned aerial vehicles using intelligent materials,” *International Journal of Mechanics and Materials in Design*, vol. 8, pp. 71–79, Mar. 2012.

[144] R. N. Saunders, D. J. Hartl, J. G. Boyd, and D. C. Lagoudas, “Modeling and development of a twisting wing using inductively heated shape memory alloy actuators,” in *Active and Passive Smart Structures and Integrated Systems 2015*, vol. 9431, p. 94310U, International Society for Optics and Photonics, Apr. 2015.

[145] O. Bilgen and M. I. Friswell, “Piezoceramic composite actuators for a solid-state variable-camber wing,” *Journal of Intelligent Material Systems and Structures*, vol. 25, pp. 806–817, May 2014.

[146] G. Molinari, M. Quack, A. F. Arrieta, M. Morari, and P. Ermanni, “Design, realization and structural testing of a compliant adaptable wing,” *Smart Materials and Structures*, vol. 24, p. 105027, Oct. 2015.

[147] J. D. Hodson, A. P. Christopherson, J. D. Deaton, A. M. Pankonien, G. W. Reich, and P. S. Beran, “Aeroelastic topology optimization of a morphing airfoil in supersonic flow using evolutionary design,” in *AIAA Scitech 2019 Forum*, (San Diego, California), American Institute of Aeronautics and Astronautics, Jan. 2019.

[148] A. Garrett, “inspyred (version 1.0.1),” 2012.

[149] M.J. Lighthill, “Oscillating airfoils at high mach number,” *Journal of the Aeronautical Sciences*, vol. 20, pp. 402–406, June 1953.

[150] E. H. Dowell, *A Modern Course in Aeroelasticity*, vol. 217 of *Solid Mechanics and Its Applications*. Cham: Springer International Publishing, 2015.

[151] S. Gillies, “Shapely: manipulation and analysis of geometric objects,” 2007.

[152] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[153] G. R. Bradski and A. Kaehler, *Learning OpenCV: computer vision with the OpenCV library*. Software that sees, Beijing: O’Reilly, 1. ed., [nachdr.] ed., 2011. OCLC: 838472784.

APPENDIX A

GEOMETRY-BASED INTERPRETATION: BOUNDARY CONSTRAINT

As discussed in Section 2.2, the creation of structures based on L-System encoded topologies relies on the spreading of branched structures throughout a predetermined bounded domain. Segments that would result in the violation of these boundaries are then actively constrained to only reach the boundaries but not violate them. The following section provides a brief overview of this constraint.

Consider a turtle (cf., Section 2.1.2) located at the coordinate (x_1, y_1) inside the design domain, which is bounded by a series of segments that include the line between (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}) as shown in Figure A.1. Now assume that the L-System generated instructions cause the turtle to create a line segment of length L, corresponding to a new turtle location of (x_2, y_2). The first step

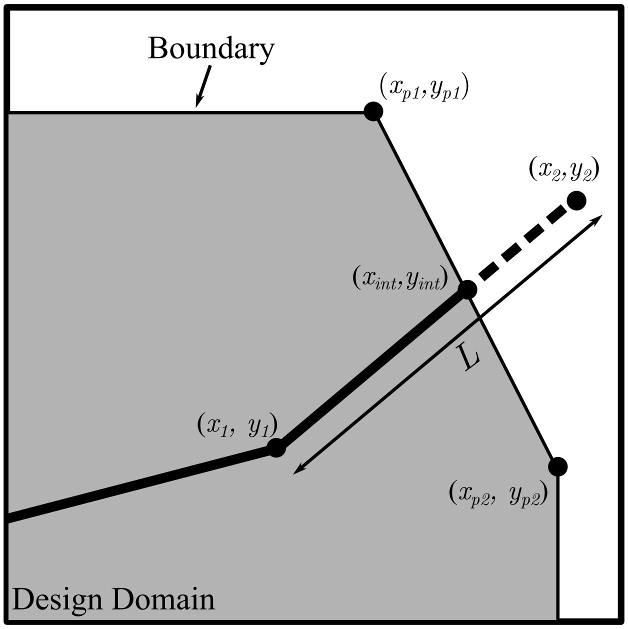


Figure A.1: Illustration of boundary constraint associated with the creation of L-System-generated topologies.

in this boundary constraint function is to check whether the turtle's new location remains inside the boundary, which is performed using the ray casting algorithm; if the turtle remains within the boundary, the constraint is not violated and the line segment is created as instructed by the L-System. However, suppose that (x_2, y_2) is found to lie outside the domain boundary; given that the segment is allowed to reach the boundary but not cross it, the intersection point of the boundary and the proposed line segment must be determined. It can be shown that for a line defined by points (x_1, y_1) and (x_2, y_2) and another line defined by points (x_3, y_3) and (x_4, y_4), the intersection coordinates (P_x, P_y) between those two lines can be determined by

$$\begin{aligned}(P_x, P_y) = \Bigg(&\frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}, \\ &\frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \Bigg).\end{aligned} \tag{A.1}$$

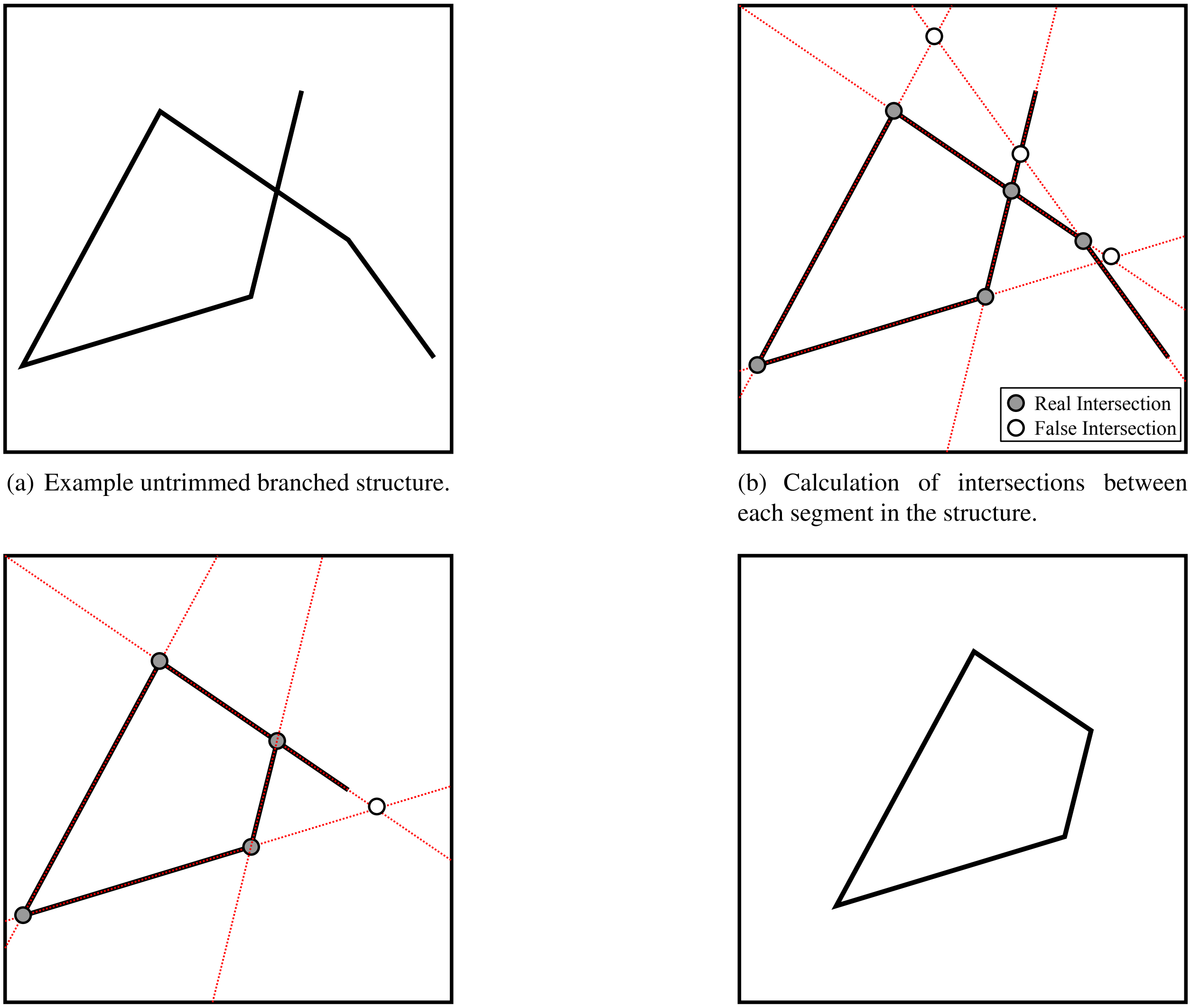
Substituting in the start and end points for both the proposed line segment and boundary segment that would be crossed obtains the coordinate (x_{int}, y_{int}), which is the turtle's actual new location as allowed by the constraint.

APPENDIX B

GEOMETRY-BASED INTERPRETATION: TRIMMING ALGORITHM

One clear disadvantage of the branched L-System/turtle approach to topological generation is the necessary existence of line segments with free ends, which do not have the capability of transferring loads into/out of the rest of the structure. Free-ended segments thus have no structural significance, artificially inflate structural mass measurement, and require increased computational run-times due to the existence of the elements that comprise them. Therefore, the implementation of a trimming algorithm to remove free-ended segments is fundamental to the approach taken in this work.

Figure B.1 illustrates the trimming process. The algorithm makes use of Abaqus' "auto-trim" feature, although the underlying principles can be translated to any CAD tool. First, the start- and end-points of each line segment in the structure are obtained and stored. Second, the algorithm loops through each line segment and calculates the point of intersection between that segment and every other segment using Equation A.1. However, as depicted in Figure B.1b, the calculation of each possible intersection is complicated by the possibility of that intersection point not lying within the actual domain of each segment (i.e., the segments intersect at some point extrapolated from one or both). The algorithm then loops through each line segment, evaluating whether the end-point of the segment corresponds to either the beginning of another segment or an intersection point between two segments. If the line segment is deemed to be a free-ended segment, Abaqus removes removes the segment up to the next intersection point or the entire branch as a whole. Note in Figure B.1c that a previously analyzed non-free-ended branch may become a free-ended segment during the iterative execution of the trimming process; the algorithm will iterate over each line segment until the structure remains unchanged (cf. Figure B.1d). The result is a structure in which all line segments are capable of transferring loads throughout the structure.



(a) Example untrimmed branched structure.

(b) Calculation of intersections between each segment in the structure.

(c) Structure after first iteration of trimming.

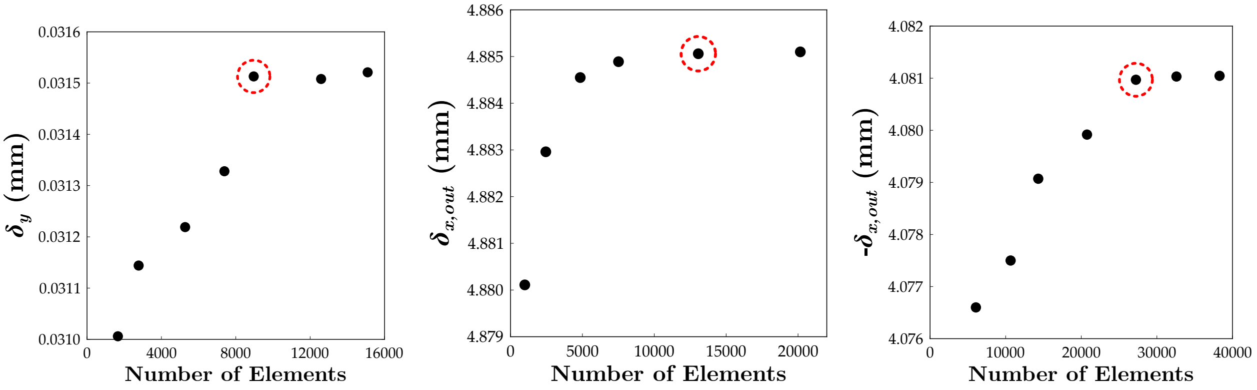
(d) Final trimmed structure.

Figure B.1: Illustration of the trimming algorithm on a simple structure.

APPENDIX C

VALIDATION OF GEOMETRY-BASED INTERPRETATION TOPOLOGIES: MESH DENSITY STUDIES

To obtain continuum-based FEA results that satisfactorily balance computational resources with the accuracy needed to represent a “numerical validation”, a mesh density study was conducted on the models associated with each design problem. For a specified test case (free displacement inversion and “pushing” displacement for the tensile inverter and kinematic rectifier, respectively), the continuum-based FEA model was analyzed with progressively smaller global element sizes until the resulting output associated with that test case converged. Due to the highly nonlinear responses of these models and their effect on the resulting output, each study was conducted while considering only linear geometric effects. The results of these studies are illustrated in Figure C.1.



(a) Cantilevered frame problem. (b) Tensile inverter problem. (c) Kinematic rectifier problem.

Figure C.1: Results of the mesh density studies associated with the continuum-based FEA model for each design problem. Selected mesh for each design problem is denoted by the dashed circle.

APPENDIX D

GRAPH-BASED INTERPRETATION: OVERLAP CONSTRAINT

A recognized drawback to the graph-based interpretation of L-System encodings proposed in Chapter 3 (as well as the geometry-based interpretation method presented in Chapter 2, although there it is less noticeable) is the presence of material overlap when converting the final topology of a graph into a structural topology. As stated in Chapter 3.1, a graph is made up of edges, which are represented in a diagram as a series of lines. Once the final topology of the graph has been generated, it is converted into a structural topology as shown in Figure D.1 by representing each edge as a structural member with some cross-section and made up of some material as specified by the material set contained in the graph. Now, consider a set of edges sufficiently close to one another in a topology, such as the three edges circled in Figure D.1. In the computational analysis proposed in the optimization framework (cf., Chapter 3.2.3), each of these structural members (i.e., edges) is approximated as a 2-D beam element, and the response of the structure is predicted under the assumption that these edges are distinct beams with in-plane thickness t regardless of how much space exists between them. However, if physically fabricated, these edges will overlap one another during the conversion process and merge together into a single structural member whose response will not be accurately captured by FEA analysis. This necessitates a constraint to ensure that material overlap is minimized.

In lieu of more rigorous mathematical checks, the proposed constraint takes a volumetric approach. Specifically, this constraint ensures that the calculated volume of the structural topology using the beam-based FEA model is equivalent the "true" volume of the topology when considering the overlapping of structural members. The volume of this topology is calculated using a Python package known as *Shapely*, which allows for the manipulation and analysis of planar geometric objects [151]. It is obvious that the volumes of the two models will never be equivalent, as the volume calculated by Shapely takes into account the small material overlap in areas where two

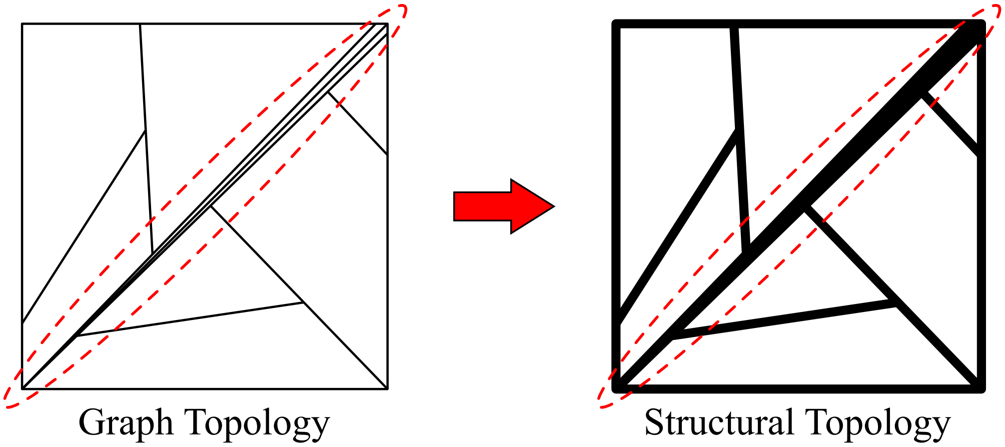


Figure D.1: Example of the topological information of a graph being converted into a structural topology. Note that because of the difference in how edges are represented in the graph versus the structure, there is a noticeable amount of material overlap which would not be captured by FEA analysis.

edges intersect. Thus, we assume that the two volumes are equivalent if they are within a certain percentage of each other (typically 5%, but can vary depending on the problem). Topologies that do not meet this constraint are penalized in the same manner described in Chapter 2.1.3.

APPENDIX E

SIMPLE IMAGE MATCHING USING GRAPH-BASED L-SYSTEM INTERPRETATION

Due to the need for available software licenses and increased computation time incurred when analyzing the physics of engineering design problems, the population sizes and number of generations considered when using the SPIDRS algorithm to interpret L-System encodings is somewhat limited. However, one benefit of SPIDRS is that the output of the algorithm is simply a graph, which enables its application to non-physical functional evaluations. Specifically, when utilizing the Increase Structural Dimensionality operation (cf., Chapter 3.5.2) a SPIDRS-generated graph can be treated as a series of shapes. Given a specified target shape or geometry, one can attempt to match that target by minimizing the error between it and SPIDRS-generated graphs. This relatively inexpensive functional evaluation allows for the consideration of much larger population sizes and increased numbers of generations.

This non-physical optimization process is illustrated in Figure E.1. The bulk of the framework remains the same: NSGA-II continues to drive the optimization process by creating genomes that are interpreted by an L-System, whose encodings are then interpreted using the SPIDRS algorithm.

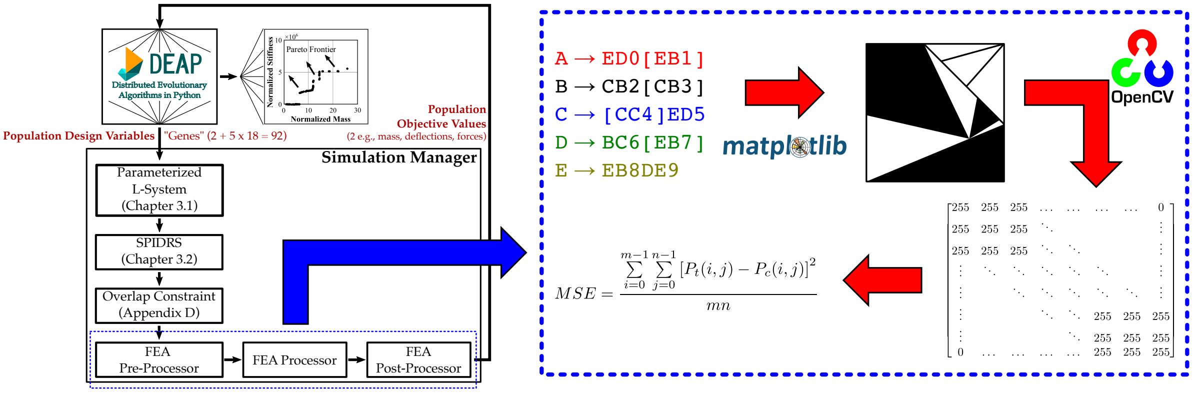

DEAP
Distributed Evolutionary Algorithms in Python
Pareto Frontier
Normalized Stiffness
×10^6
10
5
0
0
10
20
30
Normalized Mass
Population Design Variables
"Genes" (2 + 5 x 18 = 92)
Population Objective Values
(2 e.g., mass, deflections, forces)
Simulation Manager
Parameterized L-System (Chapter 3.1)
SPIDRS (Chapter 3.2)
Overlap Constraint (Appendix D)
FEA Pre-Processor
FEA Processor
FEA Post-Processor
A → ED0[EB1]
B → CB2[CB3]
C → [CC4]ED5
D → BC6[EB7]
E → EB8DE9
matplotlib
OpenCV
$MSE = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[P_t(i,j) - P_c(i,j)]^2}{mn}$
255 255 255 … … … … 0
255 255 255
0 … … … … 255 255 255

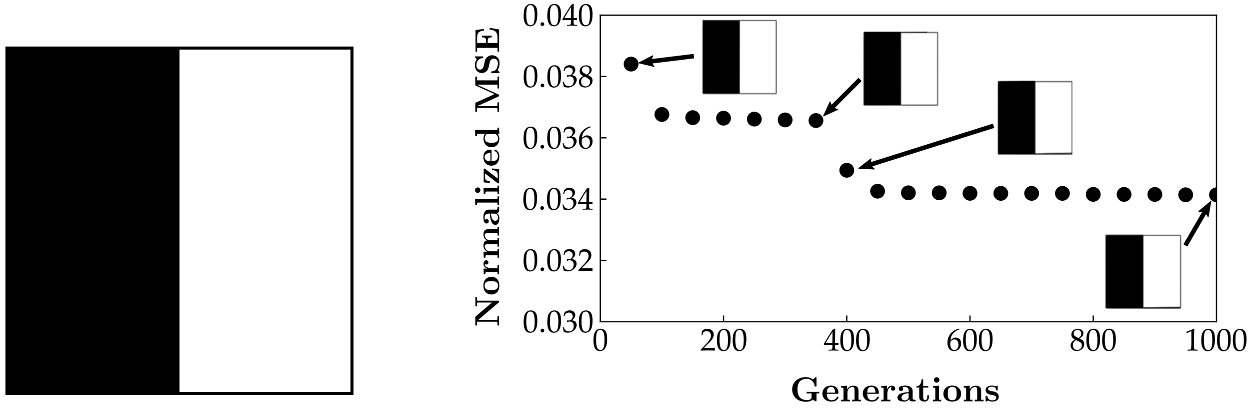
Figure E.1: Flowchart illustrating how the image matching process is implemented withing the optimization framework.

However, thanks to the flexible nature of this framework, structural analysis modules are simply replaced by the process denoted by the dashed box in Figure E.1. Here, the graph generated by SPIDRS is plotted and saved as a figure using the *Matplotlib* Python module [152]. This figure is then converted into an array of size $m \times n$ using the *OpenCV* Python module [153], where m and n are the width and height of the figure in pixels. Each element of this matrix is associated with a specific pixel of the image and is assigned an integer value based on the color intensity that pixel. For convenience, it is assumed that each element will either be assigned a value of 255 (black pixel) or 0 (white pixel), although the ability of SPIDRS to consider a number or materials could allow for other value in future studies. Once the graph has been converted into a matrix, it is compared with the matrix associated with the target image and an error value is calculated. Specifically, a mean squared error (MSE) of the form

$$MSE = \frac{\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [P_t(i,j) - P_c(i,j)]^2}{mn}, \tag{E.1}$$

where P_t is the matrix associated with the target image and P_c is the matrix associated with the SPIDRS-generated graph. The MSE is then returned to the genetic algorithm as an objective function for the associated genome.

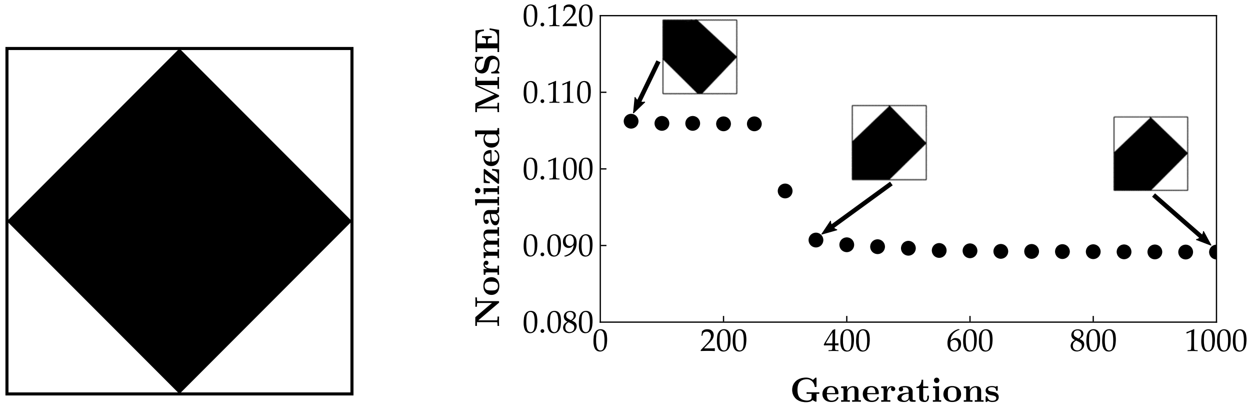
Figures E.2-E.4 show the optimization results for various target images obtained using the framework shown in Figure E.1. Each study considers three L-System recursions and a population size of 1,000 for 1,000 generations. The goal of each optimization is to minimize the error value given in Equation E.1 while also minimizing the volume of the graph (which corresponds to minimizing the number of structural creation operations required by the SPIDRS algorithm). The plots in Figures E.2-E.4 show the evolution of the normalized MSE, or the MSE value from Equation E.1 divided by the largest possible MSE value (which occurs when the image is reversed). From these figures it is clear that SPIDRS is capable of matching extremely simple shapes (Figure E.2), but struggles as target shapes become more complicated (Figures E.3-E.4). However, it should be noted that the worst performing result is still within approximately 12.5% of the target



(a) Target image.

(b) Optimization results.

Figure E.2: Half-and-half image matching optimization problem.

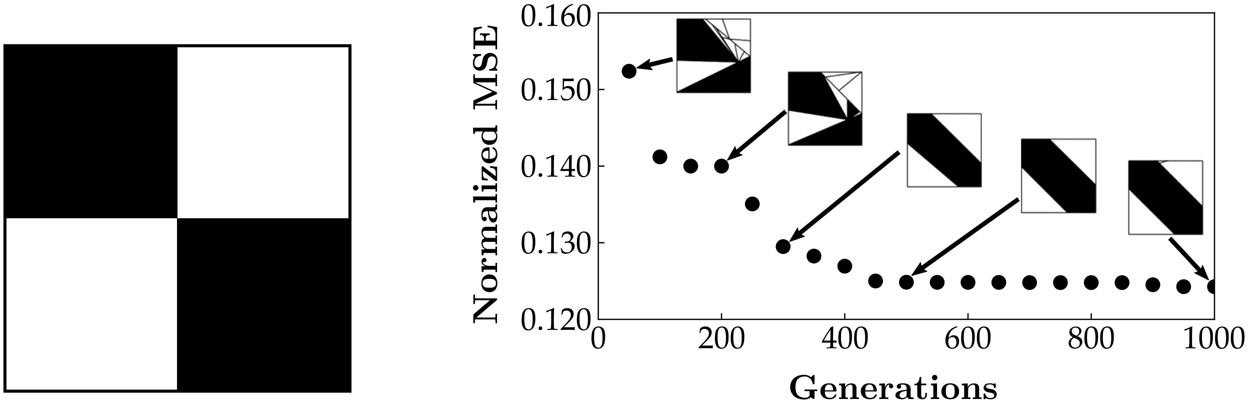


(a) Target image.

(b) Optimization results.

Figure E.3: Diamond image matching optimization problem.

image.



(a) Target image.

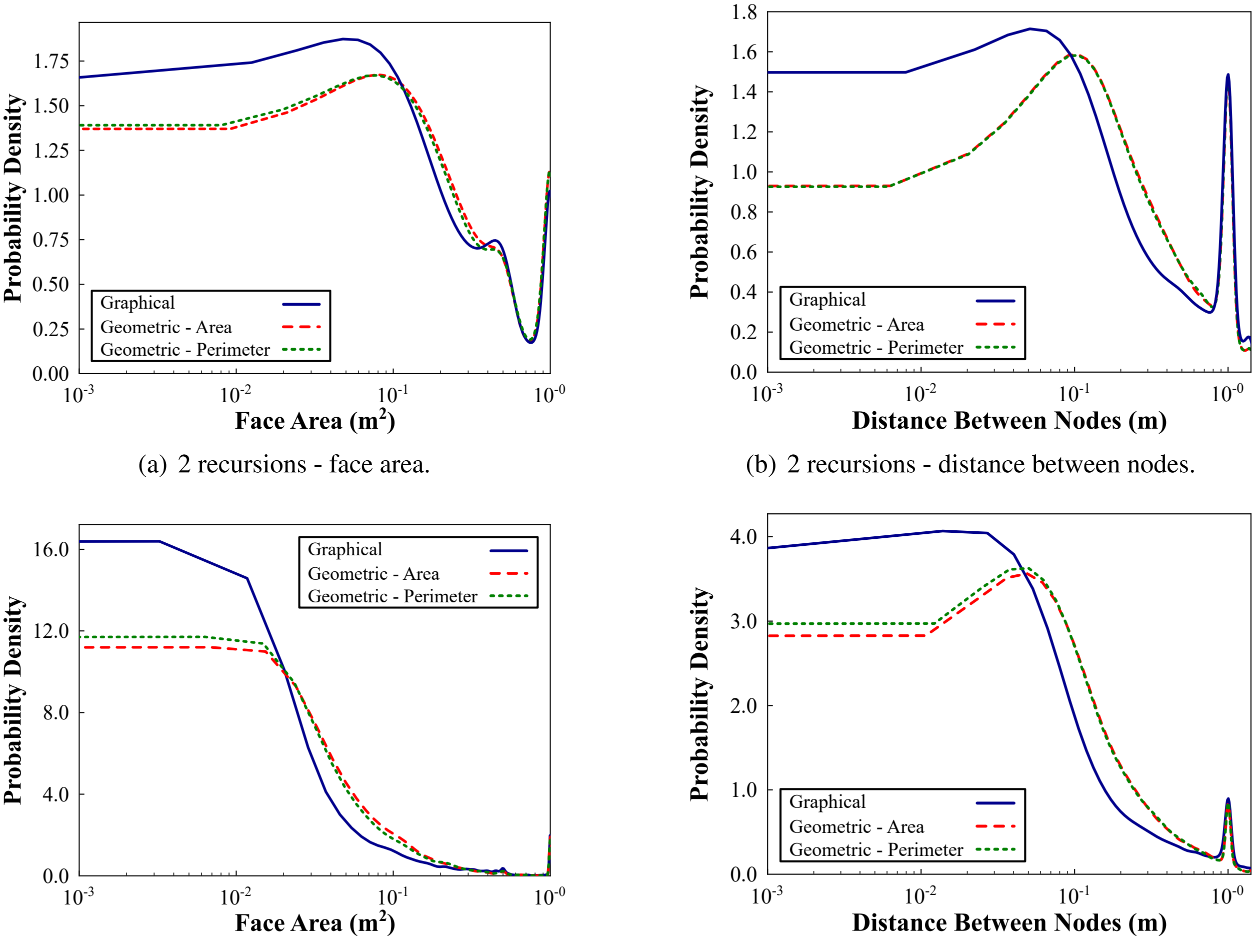
(b) Optimization results.

Figure E.4: Checkerboard image matching optimization problem.

APPENDIX F

GEOMETRIC PARAMETERIZATION OF 3-D TURNING OPERATIONS

As discussed in Chapter 4.1.2, the addition of the Create Interfacial Edge graphical operation to the SPIDRS algorithm allows for the consideration of 3-D structural design domains but renders the graph incompatible with the geometric parameterization of SPIDRS turning operations (cf., Chapter 3.6.2). To maintain computational efficiency while retaining the benefits of geometric parameterization, the parameterization of 3-D turning operations is modified to consider the perimeter of potential faces rather than area. Figure F.1 illustrating the probability density for graphical parameterization, geometric parameterization via area, and geometric parameterization via perimeter of creating faces of a certain area and nodes within a certain distance of each other for both two and three L-System recursions. These plots were made by aggregating the resulting graphs of 500 distinct L-System encodings interpreted using the three parameterization methods. These results indicate that a geometric parameterization based on the perimeter of potential faces compares favorably with the same parameterization based on area and provides a more even distribution of nodes and edges throughout the graph than the graphical parameterization.



(a) 2 recursions - face area.

(b) 2 recursions - distance between nodes.

(c) 3 recursions - face area.

(d) 3 recursions - distance between nodes.

Figure F.1: Probability density plots comparing the effect of graphical parameterization, geometric parameterization via area, and geometric parameterization via perimeter on face area and distance between nodes on 500 SPIDRS-generated graphs.

APPENDIX G

CONVERGENCE BEHAVIOR OF GENETIC ALGORITHMS IN SPIDRS TOPOLOGY OPTIMIZATION STUDIES

One of the recognized drawbacks when utilizing genetic algorithms is the inability to guarantee that a solution has converged to a global optima. As such, genetic algorithms require some user-specified stopping criteria such as generation limits, time limits, or a number of successive generations that do not produce improved results. In this work, the stopping criteria for each genetic topology optimization problem is defined using a specified maximum number of generations. This number, set to 1,000 for cantilevered truss problems and 400 for compliant mechanism problems, was chosen based on previous experience and assumed to be sufficiently high enough to demonstrate the performance of the proposed L-System/SPIDRS topology optimization framework. However, it is possible that, given the complexity of several of the problems considered, the proposed framework may not have converged to a set of solutions and could require additional generations to adequately explore the design space.

To visualize the convergence behavior of the genetic algorithm driving the SPIDRS algorithm in each design problem, a metric that compares the design closest to the utopia point at specified generations to the design closest to the utopia point at the end of the optimization is used. Specifically, after normalizing the objective functions, the design closest utopia point at each generation is determined. Next, the distance in the objective space between this design and the design closest to the utopia point at the end of the optimization is calculated. Note that because it involves solutions from the final generation, this metric is not capable of being used in situ during the optimization process. However, the insights gained using this metric are similar to what one would see if the specified stopping criteria was based on a number of successive generations that do not produce improved results. Designs closest to the utopia point are considered for this metric over the “tails” of the frontier because they provide better insight into whether the algorithm is generating designs capable of satisfying multiple objectives, which is crucial during the preliminary design stage.

Complicating this approach is the ability for the design closest to the utopia point in one generation to be absent from the Pareto frontier in the next generation. This does not mean that this design is no longer Pareto optimal, but simply that it has not been selected by NSGA-II to be included in the reproduction process for the next generation. The new design closest to the utopia point could be farther way from the final target design, which gives the false impression that the algorithm begins to diverge. Therefore, assuming that this distance at a given generation cannot increase from the previous generation, denoted as the “minimum distance”, may give a better indication as to the convergence behavior of the algorithm.

Figures G.1-G.3 indicate the convergence behavior of the genetic algorithm for the 2-D SPIDRS design problems discussed in Section 3.6.4. The cantilevered frame problem (Figure G.1) and the tensile inverter problem (Figure G.2) both appear to converge well before the maximum number of generations. However, one should note that the behavior observed in Figure G.2 could also be attributed to the fact that the tensile inverter has an ideal solution which “bounds” the problem, is easily obtained, and is approximately equidistant to the utopia point across all objective function values. For the 2-D rectifier problem discussed in Section 3.6.4.3 one can observe a large drop in the distance metric right before the maximum number of generations is reached (Figure G.3a). This could indicate that the rectifier design problem is not yet converged, and that considering

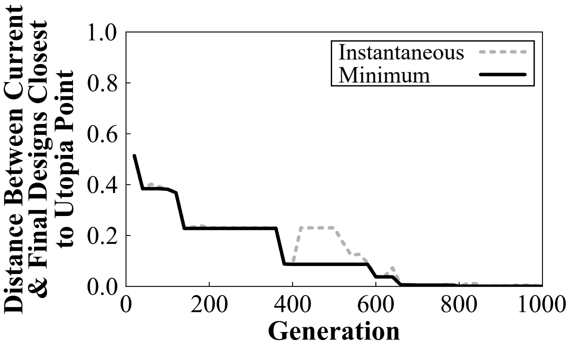


Figure G.1: Convergence plot for the 2-D cantilevered frame design study discussed in Section 3.6.4.1.

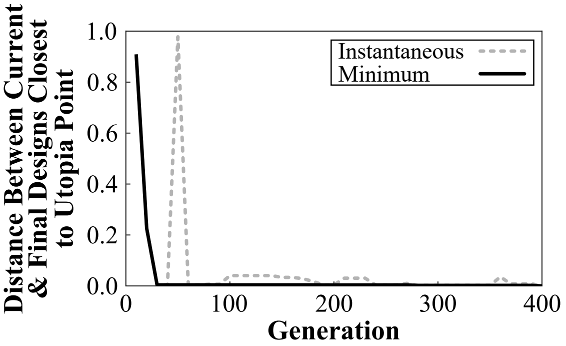
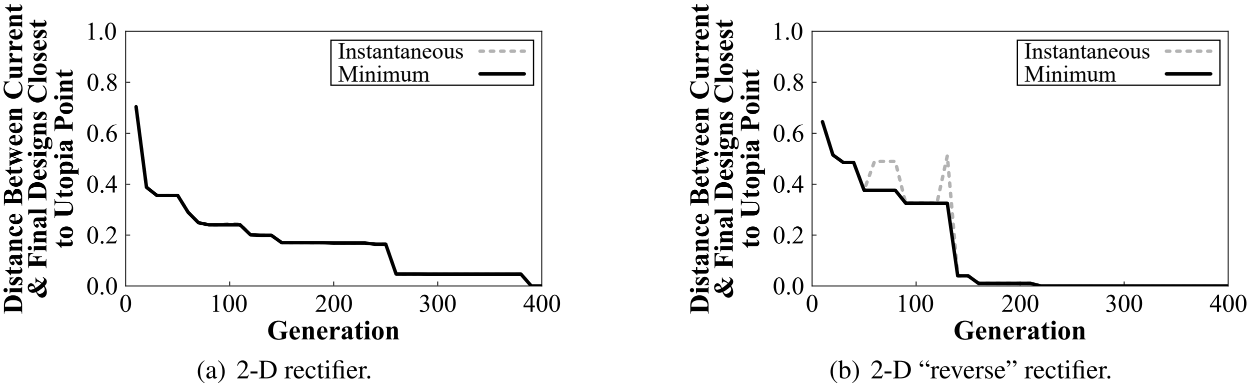


Figure G.2: Convergence plot for the 2-D tensile inverter design study discussed in Section 3.6.4.2.

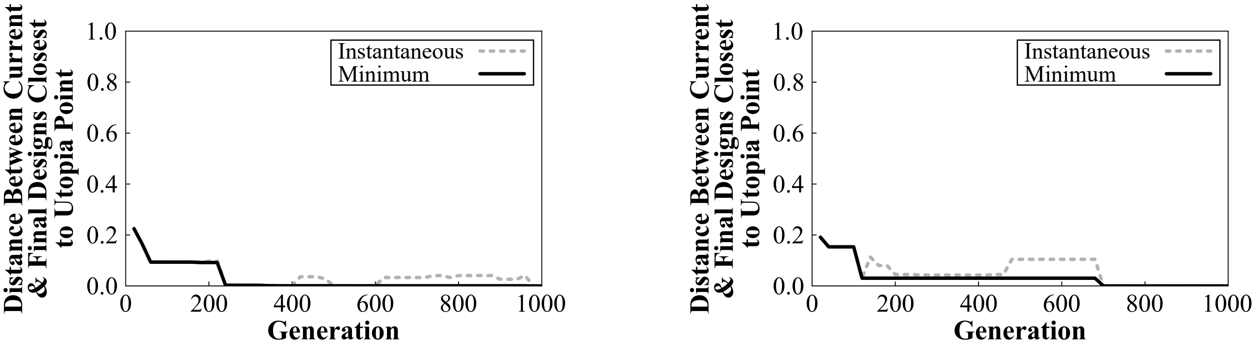


(a) 2-D rectifier.

(b) 2-D “reverse” rectifier.

Figure G.3: Convergence plot for the 2-D kinematic rectifier design studies discussed in Section 3.6.4.3.

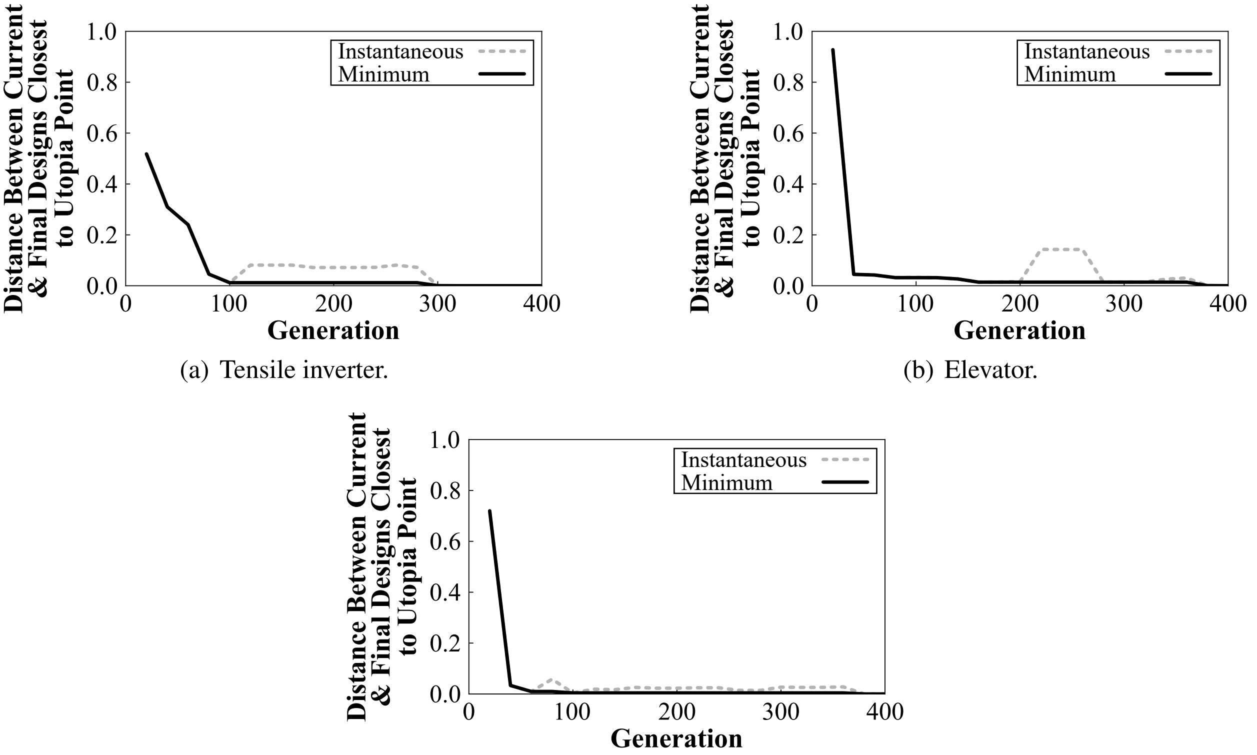
additional generations could result in a further improved set of solutions. On the other hand, the “reverse” rectifier problem (Figure G.3b) appears to converge quickly, which is interesting given the complexity of this problem (cf., Section 3.6.4.3).Figures G.4-G.5 indicate the convergence behavior of the genetic algorithm for the 3-D SPIDRS design problems discussed in Section 4.2. The cantilevered frame problem at both levels of L-System recursion considered (Figure G.4) and each of the 3-D compliant mechanism problems (Figure G.5) appear to converge prior to the end of the optimization, with the behaviors observed in Figure G.5 again possibly attributed to the ideal “bounded” solution associated with each problem.



(a) 2 L-System recursions.

(b) 3 L-System recursions.

Figure G.4: Convergence plot for the 3-D cantilevered frame design studies discussed in Section 4.2.1.1.



(a) Tensile inverter.

(b) Elevator.

(c) Cruncher.

Figure G.5: Convergence plot for the 3-D compliant mechanism design studies discussed in Section 4.2.2.