TOPOLOGY OPTIMIZATION METHODS FOR HEAT SINK DESIGN APPLIED TO POWER ELECTRONICS

BY

DANNY JOHN LOHAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Associate Professor James Allison,
Professor Daniel Tortorelli,
Assistant Professor Nenad Miljkovic,
Associate Professor Pingfeng Wang

Abstract

Topology is defined here to relate the geometric properties and spatial relations unaffected by the continuous change of shape or size of physical engineering artifacts. Researchers have used topology optimization methods to design next-generation structural components for a wide variety of applications (e.g., automotive, aerospace, industrial machines, etc.). This has enabled improved part performance, reduced component volume, and the elimination or reduction of subsystem assembly steps through the use of a free-form structural design representation for commercial products. The free-form capability of topology optimization methods has enabled realization of novel parts otherwise difficult to obtain through the optimization of parametric models.

In this dissertation, topology optimization methods are applied to design heat spreading structures for next-generation power electronic systems. Wide band-gap semiconductor devices provide an opportunity to operate next-generation power electronics at higher temperature. The resulting increase in temperature capability is not shared by all components within a power electronic circuit, and careful consideration must be taken when routing heat throughout the system. Topology optimization methods provide a flexible and powerful solution to address challenges associated with these next generation electronic systems.

First, a framework to classify general topology optimization problems is presented. This framework draws upon research from multiple application domains and defines a unifying language. Then, some practical heat conduction formulations for topology optimization are reviewed. This includes an analysis of common objective functions, in addition to some practical formulations for power electronics. Next, topology optimization methods are applied to design a heat spreader for a next-generation power inverter. The topologically-optimized design is fabricated and benchmarked against a baseline heat sink design. These studies motivated an investigation of reduced-order convection models for topology optimization, where experimental evaluation was used to assess heat spreader performance. The topology of the best-performing structure motivated a critical look at the structural representations used in topology optimization. For heat spreader design, it is common to use maximum approximation functions within a topology optimization. Several maximum approximation functions are analyzed in detail for use as optimization objectives and constraints. Advancing these topology optimization methods extends the envelope of design capabilities and therein lies the future of next generation power systems.

To John and Maria Lohan.

Acknowledgments

I would like to thank my advisor Professor James T. Allison for his continued support and unending patience throughout the past 8 years. I started as an undergraduate transfer student at the University of Illinois and was assigned Professor Allison as a faculty advisor. This relationship quickly evolved from coursework to undergraduate research forming the beginning of my graduate career. Professor Allison has always encouraged me to challenge the norm and together we have explored interesting and novel concepts. I would also like to thank my committee for their insightful suggestions in strengthening my dissertation research.

I would like to thank my industry mentor Dr. Ercan (Eric) M. Dede for his continued support during my graduate studies. Eric has co-authored several of the works presented in this dissertation and helped keep my research industry relevant.

In the second leg of my degree, I have had the opportunity to collaborate with research teams headed by Professor Robert Pilawa and Professor Kai James. Together with their graduate students Derek Chou, Zitao Liao, and Alex Jessee, we have been able to publish interesting papers. As such, I would like to thank them for their support.

I would also like to thank my labmates in both the Engineering System Design Lab (ESDL), Daniel R. Herber, Anand P. Deshmukh, Tinghao Guo, Yong Hoon Lee, Satya R.T. Peddada, Vedant, Albert Patterson, Tonghui Cui, and NSF center for Power Optimization of Electro-Thermal Systems (POETS), Andrea Wallace, Tarek Gebrael, Thomas Foulkes, Donald Docimo, for the great years of discussion that helped shape this work. In addition to my undergraduate researchers Will Jones and Xinyi (Cindy) Li, who helped perform many of the experiments herein. I would like to convey a special thanks to Pamela J. Tannous who helped in editing much of this

document. I am grateful to all for the time we spent together making this PhD a pleasant and worthwhile experience.

Finally, I would like to acknowledge the support from the NSF center for Power Optimization of Electro-Thermal Systems (POETS) and the Toyota Research Institute of North America (TRI-NA) for supporting the research presented in this document.

Contents

List of Figures

List of Tables

List of Symbols

$\circ$	Hadamard product
α	density scalar
β	threshold scalar
$\beta()$	mapping function
γ	SIMP penalty power
Γ_D	Neumann boundary
Γ_N	Dirichlet boundary
Γ_q	prescribed heat flux boundary
ϵ	boundary definiteness
Θ	objective function
Θ'	optimal objective value
$\tilde{\Theta}$	approximated objective value
κ	thermal conductivity
$\boldsymbol{\lambda}$	adjoint vector
μ	dynamic viscosity
μ_d	dynamic viscosity
Π	function of objective
ϕ_q	signed distance from point to edge
ρ	finite element density
$\tilde{\rho}$	filtered density
$\bar{\rho}$	density of projected bar
$\hat{\rho}$	penalized density of projected bar
σ	architecture generation actions
τ	generation action
$\Omega_{\square}$	design domain
a	architecture object
A	set of feasible architectures
A_d	area of device
A_e	area of finite element
$\mathbf{d}$	design vector
$d()$	distance function
E	edge
f	input heat flux
$f()$	feasibility checking function
$F()$	feasibility checking function
G	feasible set of design variables
G	vertex labeled graph
h	convective heat transfer coefficient

I_d	device current
$I()$	isomorphism checking function
$\mathbf{K}$	FEA stiffness matrix
L	label
l_g	length of vein growth step
m	model
$\mathbf{n}$	normal vector
N	neighborhood for density filter
N_e	number of evaluated architectures within a given budget
N_g	number of evaluable architectures within a given budget
N_f	number of realized architectures
n_r	number of rules
n_g	number of growth stage
n_s	number of stem nodes
O	overlap constraint
p	p-norm power
P	pressure
$\mathbf{P}$	FEA loading vector
P_L	power loss
q	heat flux
Q	heat generation
R	realizable set of architectures
R()	radius function
$R_{\min}$	minimum radius for filter
$\square_r$	relaxed form
r_e	ratio of evaluated architectures to generated architectures
r_g	ratio of generated architectures to feasible architectures
$\mathcal{S}$	set of nodes
T	temperature
∇T	temperature gradient
T_{amb}	ambient temperature
T_{av}	average temperature
$T_{\max}$	maximum temperature
u	velocity
$\mathbf{U}$	FEA displacement vector
V	vertex
V()	volume
$V_{\max}$	maximum volume
V_d	device voltage
w	weighting function for density filter
Z	set architecture variables

List of Abbreviations

AMI	Adjacency matrix interconnects
CFM	cubic feet per minute
EMI	Eletro-magnetic interference
FCML	flying capacitor multi-level
GaN	Gallium Nitride
ILP	Inner-loop problem
MNP	Modified nodal analysis
NSC	network structure constraints
PCB	printed circuit board
PM	Perfect matching
SiC	Silicon carbide
SIMP	solid isotropic microstructure with penalization method
TIM	thermal interface material

Chapter 1

Introduction

1.1 Topology Optimization

In mathematics, topology is defined as:

> *"the study of geometric properties and spatial relations unaffected by the continuous change of shape or size of figures".*

In some cases, the topology of an entity may be represented as a graph. If this is the case, topology involves the connection of a set of nodes by edges. This is demonstrated in Fig. 1.1.

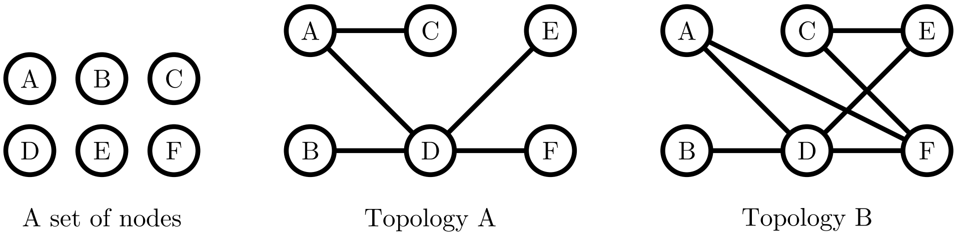


Figure 1.1: Topologies generated from a set nodes (left), Topology A (center), Topology B (right).

Given a set of nodes $\mathcal{S} := \{A, B, C, D, E, F\}$, there are a combinatorial number of unique topologies that can be produced. Two topologies have been presented in the figure as examples. Non-topological changes that can be applied to a topology may be defined by two categories, changes in shape and changes in size, as shown in Fig. 1.2.

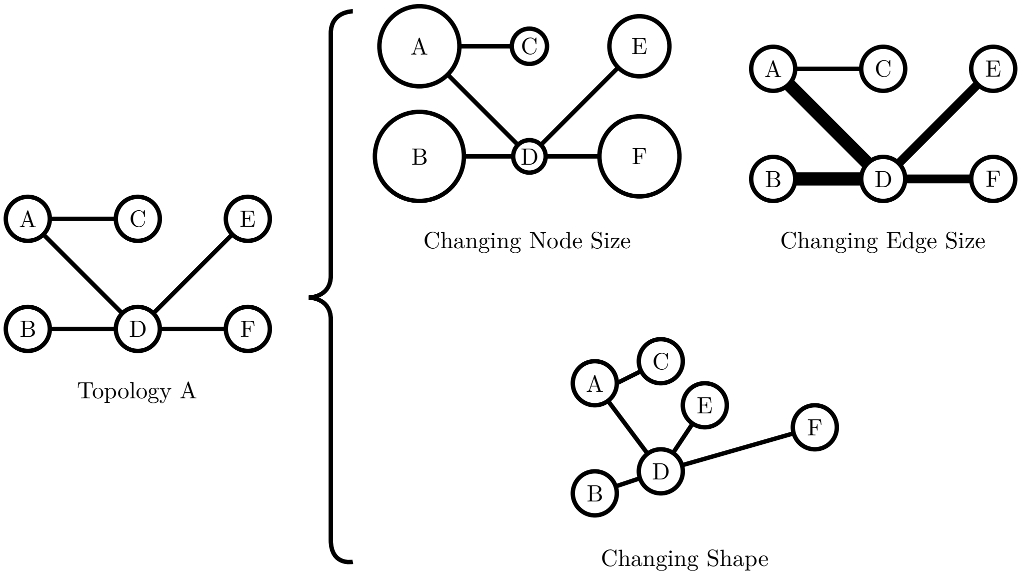


Figure 1.2: Graph operations with constant topology.

Size modifications are easily understood as a scaling of nodes, edges or a combination of both. These can be done separately, such as only node sizing or only edge sizing, or may be done together. Shape or geometry operations require spatial information for a graph and are defined by translating the nodes of the graph. Understanding these key differences allows for a powerful categorization of topology optimization problems.

1.1.1 Structural Representations

Over the past three decades, there have been a variety of new design representations developed based on the use of finite element analysis with a fixed analysis mesh [1–4]. Using gradient-based optimization with these fixed analysis mesh methods, structural topology optimization problems with millions of design elements have been solved [5]. Consider the seminal paper by Bendsoe [6], where the density of individual finite elements are allowed to vary between solid and void as the design variables. Mathematically, a size optimization problem is being solved, however, at a macro-scale new topologies visually emerge. A popular adaptation to this method involves using a level-set function to represent the design and mapping this level-set function to the density field [2]. Optimizing the level-set function is mathematically equivalent to shape optimization, though at a macro-scale new topologies may visually emerge. More recently, differentiable projections for explicit geometric elements have been developed and coupled with these fixed-mesh strategies [3, 4]. Using these methods is certainly shape optimization, though new topologies may emerge visually. The general design representations can be mapped with additional abstractions to perform targeted searches of the solution space [7, 8], however, these are limited in flexibility. In this dissertation, several structural design representations/abstractions are used for heat spreader design.

1.2 Dissertation Overview

This chapter has introduced the concept of a topology and has presented several ways to construct or represent a topology for use in structural optimization. First, a unifying framework is presented to encapsulate a wide class of topology optimization problems, Ch. 2. This classification is used to identify a subclass of topology optimization methods suitable for structural optimization. Then, an investigation into conduction dominated power electronics relevant problem formulations is presented, Ch. 3. This is followed by two studies using reduced-order convection models for convective heat transfer design in Chapters 4 and 5. Findings are used to motivate an assessment of different structural design representation and enhancements are proposed in Chapters 6 and 7. This is followed by a study of maximum approximation functions, Ch. 8, and the dissertation is concluded in Ch. 9.

- Chapter 2 defines a general framework for classifying topology optimization problems using mathematic definitions. This framework is developed in an effort to bridge the gap between related, but isolated, research communities. Aspects of topology optimization design problem formulations are decomposed and mathematically expressed to allow for direct comparison between research communities.

- Chapter 3 focuses on the adaptation of the Solid Isotropic Microstructure Penalization topology optimization formulations using conductive heat transfer for power electronics applications. This study stems from the common (incorrect) use of pseudo-objectives in literature. Differences in optimized structure for a variety of objective functions are presented and power electronics relevant problem formulations are presented and solved.

- Chapter 4 presents an application of reduced-order topology optimization for a novel folded flying capacitor multi-level inverter architecture. This includes design, fabrication, and experimental validation for a benchmark heat spreader and optimization adapted heat spreader design.

- Chapter 5 extends previous findings to perform an exploratory search for reduced order methods for convective heat sink design. This includes varying problem formulation aspects such as the design representation, objective function, physics, and boundary conditions. A selection of optimized designs is experimentally benchmarked to a simple heat sink design.

- Chapter 6 compares three fidelities of design representation as they apply to the heat conduction optimization problem. This includes the voxel-based representation of Solid Isotrpic Microstructure with Penalization method, an explicit geometric representation as used in the Geometric Projection Method, and a generative abstraction through the use of the Space Colonization Algorithm.

- Chapter 7 outlines challenges in using projections of explicit geometric components for heat conduction structural optimization. These challenges are similar to those present when using the Solid Isotropic with Penalization method and several enhancements are suggested to produce feasible structures.

- Chapter 8 presents an analysis of maximum approximation functions in the context of requirements for topology optimization. Suitable functions are identified and suggestions for improving optimization stability are discussed.

- Chapter 9 concludes with research summary of key findings and suggestions for future areas of work.

Chapter 2

A classification of topology optimization problems.

Collaborative Acknowledgement: This chapter is a culmination of several years discussion between myself, Professor Daniel Herber and Professor James Allison. The formalized definitions, interpretations, and classifications are a collaborative effort between myself and Daniel Herber. **Dr. Daniel Herber** contributed with his expertise in graph theory, enumeration, and heterogeneous system design. My own contributions focused on relaxation methods, design abstractions, and evolutionary approaches. Together we have been able to create standardized definitions applicable to a wide variety of topology optimization domains and bring to light key aspects that have otherwise been overlooked. Note that in this chapter the word “architecture” is chosen to describe topology, or configuration, as it is more widely used in design literature.

2.1 Introduction

Determining the preferred system architecture is a challenging endeavor for many engineering systems, such as material structures [5, 7, 9–12], electric circuits [13–20], hybrid powertrains [21–26], fluid networks [27, 28], vehicle suspensions [29–31], mechanisms [32, 33], synthetic biology [4, 34], and more [35–42]. These domains have their specific challenges, but there are insights and shared features between them because they are fundamentally architecture design problems. While the system architecture decisions of what should comprise the system and how it is interconnected has been left frequently to human designers, many recent advances in computational methods have enabled the realization of novel and high-performance solutions via design automation. The field of systematic architecture design, however, is still developing, and there is room for improvement. There is a need for universally-accepted representation and language that facilitates the exchange of research approaches and insights, and the development of appropriate and effective generation, search, and evaluation methods [43].

In this chapter, we only consider architecture design problems with a discrete space of potential architectures of finite size. These discrete architectures are represented by labeled graphs to result in a problem form that is similar to the connecting pattern of system architecting decisions [44]. Furthermore, this work focuses on describing architecture problems that feature complex structure and evaluation to assess performance, such as a complete engineering design optimization problem. With this broad definition for the architecture problem class of interest, it can be challenging to understand the solution space and develop effective solution strategies for identifying the preferred architectures.

There are many review papers, such as Refs. [10, 21, 43, 45–47], covering a variety of architecture design domains and methodologies, but here we focus on formalizing the mathematical descriptions of the considered class of architecture design problems. The contributions of this work are: 1) the presentation of a unifying mathematical representation that covers a wide range of seemingly divergent existing architecture design problems; 2) showing the commonalities and differences between domains and methods; and 3) analyzing how a limited number of diverse existing studies fit within the proposed representation. First, some important concepts and definitions are presented. Next, a generalized problem formulation is presented encompassing the class of architecture design problems. The following sections discuss the four solution strategy modules (generator, modeler, evaluator, and searcher). With the universal representation defined, a few select architecture design studies are analyzed. Finally, a discussion of some important observations and suggestions for future work are in the conclusion section.

2.2 Foundational Concepts and Definitions

First, some relevant fundamental concepts and definitions are given. Here we will be discussing how to solve a particular class of combinatorial optimization problems.

> **Definition 1 (Combinatorial Optimization).** *A combinatorial optimization problem has the following form:*
>
> $$\min_{a} \quad \Theta(a) \tag{2.1a}$$
>
> $$\text{subject to:} \quad a \in A \tag{2.1b}$$
>
> *where a is the discrete object that must be chosen, Θ is the objective function, and A is the set of feasible solutions.*

There are many classical combinatorial optimization problems, such as the traveling salesman, knapsack, assignment, or vertex coloring problems [48]. Methods designed for these combinatorial problems are challenging, and in most cases impossible to apply to our problems of interest. Although our architectures are defined by graph-theoretic concepts, their performance evaluation as a complex engineering systems is not.

A potentially straightforward approach to determine the solution to a combinatorial optimization is to evaluate $\Theta(a)$ for all discrete elements in A.

> **Definition 2 (Enumeration).** *An enumeration is an ordered listing of all the elements of a set. An enumerative method is a technique for generating an enumeration (and not just counting how many elements are present in the complete set). An enumeration-based method for solving Prob. (2.1) evaluates all candidates from the enumeration of A and selects the one with minimum value.*

Both generating an enumeration and evaluating Θ can be prohibitively expensive, motivating the development of alternative approaches.

The objects of interest can be represented by labeled graphs [49]. However, all references to labeled graphs here imply vertex-labeled graphs for simplicity.

> **Definition 3 (Labeled Graph).** *A vertex-labeled graph G is a triplet (V, E, L) of sets satisfying $E \subset [V]^2$ where V are the vertices, the elements of E are the edges, and $L = \{V_i\}_{i=1}^{k}$ is a partition of the vertices into label sets ($V_i \cap V_j = \Theta, i \neq j$, and $(\cup_{i=1}^{k} V_i) = V$). For convenience, we define the function ℓ that maps every vertex to a label.*

In some cases, an enumerative method may produce duplicate graphs. Identifying *unique* graphs can be performed through isomorphism checks [49].

> **Definition 4 (Labeled Graph Isomorphism).** *$G = (V, E, L)$ is isomorphic to $G' = (V', E', L')$ if and only if there exists an objective function $\rho : V \to V'$ such that:*

> 1. *For all* $v_i, v_j \in V$*,* $(v_i, v_j) \in E \Leftrightarrow (\rho(v_i), \rho(v_j)) \in E'$
> 2. *For all* $v \in V$*,* $\ell(v) = \ell(\rho(v))$

For the class of design problems considered in this work, the discrete objects are termed architectures.

> **Definition 5 (Architecture).** *An architecture is a discrete object that represents the system's function and can be represented by a labeled graph. Alternative terms are often used to describe the same concept including topology, configuration, and network.*

Finally, understanding the problem structure and dependencies will be critical to developing effective solution strategies.

> **Definition 6 (Dependence).** *A mathematical object* O *is said to depend on another object* d *if any property of* O *changes when* d *changes. This dependence relation is denoted* O^d*.*

2.3 Architecture Design Problem Formulations

In this section, we define a number of problem formulation elements and features that might be present when investigating graph-based architecture design problems. While this is not a complete list, the features will be useful for discussing feasible regions, optimality, potential solution strategies, and more.

2.3.1 Problem Elements

First, an architecture object is denoted with the symbol a. We will want the architectures to be in some finite feasible space, denoted as:

$$a \in A \tag{2.2}$$

and this feasible space can be described in a variety of ways depending on the specific architecture class considered.

Many problem elements may *depend* on the considered architecture, and this relation is shown with the superscript a. We will also utilize the notation of a model m^a which is broadly defined as some artifact that is needed during the evaluation of a design problem. Additional parameterizations with design variables $\mathbf{d}^a$ may also be required for evaluating a given architecture problem. Now, given some candidate architecture a and candidate design variables $\mathbf{d}^a$, the performance metric can be represented by:

$$\Theta^a(\mathbf{d}^a, m^a) \tag{2.3}$$

However, not all values for $\mathbf{d}^a$ may be considered feasible, so we can also define the feasible set of the design variables G:

$$\mathbf{d}^a \in G^a(m^a) \tag{2.4}$$

which may depend on the model m^a and architecture a. The optimal performance for a given architecture can be determined by solving an optimization problem with respect to $\mathbf{d}^a$, considering the performance defined in Eqn. (2.3), and the feasible region in Eqn. (2.4):

$$\Theta'(a) = \inf_{\mathbf{d}^a} \{\Theta(\mathbf{d}^a, m^a) : \mathbf{d}^a \in G^a(m^a)\} \tag{2.5}$$

Determining the value in Eqn. (2.5) for a given architecture is termed the *inner-loop problem* (ILP). The use of infimum is important as there may not be any $\mathbf{d}^a$ that is feasible.

2.3.2 Nested vs. Simultaneous Optimization Form

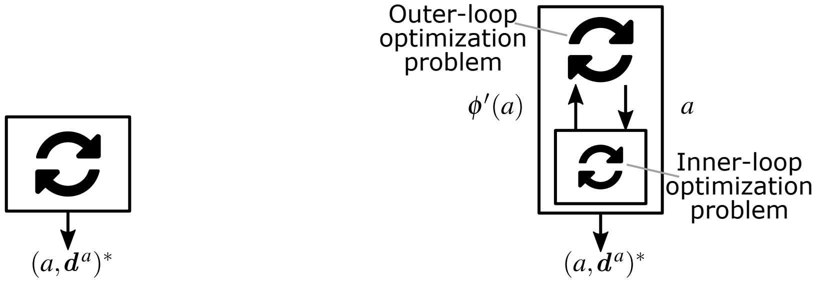
The first problem formulation feature is the distinction between the nested and simultaneous forms of the combined architecture and parameterization problem. In a nested form, an ILP represented by Eqn. (2.5) is defined which depends on the architecture. The determination of the optimal performance for a given architecture is used as the objective in the outer loop which seeks to minimize this performance value subject to a feasible set of candidate architectures. The nested form then is represented as:

$$\begin{aligned} \min_{a} \quad & \phi'(a) && \text{(2.6a)} \\ \text{subject to:} \quad & a \in A && \text{(2.6b)} \end{aligned}$$

which is quite similar to the form of general combinatorial optimization problems in Prob. (2.1) [48].

In contrast with the nested form, a simultaneous form does not determine ϕ' for every candidate architecture, but rather explores different values of $\mathbf{d}^a$ while simultaneously searching the set of architectures. The simultaneous form is represented as:

$$\begin{aligned} \min_{a,\mathbf{d}^a} \quad & \phi^a(\mathbf{d}^a, m^a) && \text{(2.7a)} \\ \text{subject to:} \quad & a \in A && \text{(2.7b)} \\ & \mathbf{d}^a \in G^a(m^a) && \text{(2.7c)} \end{aligned}$$


$(a, \boldsymbol{d}^a)^*$
Outer-loop optimization problem
$\phi'(a)$
a
Inner-loop optimization problem
$(a, \boldsymbol{d}^a)^*$

(a) Simultaneous form.

(b) Nested form.

Figure 2.1: Two problem forms.

Figure 2.1 illustrations conceptually the nested and simultaneous problem forms.

2.3.3 Object vs. Assembly Architecture Form

Object form directly seeks to identify the best architecture object by searching over candidate architecture objects. Both Eqn. (2.6) and Eqn. (2.7) are presented in object form. A commonly used alternative is to *assemble* architectures from architecture assembly variables $\boldsymbol{z} \in Z$, where Z is a set of feasible assembly variables. The assembly method maps $\boldsymbol{z}$ to a and architectures assembled in this manner are denote $a_{\boldsymbol{z}}$.

The key notion differentiating these forms is the use of assembly variables. In the object form, candidate architectures are evaluated holistically as an object. In the assembly form, there are independently adjustable variables that define an architecture; assembly variables are used in the optimization process.

2.3.4 Architecture-Independent Problem Elements

All currently presented problem elements have included dependence on the architecture if a dependence relationship could potentially exist. However, when discussing the structure of architecture design problems, it is essential to understand which problem elements depend on the architecture and which do not. Problem elements may be partitioned into parts that do depend on the architecture and those which do not. For example, for the feasible set of the design variables, we might have the following forms: G^a, G, or $G \cup G^a$. In Ref. [46], the three variations $\mathbf{d}^a$, $\mathbf{d}$, and no design variables are discussed. Another important dependence to consider is the model and whether or not it is "architecture independent", i.e., does a new model need to be constructed for each architecture? This will be discussed more in Sec. 2.6.1.

2.3.5 Relaxed Forms

Another common strategy is to relax the original set of feasible architectures A using some suitable relaxation denoted with the subscript r:

$$a \in A_r \text{ with } A_r \in A \tag{2.8}$$

In the case of the assembly form, we can potentially relax two problem elements:

$$\boldsymbol{z} \in Z_r \text{ and } a_z \in A_r \text{ with } A \in A_r \text{ and } Z \in Z_r \tag{2.9}$$

where we may relax either the set of feasible architectures and/or the architecture assembly variables that construct them. A continuous relaxation of some discrete variables is a common relaxation strategy [10].

A key feature of using relaxed forms is a defined mapping $D : A_r \mapsto A$ used to return back to the original space of architectures. Without this mapping, we are not solving the originally-specified architecture synthesis task. The use of a relaxed form can enable effective solution methods, but fundamentally changes the search space.

2.3.6 Single vs. Multiple Objectives

Another important feature of an architecture design problem is the number of objective functions. Due to the nature of many system architecture studies, we seek to understand tradeoffs between different metrics, leading to multiobjective problem formulations [50] and subsequent tradespace analysis [44, 51]. The simultaneous object form with multiple objectives is:

$$\begin{aligned} \min_{a} \quad & (\phi'_1(a), \phi'_2(a), \dots, \phi'_k(a)) && \text{(2.10a)} \\ \text{subject to:} \quad & a \in A && \text{(2.10b)} \end{aligned}$$

where we have k different objectives, and solving Prob. (2.10) requires finding the set of nondominated solutions. Determining this Pareto or fuzzy Pareto set can require special methods [44, 52].

2.4 Solution Strategy Modules

In the previous section, we focused only on the design problem formulation and not on any specific solution strategy for determining the optimal architecture(s). Independent of the specific solution strategy, we describe four generic modules, adapted from the general frameworks in Refs. [44, 45, 47] for finding "preferred" architectures given an architecture design problem formulation.

These modules and their interconnections are shown in Fig. 2.2.

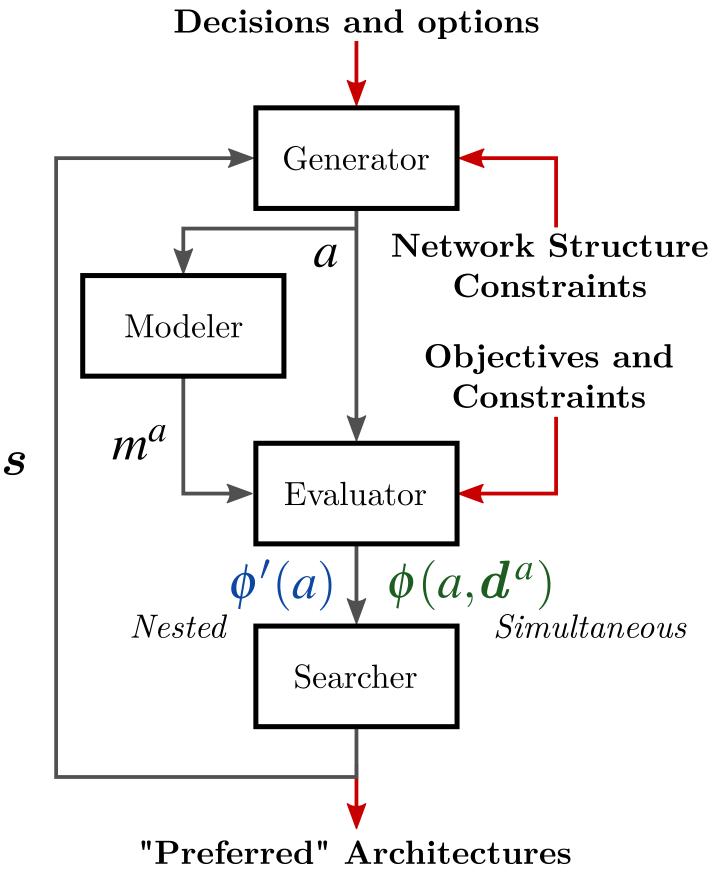

Decisions and options
Generator
Network Structure Constraints
a
Modeler
Objectives and Constraints
s
m^a
Evaluator
$\phi'(a)$
$\phi(a,\boldsymbol{d}^a)$
Nested
Simultaneous
Searcher
"Preferred" Architectures

Figure 2.2: Generic modules and their interconnections for architecture design problem solution strategies.

There are four main modules in this framework: a generator, modeler, evaluator, and searcher. Each of these modules performs a particular function towards the goal of identifying the preferred architectures. Decisions and options are used by the generator to create architectures, and network structure constraints may be included to limit what the generator outputs. The output of the generator may be one or many architecture objects. The modeler creates the model artifact m^a given a. Next, the objective(s) and constraints are used in the evaluator, and the evaluator's role is dependent on whether a nested or simultaneous optimization form is used, as discussed in Sec. 2.7. Finally, the searcher takes in performance metrics and all other architecture information (potentially assembly and design variables, optimal or not, as well as the architecture object) and determines suitable search directions defined by $\boldsymbol{s}$. These search directions are then passed to the generator which uses this information to generate new architectures. The procedure continues in this manner until a stopping criteria is met.

The following four sections will describe each of these modules in more detail and provide some concepts that can be considered for both existing and new architecture design problems.

2.5 Generator

The generator is responsible for producing sets of candidate architectures. In this section, we will discuss some important properties related to the generator and review some of the more common structures that can be used to define the feasible set A.

2.5.1 Set of Realizable Architectures

To begin the discussion on the properties of the generator, consider again some ideal bounded feasible space A that we consider to be the set of feasible architectures. Understanding the magnitude of this number is both conceptually and practically important to developing effective solution methods.

As shown in Sec. 2.3.3, we may consider an assembly form which defines a set of all realizable architectures:

$$R = \{a_z : \boldsymbol{z} \in Z\} \tag{2.11}$$

We can develop some insights into the chosen assembly form by considering the following three relationships between the sets A and R:

1. If $R = A$, then the generator can produce all feasible architectures exactly

2. If $R \subset A$, then there may be better performing architectures that are not realized by the chosen generator (unless all architectures not in R are worse than the best architecture in R)

3. If $R \backslash A \neq \emptyset$, then we must avoid certain combinations of $\boldsymbol{z}$ as they would not produce feasible architectures

These relationships are briefly discussed in Ref. [47]. Similar relationships can be discussed if we are considering the relaxed form substituting A with A_r.

2.5.2 Network Structure Constraints

Network structure constraints (NSCs) is a general term for restrictions placed on certain arrangements of the labels and connections in an architecture; NSC violation renders an architecture topologically infeasible [53]. By definition then, all architectures in A satisfy all NSCs. Generally, NSCs vary from problem to problem and can be based on fundamental restrictions or general knowledge/intuition surrounding the system of interest. References [39, 53, 54] discuss general NSCs and methods for checking/satisfying them.

Here we define the NSC feasibility checking function $f(\cdot)$ as equal to one when all NSCs are satisfied, and zero otherwise. This function can be applied to a set of architectures to obtain all feasible architectures:

$$R_F = F(R) := \{a \ : \ a \in R \text{ and } f(a) = 1\} \subseteq A \tag{2.12}$$

2.5.3 Isomorphic Architectures

In Sec. 2.2, the concept of isomorphic (labeled) graphs was presented. Ideally, for the nested form, we do not want to consider two architectures that are isomorphic with respect to their labeled graph representations because 1) computational cost is increased without producing any additional information, and 2) the searcher may be biased toward frequently occurring architectures rather than only architecture performance.

This condition is slightly different for the simultaneous form where two architecture instances are isomorphic if both the labeled graph representations and current values of the design variables are isomorphic (i.e., we would still want to consider isomorphic architectures in the simultaneous form, but not isomorphic architecture instances). Determining the set of nonisomorphic architectures from a given list can be computationally expensive [49, 54, 55]. Therefore, checking for isomorphic architectures may only be practical when this cost is sufficiently small, but the two previous issues would remain.

Here we define a set isomorphism checking function, $I(\cdot)$, where all architectures that are nonisomorphic in a given set are determined.

2.5.4 Relative Generator Coverage

While discussing the entirety of the feasible and realizable sets is useful, the practicalities of using some specific generator are also unavoidable. Therefore, we can define the coverage of a selected generator for some fixed computational cost budget that may be used to generate candidate architectures. We define the generator coverage ratio as:

$$r_g = \frac{N_g}{N_f} \tag{2.13}$$

where N_g is the number of feasible/evaluable architectures that the generator can produce in the given computational budget, and N_f is the number of feasible architectures that are realizable using the generator. This ratio can be used to identify three cases of particular interest:

1. $r_g = 1$: The generator can produce all feasible architectures

2. $0 \ll r_g < 1$: The generator can not produce all feasible architectures but some reasonable portion

3. $r_g = \epsilon$ where $\epsilon > 0$ is a small number: The generator can only produce a near zero percentage of the total feasible architectures

Different generators for the same feasible set A could have widely varying values for r_g. Seeking a larger value for r_g does not guarantee the existence of the best or good architecture(s) in A, as such it is important to considered a targeted generation procedure. For example, we may consider using one generator to analyze the space, A, then select another, more useful generator, to produce an available subset of architectures. We also note that there may be a nonlinear relationship between r_g and the computational budget. In some cases, it may be more challenging to find unique, feasible architectures as more architectures are generated.

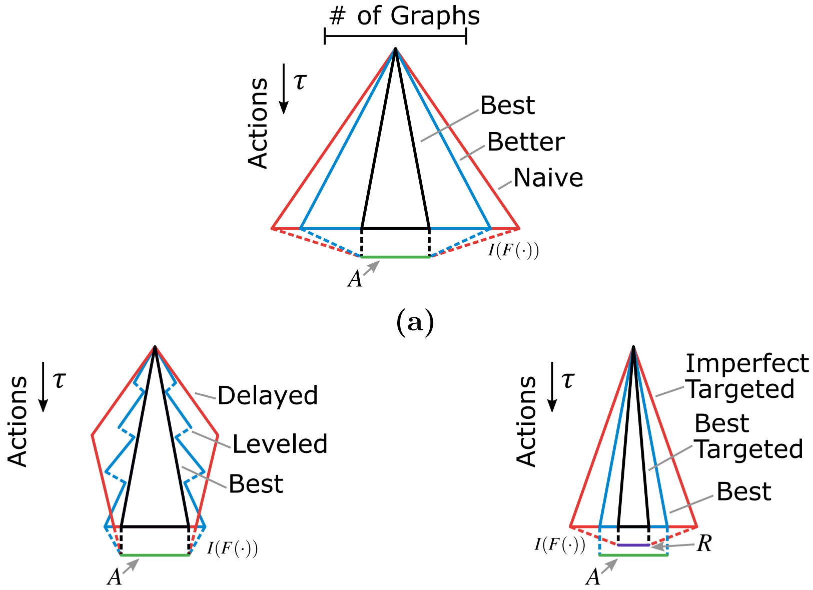
2.5.5 Action-Based Generators

The choice of architecture generator will influence the size and composition of A. The most naive "generator" is a list where candidate architectures are simply directly available, and is only appropriate for object form. Where this list comes from can vary; two interesting examples are suspension graphs found using linear matrix inequalities [31] and a list of best low-pass filter graphs determined from an enumeration-based study [56].

Frequently, an appropriate list is not readily available, and there is a need for an effective means of generating architecture candidates. Many generators use some particular representation and assembly method to define architecture candidates. This assembly process for a single architecture is represented as:

$$a = a_0^{(\tau_1, \tau_2, \ldots, \tau_k)} \tag{2.14}$$

where the architecture a is assembled from the starting architecture a_0 and k actions are performed in sequence as $\tau_1, \tau_2, \ldots, \tau_k$. These actions are graph modifications used to construct/modify an architecture. Understanding the space of all realizable architectures is equivalent to understanding all possible permutations of the action sequence. This space of permutations can be analyzed through tree diagrams where the root of the tree is the initial architecture, each level in the tree represents a specific action that was taken, and the branches represent all potential actions that could be taken. For some generators, the number of actions required to define an architecture depends on the chosen assembly method, while for other generators it can be a user-defined parameter.



(a)

(b)

(c)

Figure 2.3: Illustrations of action-based generator structure.

To illustrate the process and structure of the applied actions, Fig. 2.3 is provided. In order to compare the result of the generation process to A, both $I(\cdot)$ and $F(\cdot)$ are applied to the resulting R to obtain the set of unique feasible architectures. In Fig. 2.3a, all three cases match A in this illustration, with the only difference being the number of generated graphs needed to enumerate A. In the naive case, many additional infeasible and/or isomorphic architectures are generated. In the best case, only unique feasible architectures are generated. This behavior is illustrated for two particular methods in Ref [37]. Figure 2.3b illustrates a different concept of intermediary reductions in the set of intermediary graphs. The mechanism for the reductions can vary; for example, in a delayed case, the number of graphs grows until a point is reached where many candidate graphs have no feasible potential action. For the leveled case, feasibility or isomorphism checks could be applied to the current set of graphs to reduce the number of graphs considered at the next action. Finally, Fig. 2.3c illustrates the concept of a targeted realizable space that does not match with the original A. Some targeted methods may only generate nonisomorphic feasible graphs but others might require the application of F and I to obtain the desired R.

Here we will discuss three different generator types beyond the basic list, namely, adjacency matrix interconnections (AMI) [34, 44, 57], perfect matchings (PM) [54], and rules [7, 39, 43, 45, 46]. For the adjacency matrix interconnections, the number of actions is equivalent to the number of potential connections between vertices [34, 44, 57]. It is important to note that a specific relaxation approach is commonly used for this generator, namely replacing $z \in \{0, 1\}$ with the continuous relaxation $z \in [0, 1]$ [9–11]. In this case, the tree diagram representation does not make sense, as the realizable set is continuous rather than finite, resulting in a drastic change in the search space.

For the PM approach, an even number of connections is assumed and all vertices are associated with exactly one edge (a perfect matching) [54]. Therefore, each action adds a single edge and the number of actions is half the number of connections. The AMI (naive) and PM (better) generators are visualized in Fig. 2.3a as neither produce A directly, but the number of graphs generated using PM is much smaller than with AMI [54]. Algorithmic improvements discussed in Ref. [58] feature both the delayed and leveled characteristics of Fig. 2.3b.

The final generator discussed here is a rule-based approach such as a grammar method [7, 43, 45, 46]. These rules can be any arbitrary graph modification procedure and can encode general knowledge, targeting the search space as shown in Fig. 2.3c. One basic assumption we make here for our analysis is that every rule procedure should terminate [59]. If every rule can be applied at every point in the action sequence, then there would be $(n_r)^k$ graphs in an enumeration where n_r is the number of rules, but not all graphs may be unique or feasible (and this is discussed briefly in Ref. [60]). For more information on identifying and applying the appropriate rules, please refer to Refs. [39, 43, 45].

2.6 Modeler

The modeler can come in many forms depending on the domain that the system architecture represents. While models can be constructed manually for each candidate architecture, this approach may prove to be time-consuming and limit the number of architectures that may be evaluated. For classes of architectures, graph and model libraries could be constructed *once* and reused when new, but similar, architecture problems are posed. Typically, automatically-constructed models are desired in many architecture problems where only the architecture object is needed. In some cases, custom interfaces can be constructed that automate human-performed tasks, such as placing submodels and connecting them [56, 61]. Some of the more common modeler frameworks that support some level of architecture-specified models are general block-diagram models [62], bond graph models [26, 61, 63], modified nodal analysis (MNA) method for electric circuits [64], fluid-based thermal networks [65], and other nonphysics-based frameworks [44].

2.6.1 Model Reusability

One of the primary concerns with a modeler used in architecture design is whether or not the architecture model can be reused. Model creation can be costly, and universally applicable models may have forms that can be leveraged in the manifestation of a solution strategy. For this discussion, we will consider models that can be represented in the form:

$$m^a: \;\; 0 = \Pi^a\left(a, \mathbf{d}^a\right) \tag{2.15}$$

The most nonreusable case is when Π^a must be recreated for every architecture instance (i.e., any change to either a or $\mathbf{d}^a$). Improving upon this scenario, we often can construct Π^a such that it does not need to be recreated for every instance of $\mathbf{d}^a$. An example of this case is generating the analytic transfer function equations for a given circuit architecture using MNA [56, 64], which can predict performance for arbitrary values of the circuit sizing parameters ($\mathbf{d}^a$), but must be reconstructed when the architecture changes.

In some special cases, the model can be shared between architectures such as:

$$m^a: \;\; 0 = \beta(a)\Pi\left(a, \mathbf{d}^a\right) \tag{2.16}$$

where $\beta(a)$ is a known efficient mapping between Π and the desired model m^a. In this case, a major element of the model does not have to be reconstructed for each candidate architecture. An example of this case is presented in Refs. [4, 65], where β is the incidence matrix for the system which can be multiplied by a single-state dynamic equation to describe the behavior of an architecture.

This special case is sometimes further simplified to the forms $0 = \beta(a)\Pi(\mathbf{d})$ or $0 = \beta(a)\Pi$, where the latter is commonly found in structural architecture problems when there are no design variables, but only assembly variables [10, 46]. In either of these cases, the model structure is more understood than the arbitrary structure of the models in Eqn. (2.15).

2.7 Evaluator

The role of the evaluator is to determine the performance metric(s). With the nested formulation, the evaluator will solve an ILP to provide the optimal performance of an architecture, $\Theta'(a)$. When using a simultaneous formulation, the performance of an architecture instance, $\Theta(a, \mathbf{d}^a)$, is determined. The computational expense in either case will depend on several factors, such as whether models are architecture dependent, or whether high-fidelity simulations are required. Several evaluator topics relating to the model, relative evaluation cost, and reducing computational expense follow.

2.7.1 Relative Evaluation Cost

Given a fixed computational budget, it is important to understand the implications of the evaluation cost. Consider again the number of unique architectures generated, N_g, as discussed in Sec. 2.5.4. We will assume that all of the candidate architectures have a similar evaluation cost to define a relative measure of expense, r_e:

$$r_e = \frac{N_e}{N_g}. \tag{2.17}$$

where N_e is the number of architectures that can be evaluated. This formula facilities the description of three general cases:

1. If $r_e = 1$, then all of the architectures may be evaluated. In this case, the evaluation has *low* relative cost.

2. If $0 \ll r_e < 1$, then some portion of the architectures can be evaluated. In this case, the evaluation has *medium* relative cost.

3. If $r_e = \epsilon$ where $\epsilon > 0$ is a small number, then only a very small portion of the architectures can be evaluated. In this case, the evaluation has *high* relative cost.

For problems with a relatively high evaluation cost, $r_e = \epsilon$, advanced techniques may be implemented to reduce overall computational expense and increase design space coverage.

2.7.2 Reducing Evaluation Cost

If the modeler and ILP under consideration has medium or high relative cost, then there are some general strategies for reducing the computational cost associated with the evaluator. One elementary strategy is to use utilize models and ILPs that have lower computational cost, but still have sufficient efficacy in determining an architecture's performance relative to the other candidates. To understand this potential tradeoff, consider the intention of the evaluator: to provide *absolute* or *relative* performance metrics. If the identification of the best architecture(s) is the primary goal, then maintaining the rankings between different models or formulations will suffice. Metrics can be used to quantify the agreement between problems of differing expense, such as the root-mean-square error or the Kendall tau rank distance [66], which counts the number of pairwise disagreements between two ranking lists as is done in Ref. [14].

There are other general strategies that seek to build performance approximations in a systematic manner. The use of an approximation function, $\tilde{\Theta}(\cdot)$, whose evaluation cost should be less that the original Θ, may manifest itself in different forms:

1. $\tilde{\Theta}^a(a, d^a) \approx \Theta^a(a, d^a)$: Individual approximations for each architecture

2. $\tilde{\Theta}(a, d^a) \approx \Theta^a(a, d^a)$: Shared approximation for all architectures

3. $\tilde{\Theta}(a) \approx \Theta'(a)$: Shared approximation for all architectures based only the architecture object (no direct determination of the design variables)

The first case constructing $\tilde{\Theta}^a(a, d^a)$ is aligned with traditional surrogate modeling [67] or multi-fidelity frameworks [68]. The second case constructing $\tilde{\Theta}(a, d^a)$ has been investigated in the evolutionary computing literature [69], where a relatively large number of fitness evaluations are usually required to achieve convergence. The third case predicting the ILP performance using only the architecture has been investigated using machine learning techniques [20]. In all cases, developing effective computational expense reduction strategies is open.

The final approach discussed here is applicable to the nested form. Understanding and leveraging the structure of the ILP can greatly decrease the computational time required to obtain $\Theta'(a)$ and therefore increase r_e. For example, obtaining analytic sensitivities can improve gradient-based optimization procedures [5, 70]. However, effectively leveraging the ILP structure can be challenging if there are many problem elements that depend on the architecture; the form of the problem elements could differ radically for candidate architectures. Furthermore, even if many of the problem elements are consistent, architecture dependencies could result in convex or nonconvex ILPs, as is shown in Ref. [30] for a vehicle suspension architecture study. All model complexities would need to be managed effectively if the evaluator is fully automated.

2.8 Searcher

The role of the searcher is to use the results from the evaluator to define search directions which encode new candidate architectures (and design variables). The choice of problem formulation, generator, and modeler all have an impact on determining search directions.

2.8.1 Search Methods

In this section, we generally present potential search methods as their applicability and effectiveness tends to be problem dependent. Comparing the applicability and effectiveness of different searchers has been discussed for some specific engineering domains, including material structure design [10, 71] and electric circuits [56].

As defined in Sec. 2.2, the most natural (but not necessarily effective) search method is an enumeration-based strategy where search directions are simply the next architecture in an ordered list of all feasible architectures. If evaluating or generating all architectures is not possible, then a stochastic approach could be utilized with an action-based generator through randomly-chosen actions, feature-directed search with internal parameter tuning [46], or Monte Carlo tree search [72]. Many popular searchers for architecture design are population-based and use metaheuristics to select search directions. For example, nature-inspired algorithms such as a genetic algorithms [73, 74], particle swarm [74, 75], simulated annealing [76], and tabu search [77] have been used to solve architecture design problems. Gradient-based methods have also be used to define search directions when using a relaxed form, such as the relaxation mentioned in Sec. 2.5.5 [5, 11]. A condition of using these methods is the ability of the evaluator to handle architectures in the relaxed set A_r.

2.8.2 Exploration vs. Exploitation

The need for exploitation is dependent on ϵ-relative metrics defined previously for generator coverage and evaluation cost. An enumeration-based method performs only complete exploration and no exploitation. On the other end of the spectrum, gradient-based methods (that have been applied to relaxed forms [9, 10]) are criticized for their large focus on exploitation and often lead to local minima [9]. Therefore, many searcher strategies that have been used for architecture design have sought to balance exploration and exploitation [45–47, 73–75, 77]. Multi-start methods are one such example [78]. A more prevalent approach is to use metaheuristic-based methods [73–77]. However, careful application is needed as these methods can struggle to converge for large-scale problems [79]. Understanding and developing methods with a successful balance between exploration and exploitation is an important research area.

2.8.3 Searcher and the Nested vs. Simultaneous Forms

Many frequently-used searchers use a population (small set of architectures) and positive reinforcement (relatively *good* performances are preferred and exploited) to select directions [44, 73]. From a normative perspective, we can consider four cases of relative performance a candidate architecture might fall under compared to the current population, shown in Table 2.1. For both formulations, if both have the same relative performance (either good or poor) compared to the current population, then a searcher would perform the same

Nested			Simultaneous	
$\Theta'(a)$ compared to current population	Searcher exploits a	Should a be exploited?	Searcher exploits a	$\Theta^a(a, \mathbf{d}^a)$ compared to current population
Good (+)	Yes	Yes	Yes	Good (+)
Good (+)	Yes	Yes	No	Poor (−)
Poor (−)	No	No	Yes	Good (+)
Poor (−)	No	Maybe	No	Poor (−)

Table 2.1: Qualitative assessment of the positive reinforcement searcher behavior using both the nested and simultaneous forms.

action. The interesting cases are when the relative performance assessments differ. If a particular instance of a with some $\mathbf{d}^a$ has poor relative performance but good relative *potential* performance using $\Theta'(a)$, then a searcher should want to exploit such an architecture. However, it may not when using a simultaneous form, which may bias the search away from architectures with the best potential performance if the current instance of the design variables happen to produce poor performance. A similar statement can be made for the converse case, and the conclusion is that the nested approach is more aligned with this view of the conditions such that the searcher should exploit the current architecture if the potential performance is good.

Such insights have been exposed by some in the electric circuit synthesis community stating that the ILP should be solved for each candidate circuit topology to provide a proper assessment of the value of the candidate architecture [18]. The building block hypothesis is one explanation of how genetic algorithms are successful, so employing a representation that can be successfully used in identifying and exploiting building blocks should be a key consideration [73]. Coupling a and $\mathbf{d}^a$, as the simultaneous form does, using a genetic algorithm should be a carefully considered decision.

One final comment is related to the architectures with poor performance in both forms. Many traditional positive reinforcement methods do not keep this information to inform future searcher actions away from known poor architectures. One strategy to fully exploit this information is to use a learning algorithm [80].

2.9 Application to Existing Case Studies

In Table 2.2, the proposed formulation and solution strategy considerations are analyzed for a select number of engineering architecture design case studies. While this selection is not comprehensive, it does highlight the diversity of solution methodologies across various application domains. Within the diversity of examples, there are common themes that become apparent based on problem formulation; such as the typical enumeration of list-type generators. This table can be seen as a starting reference for how others have formulated and solved their specific problems. Since the focus of this article was to be descriptive, a normative discussion of method utility is left as a topic of future work.

		Architecture		Generator			Evaluator			Searcher
Reference	Application	Form	Space	Type	Set	Coverage	Metric	Model	Expense	Method
Deb [12]	Truss	NAF	A_r	ADJ	$R \setminus A \neq \emptyset$	$r_g \ll 1$	$\Theta(a)$	m^a	*low*	GA
Aage [5]	Truss	NAF	A_r	ADJ	$R \setminus A \neq \emptyset$	$r_g \ll 1$	$\Theta'(a)$	m	*med*	GB
Pedro [7]	Truss	SAF	A	RUL	$R = A$	$r_g \ll 1$	$\Theta(a)$	m	*med*	GA
Zhou [22]	Powertrains	NOF	A	LST	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*high*	ENM
Ing [23]	Powertrains	NAF	A	ADJ	$R \setminus A \neq \emptyset$	$r_g = 1$	$\Theta'(a)$	m^a	*med*	ENM
Bayrak [24]	Powertrains	NAF	A	ADJ	$R \setminus A \neq \emptyset$	$r_g = 1$	$\Theta'(a)$	m^a	*med*	ENM
Silvas [25]	Powertrains	NAF	A	ADJ	$R \setminus A \neq \emptyset$	$r_g \ll 1$	$\Theta(a)$	m	*low*	CP
Das [16]	Electronics	NAF	A	ADJ	$R \setminus A \neq \emptyset$	$r_g \ll 1$	$\Theta'(a)$	m^a	*high*	GA
Guo [14]	Electronics	NOF	A	LST	$R = A$	$r_g = 1$	$\tilde{\Theta}'(a)$	-	*low*	ML
Bortis [15]	Electronics	SAF	A_r	ADJ	$R \setminus A \neq \emptyset$	$r_g \ll 1$	$\Theta(a)$	m^a	*high*	GP
Kolar [13]	Electronics	NOF	A	LST	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*high*	ENM
Chocron [32]	Mechanisms	NAF	A	RUL	$R = A$	$r_g \ll 1$	$\Theta'(a)$	m^a	*med*	GA
Hiller [33]	Mechanisms	NAF	A_r	ADJ	$R = A$	$r_g \ll 1$	$\Theta'(a)$	m	*med*	ENM
Peddada [28]	Fluid Networks	NAF	A	ADJ	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*med*	ENM
Doman [27]	Fluid Networks	NOF	A	LST	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*high*	ENM
Herber [30]	Suspensions	NAF	A	PM	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*high*	ENM
Chen [31]	Suspensions	NOF	A	LST	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*med*	ENM
Pornet [41]	Aircraft	NOF	A	LST	$R = A$	$r_g = 1$	$\Theta'(a)$	m^a	*high*	ENM
Qi [42]	Aircraft	SAF	A	ADJ	$R = A$	$r_g \ll 1$	$\Theta(a)$	m	*low*	GA
Guo [4]	Synthetic Bio.	NAF	A	ADJ	$R = A$	$r_g \ll 1$	$\Theta'(a)$	m	*med*	GA
		SAF	A_r	ADJ	$R = A$	$r_g \ll 1$	$\Theta'(a)$	m	*med*	GB
Ma [34]	Synthetic Bio.	NAF	A	ADJ	$R = A$	$r_g \ll 1$	$\tilde{\Theta}'(a)$	m	*med*	ENM

(Generators) LIST: List, ADJ: Adjacency Matrix, PM: Perfect Matching, RUL: Rule/Grammar

(Searchers) ENM: Enumeration, GA: Genetic Algorithm, CP: Constraint Programming,
ML: Machine Learning, GP: Genetic Programming, GB: Gradient Based

Table 2.2: Analysis of select references in the context of the proposed representation, considerations, and metrics.

2.10 Discussion

In this article, a unifying mathematical representation that encompasses a class of architecture design optimization problems characterized by a discrete space of potential architectures of finite size and feature complex structure and evaluation to assess performance is presented. This includes definitions of the modules of general solution strategies for architecture design problems, examples of current module strategies, and quantifiable descriptors to capture module-specific considerations. In Table 2.2, this unified problem representation (and associated qualitative and quantitative descriptors) was applied to a limited selection of existing engineering architecture case studies and may serve as a starting reference for systematic investigation of the current methods and challenges over a wide range of applications.

It is hoped that this work provides a new perspective in the area of system architecture design problems and motivates the investigation of alternative problem and appropriate formulations. The "preferred" method for structural optimization alternated through the decades between evolutionary [12] and gradient-based [5] forms depending on the state of method development and the advancement of computational resources (and generating the adjacency matrix interactions will always be impractical [11]). Enumeration-based methods have been effective in a number areas such as hybrid powertrains design [23, 24] due to the specific complexity of their graph representations and inner-loop problems. Comparisons between widely different methods can also be useful, as is shown in Ref. [56] for an electric circuit architecture problem where an enumeration-based method confirmed that a metaheuristic-based approach [19] identified one of ten equivalent Pareto-optimal architectures.

Adopting a common representation for evaluating the performance of architectures enables a fairer discussion relating to problem formulation and solution strategies. Some of the foundational work could include: 1) quantifying the scalability and coverage of specific generators, and ways to improve the generation process; 2) effectively navigating the fidelity trade-offs in the various modules, especially in the context of complex inner-loop problems; 3) investigating the validity and effectiveness of the searcher and search directions; and 4) tailored solution strategies (i.e., appropriate methods for each module) applicable to certain subclasses of the considered architecture design problem class.

2.11 Summary

Architecture, or topology, optimization problems have been investigated in several research communities highlighting domain-specific challenges. In this chapter, a unifying language is presented to facilitate a discussion between research communities. There are some new key points introduced in this chapter that are not typically given emphasis in topology optimization problems.

- The distinction between an architecture, a, and associated design variables, $\mathbf{d}^a$.
- Formalizing generator coverage as a function of the realizable set of architectures and illustrating action-based generation sequences.
- Emphasizing the role of the Modeler Fig. 2.2. The modeler is a key bottleneck when considering architecture design with heterogeneous components, it is also the key enabler when shared as in SIMP structural topology optimization.

- Defining a relative expense metric for the evaluator and differentiating styles for problem approximation
- Clarifying the use of design data when using nested vs simultaneous forms in a searcher.
- Applying a these definitions across multiple application domains to draw similarities between research methods.

The next chapters in the thesis focus on relaxed topology optimization with a shared model and a simultaneous searcher.

Chapter 3

Practical Objectives and Constraints for Heat Conduction Topology Optimization

3.1 Introduction

Heat conduction is a dominant mode of heat transfer in many engineering systems. This physics phenomenon can be used purposefully to heat and cool matter through the designed use of a heat spreading structure. The research presented in this chapter focuses on the design of heat spreading structures using topology optimization methodologies, Bendsoe and Kikuchi [6]. Topology optimization has been used successfully to design heat spreaders for various applications. One similarity between the design of these heat spreaders is the use of a structural mechanics analog objective: thermal compliance. This objective is favorable from a numerical optimization solution performance perspective, but may not represent the true goal of a heat spreader. The investigations presented here will focus on reformulating the topology optimization problem for various different heat spreader design tasks.

Researchers in topology optimization have investigated several problem formulations for heat spreader design. Gersborg-Hansen et al. [81] presented a well-known example where a thermal compliance based heat conduction problem was solved using the finite volume method. Burger et al. [82] solved a 3D thermal compliance problem to evaluate the performance of optimized structures for a variety of finite fixed temperature boundary conditions. Asymmetric optimal structures were obtained when dividing and segregating the Dirichlet boundary. In a recent study, Yan et al. [83] investigated the optimality of dendritic structures for compliance and maximum temperature minimization problems. It was concluded that lamellar needle-like topologies outperform the typically reported dendritic structures. Given the local nature of gradient-based optimization, careful consideration should be given to the initial design vector value to support convergence towards better-performing optimized structures. One strategy for selecting a good initial design vector is to perform optimization using a reduced-dimension design representation, such as a generative algorithm [8]. This strategy was shown to improve design performance consistently when compared to topology optimization with a homogeneous initial material distribution.

More complex formulations that involve elements beyond a simple compliance objective and volume constraints have been investigated. For example, Mark et al. [84] investigated the tradeoff between mean temperature and temperature variance to formulate a more realistic design problem. The authors used the method of weighted sums to produce a Pareto front by calculating both combined objectives and derivatives. The optimized topological structures have clear differences when comparing mean temperature solutions to variance-optimized solutions. In another example, Dirker and Meyer [85] investigated several objectives in a problem formulation with design-dependent volumetric heat generation. Their objectives included the maximum domain temperature, the average domain temperature, and a summation of the temperature across volumetric heat generating elements. The authors observed dendritic patterns in the optimized structures for all objective functions. Utilizing more realistic objectives may improve the practical applicability of topology optimization, however, the functional role of a heat spreader may depend on additional practical constraints. For example, Zhuang and Xiong [86] solved a minimum compliance design problem incorporating point and local area temperature constraints. The fundamental structure of the optimized topology changed when considering the area temperature constraints.

As a logical extension to heat conduction problems that are commonly solved, design-dependent loading has been investigated in applications with convection. Two strategies for parameterizing convection in a conduction framework have been proposed by Bruns [87] and Iga et al. [88]. These approaches have been extended recently to 3D problems. Consider the work of Zhou et al. [89], where a static conduction-based convection topology optimization was performed and compared to a benchmark heat spreader design. The authors observed a $20°C$ decrease in temperature when comparing the optimized design to the reference design using thermo-fluidic simulations. Another example is in the work of Dede et al. [90], where conduction-based convection optimization was performed to design a heat spreader for a circular air jet. The authors fabricated the optimized topology and experimentally compared performance to standard heat spreader topologies. The optimized heat spreader showed comparable performance to conventionally designed heat spreaders.

Fully coupled thermo-fluidic analysis has also been solved for in many cooling design applications. For forced air convection, a 3D to 2D model reduction was used to design fin topology, Haertel and Nellis [91], using COMSOL Multiphysics. The lower-order model results agreed with 3D simulation, and a decrease in thermal resistance of 13% was achieved through optimization. A full 3D thermo-fluid topology optimization including buoyancy effects was performed by Alexandersen et al. [92]. It was shown that complex geometries can improve cooling performance when compared to straight fin heat spreaders. In a recent paper, Dbouk [93] asserted that topology optimization methodologies are not yet robust enough to be used in industrial applications. Though this is true for some formulations, there are many applications where industry-relevant problems can be solved well using topology optimization methods. Take for example this heat conduction dominant system where topology optimization has been practically used to achieve a large reduction in domain temperature, Dede et al. [94]. In the research presented here, we investigate a similar application: power electronic systems.

In electronic systems spreading structures are often selected based on commercially-available products. Recent work has demonstrated the interesting opportunity of employing heat spreaders tailored using optimization to further improve performance for specific applications. Consider the work of Christen et al. [95] where a thermal resistor network was used to model heat spreaders for natural convection. Using optimization, the authors reduced the heat spreader size by 50% and experimentally verified performance. As another example, a radial heat spreader was designed for an LED by Yu et al. [96] solving the Navier-Stokes equation. Different styles of radial heat spreaders were optimized using evolutionary algorithms to explore the effects of fundamental design changes. Moving towards a more flexible formulation, Ramphueiphad and BUreerat [97] optimized both fin cross-section and height for a 3D heat spreader using a multi-objective evolutionary algorithm. This study revealed that non-uniform fin cross-sections and fin heights are superior to pin fin designs with a constant fin height. In our research, we utilize topology optimization to provide maximal design freedom, motivated by the positive correlation between design flexibility and system performance observed in the literature across these multiple application domains.

The heat conduction topology optimization problems as previously discussed translate to electronics applications where heat generating devices also must satisfy strict temperature constraints. In this work, two different sets of topology optimization problems are investigated. First, an investigation of optimization objective functions is presented on a benchmark heat transfer topology optimization problem. The findings are applied to several practical problem formulations for power electronics on a domain with discrete heat sources. The following sections will describe the topology optimization methodology and present numerical results for all problem formulations.

3.2 Topology Optimization Methodology

To address the topological design of heat spreaders, a density-based topology optimization approach is used. The design domain is discretized into finite elements to match the analysis domain. The analysis problem is solved using a standard Galerkin finite element procedure with linear shape functions. A design vector, $\mathbf{d}$, is comprised of parameters, α_i, assigned to each finite element and is used to scale the material densities of the elements. In this study, the solid isotropic micro-structure with penalization (SIMP) method is used to bias the elements towards solid (1) or void (0) material properties. In this implementation the SIMP penalization, γ, is applied on the filtered density field, $\tilde{\rho}$, to scale the the thermal conductivity, κ_i, of a finite element, i,

$$\kappa_i = \kappa_{\min} + \tilde{\rho}_i{}^{\gamma}\kappa_0. \tag{3.1}$$

The penalization parameter, γ, is increased at intervals to improve optimization convergence. The elemental thermal conductivity is allowed to vary between, $\kappa_{\min} = 1$W/mK, and the sum of $\kappa_{\min}$ and $\kappa_0 = 400$ W/mK. To enforce the minimum length-scale requirements, a linear density filter is used.

$$\tilde{\rho} = \sum_j w_{ij}\alpha_j, \tag{3.2}$$

where the weighting function, w, is defined based on the centroid distances of neighboring elements in the neighborhood N_i:

$$w_{i,j} = \begin{cases} \frac{R-d(i,j)}{\sum(R-d(i,k))} & j \in N_i \\ 0 & j \notin N_i. \end{cases} \tag{3.3}$$

The filter weight, w_{ij}, is scaled based on the difference between a prescribed minimum radius, R, and the distance between element centers $d(i,j)$, and a summation of neighboring element distances, $d(i,k)$. This simple filter is effective at biasing the designs to satisfy the minimum radius constraint. There are many other filtering strategies that can be used to influence design representation; refer to Svanberg and Svard [98] for a helpful study of filters. The method of moving asymptotes (MMA) algorithm [99] is used to optimize the nonlinear design optimization problem until a convergence tolerance of 0.01 on design variable changes, or a maximum of 100 outer loop MMA iterations is reached.

A custom finite element analysis code was written in MATLAB to analyze thermal properties on the domain using four node elements with linear shape functions. A regular mesh of 300 × 300 elements was chosen for both design and analysis as it produced $< 0.1\%$ error in the maximum temperature when compared to a benchmark COMSOL analysis. A density filter is used where appropriate to enforce length-scale control on the domain, and the adjoint method is used to obtain gradient information for the objective and constraint functions. These definitions are consistent across all presented case studies unless otherwise noted. The adjoint method is detailed next.

3.2.1 Adjoint Differentiation

The derivative of the objective function is obtained using the adjoint method. A discretize-then-optimize approach is used to obtain the gradient of the objective function. Consider an objective function, Θ, which can be represented by some function, $\Pi(\cdot)$:

$$\Theta(\mathbf{d}) = \Pi(\mathbf{U}(\mathbf{d}), \mathbf{P}(\mathbf{d}), \mathbf{d}), \tag{3.4}$$

that depends on the thermal load, $\mathbf{P}$, temperature, $\mathbf{U}$, and the design variable vector, $\mathbf{d}$. The residual of the finite element analysis is multiplied by an adjoint variable $\boldsymbol{\lambda}$ and is added to this objective:

$$\Theta(\mathbf{d}) = \Pi(\mathbf{U}(\mathbf{d}), \mathbf{P}(\mathbf{d}), \mathbf{d}) + \boldsymbol{\lambda}^T \left[\mathbf{K}(\mathbf{d})\mathbf{U}(\mathbf{d}) - \mathbf{P}(\mathbf{d})\right]. \tag{3.5}$$

Note that $\mathbf{K}(\mathbf{d})$ is the stiffness matrix for the finite element analysis. The gradient of this expression can be obtained using the chain rule:

$$\begin{aligned} d\Theta(\mathbf{d}) = \frac{\partial \Pi}{\partial \mathbf{U}(\mathbf{d})}\frac{\partial \mathbf{U}(\mathbf{d})}{\partial \mathbf{d}} + \frac{\partial \Pi}{\partial \mathbf{P}(\mathbf{d})}\frac{\partial \mathbf{P}(\mathbf{d})}{\partial \mathbf{d}} + \frac{\partial \Pi}{\partial \mathbf{d}} \qquad & (3.6a) \\ + \boldsymbol{\lambda}^T \left[\mathbf{K}(\mathbf{d})\frac{\partial \mathbf{U}(\mathbf{d})}{\partial \mathbf{d}} + \frac{\partial \mathbf{K}(\mathbf{d})}{\partial \mathbf{d}}\mathbf{U}(\mathbf{d}) - \frac{\partial \mathbf{P}(\mathbf{d})}{\partial \mathbf{d}}\right]. \qquad & (3.6b) \end{aligned}$$

These terms can be rearranged to suggest useful choices for the adjoint vector:

$$\begin{aligned} d\Theta(\mathbf{d}) = \left(\frac{\partial \Pi}{\partial \mathbf{U}(\mathbf{d})} + \boldsymbol{\lambda}^T\mathbf{K}(\mathbf{d})\right)\frac{\partial \mathbf{U}(\mathbf{d})}{\partial \mathbf{d}} \qquad & (3.7a) \\ + \left(\frac{\partial \Pi}{\partial \mathbf{P}(\mathbf{d})} - \boldsymbol{\lambda}^T\right)\frac{\partial \mathbf{P}(\mathbf{d})}{\partial \mathbf{d}} \qquad & (3.7b) \\ + \frac{\partial \Pi}{\partial \mathbf{d}} + \boldsymbol{\lambda}^T\frac{\partial \mathbf{K}(\mathbf{d})}{\partial \mathbf{d}}\mathbf{U}(x). \qquad & (3.7c) \end{aligned}$$

Based on the problem formulation, the adjoint vector, $\boldsymbol{\lambda}$, can be chosen such that expensive gradient calculations are avoided. For example, if the problem does not have design-dependent loading, Eqn. (3.7b) drops out. Simplifications such as this become apparent when considering the boundary conditions in the problem formulation.

3.2.2 Boundary Definiteness

Using a filter effectively biases the design towards satisfying the minimum radius constraint and makes the design solution independent of the analysis mesh size. However, using a filter also results in a design with an ill-defined solid/void interface. This interface is defined by elements with partial density. A strategy to measure the boundary definiteness was proposed by Dirker and Meyer [85] and is presented in the following equation:

$$\epsilon = 1 - \frac{\text{mean}(\tilde{\boldsymbol{\rho}}^2 - \tilde{\boldsymbol{\rho}})}{V_{\text{max}}^2 - V_{\text{max}}}, \tag{3.8}$$

where the boundary definiteness, ϵ, is measured as the difference between 1 and a function of the partially defined material. This function compares the fraction of elements which are partially defined to the fraction of elements which are conductive, V_{max}. This measure of boundary definiteness will be used in the following sections to compare the boundary quality of the heat spreading structures.

3.3 Objective Function Investigations

This first set of studies involves designing a heat spreading structure using three sets of boundary conditions for a 1x1 m^2 homogeneously-heated domain (Fig. 3.1). This analysis problem was chosen to showcase the effect of objective function selection for a design problem that is commonly solved in the literature. This design domain can serve as an abstraction for a system with some distribution of heat on the domain; solving for discrete sources, as are common in power electronic systems, is a simple extension of this, as will be demonstrated in the next section.

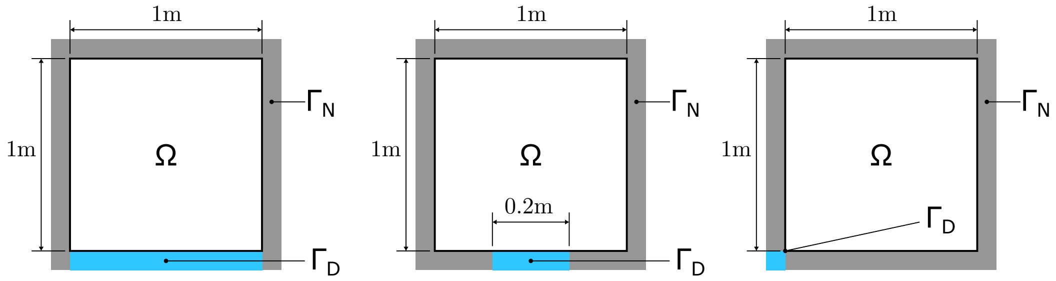


Figure 3.1: Three insulated homogeneously heated domains with fixed temperature boundaries. A fully-fixed boundary (left), a partially-fixed boundary (center), and a point-fixed boundary (right) are shown.

The design domain, Ω, is homogeneously heated, the Neumann boundary (left, right, and top boundaries), Γ_N, is adiabatic, and the Dirichlet boundary (bottom boundary, or point), Γ_D, has fixed temperature. These properties can be expressed as:

$$\nabla \cdot (\kappa \nabla T) + f = 0 \text{ on } \Omega, \tag{3.9}$$

$$T = 0 \text{ on } \Gamma_D, \tag{3.10}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{3.11}$$

In this design problem, a constant power of 1,000 W is applied uniformly to the design domain, independent of the design parameters. A conductive heat spreader is optimized to extract heat from this domain through the fixed temperature boundary. In application, a secondary heat exchanger may be present to maintain the lower boundary at a fixed temperature. In all of the following examples, this boundary temperature is fixed at $0°C$.

To achieve the goal of reducing the temperature on the domain, consider this commonly used topology optimization design problem formulation presented in (4.10):

$$\begin{aligned} \min_{\mathbf{d}} \quad & \Theta(\mathbf{d}) \\ \text{s. t.} \quad & V(\mathbf{d}) \leq V_{\max} \\ & R(\mathbf{d}) \geq R_{\min} \\ & 0 \leq \mathbf{d} \leq 1, \end{aligned} \tag{P1}$$

where some objective, Θ, is minimized subject to a volume constraint. The volume, V, is constrained by a maximum value, $V_{\max}$, and a minimum radius constraint is imposed where the radius of design features, R, must be greater than a prescribed value, $R_{\min}$. For 2-dimensional examples, the volume constraint translates to an upper limit on area, constraining the amount of conductive material used. For this problem formulation, conductive material is restricted to 30% of the domain area. The minimum radius constraint is implicitly satisfied via the density filter described in the previous section; the filtering radius is set to 0.04 m. This formulation (P1) works particularly well for structural topology optimization, where a volume constraint may represent a cost requirement and the minimum feature size of a structure may be restricted by manufacturing technology, (Patterson et al. [100]). Furthermore, the formulation is straightforward to implement and fairly numerically stable when using a SIMP topology optimization approach with the MMA algorithm. It is important to note that the actual numerical optimization problem the MMA algorithm solves does not include the radius constraint, as this is implicitly satisfied using the density filter.

3.3.1 Candidate Objective Functions

The first objective of interest is thermal compliance. This objective is defined as the product of the heat flux, q and temperature gradient, ∇T:

$$\Theta(\mathbf{d}) = \int_{\Omega} qTd\Omega \rightarrow \Theta(\mathbf{d}) = \mathbf{U}^T\mathbf{P}. \tag{3.12}$$

The compliance metric may be represented simply by the dot product of the thermal load, $\mathbf{P}$, and the nodal temperature vector, $\mathbf{U}$. This is a popular objective function because the adjoint variable can be defined directly using a known quantity $\mathbf{U}$, eliminating the need to solve an additional linear system for the adjoint variable.

A mean temperature objective is an alternative to the compliance objective that may align better with the goal of reducing the domain temperature. The thermal compliance objective biases the performance of the structure in areas where there is input heat flux. A mean temperature objective does not explicitly account for the influence of the loading conditions:

$$\Theta(\mathbf{d}) = \frac{1}{n}\int_{\Omega} Td\Omega \rightarrow \Theta(\mathbf{d}) = \frac{\mathbf{L}^T\mathbf{U}}{n}. \tag{3.13}$$

In discrete form, the mean temperature objective can be represented—in terms of finite element model quantities—as the product of a vectors of ones, $\mathbf{L}$, and the nodal temperature vector, $\mathbf{U}$, divided by the number of finite element nodes n.

Often, the goal of a heat spreader is to help reduce the maximum temperature on the domain. If point-wise hot spots are a primary concern, then minimizing maximum temperature may be a more appropriate objective than thermal compliance or mean temperature. The max function, however, is not differentiable. As an approximation to the max function, a p-norm approximation is used here:

$$\Theta(\mathbf{d}) = ||T||_p \rightarrow \Theta(\mathbf{d}) = \left(\mathbf{L}^T\mathbf{U}^p\right)^{1/p}. \tag{3.14}$$

The p-norm approximates the maximum temperature by applying a root to the power p on the summation of temperature to the power p. The accuracy of this approximation increases with the magnitude of p, and as $p \rightarrow \infty$ the norm converges to the max function. The discrete problem can be described similarly, where the summation is replaced with the product of a ones vector, $\mathbf{L}$, and the nodal temperature vector, $\mathbf{U}$, raised to the power p. To maintain a smooth approximation function with well-defined derivatives, $p = 10$ is chosen. This will not result in an accurate measure of the maximum temperature on the entire domain, but the gradient of which should direct the topology towards a "lower" temperature solution.

While minimizing temperature is an important goal of heat spreader design, some situations involving electronics may favor temperature uniformity, where large temperature gradients adversely affect performance. While the max or mean temperatures provide some useful information about the temperature distribution on the domain, these single parameters do not capture more global information. One way to quantify temperature homogeneity across the domain is to use temperature variance as an objective:

$$\Theta(\mathbf{d}) = \frac{1}{n}\int_\Omega (T - T_{\text{av}})^2 \, d\Omega \tag{3.15a}$$

$$\rightarrow \Theta(\mathbf{d}) = \frac{1}{n}\left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n}\right)^T\left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n}\right), \tag{3.15b}$$

where the temperature variance is given by the average of the squared difference of an element temperature, T, and the average temperature, T_{av}. This can be represented in discrete form in terms of the ones vector, $\mathbf{L}$, and the temperature vector **U**.

3.3.2 Optimization Results

The optimized designs for the homogeneous heating design problem considering all optimization objectives and boundary conditions are presented in Table 3.1.

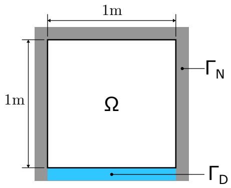
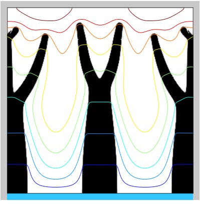
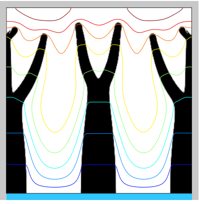
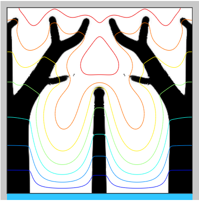
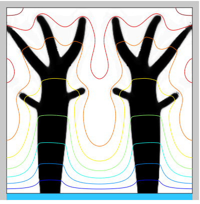
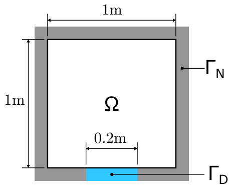
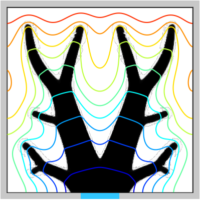
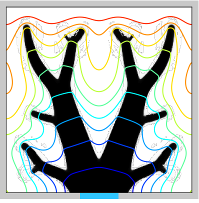
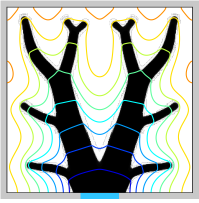
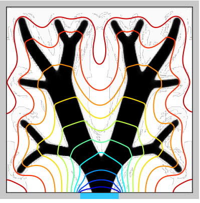
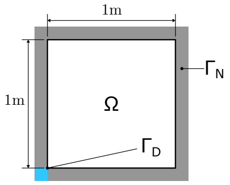
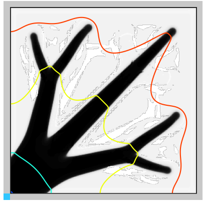
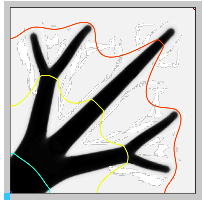
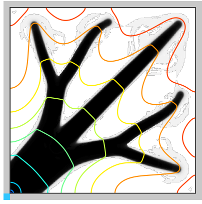
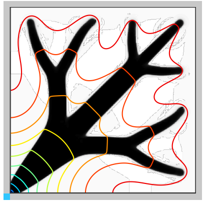
Boundary	Compliance	mean(T)	max(T)	var(T)
				
compliance	3108 (**best**)	3120 (<1%)	3616 (+16%)	4107 (+32%)
mean(T)	4.06 (**best**)	4.06 (**best**)	4.52 (+10%)	5.00 (+19%)
max(T)	8.51 (+11%)	8.51 (+11%)	7.54 (**best**)	8.22 (+8%)
var(T)	4.49 (+21%)	4.49 (+21%)	3.84 (+08%)	3.52 (**best**)
ϵ	66.85%	76.98%	75.72%	71.78%
				
Compliance	**4065 (best)**	4088 (+1%)	4368 (+7%)	7150 (+76%)
mean(T)	**5.075 (best)**	5.1274 (+1%)	5.295 (+4%)	8.065 (+59%)
max(T)	9.591 (+16%)	9.733 (+17%)	**8.290 (best)**	10.81 (+30%)
var(T)	5.8699 (+55%)	6.0431 (+60%)	4.3272 (+15%)	**3.776 (best)**
ϵ	74.74 %	74.26%	73.74%	67.68%
				
Compliance	**15500 (best)**	15824 (+2%)	15526 (<1%)	18884 (+22%)
mean(T)	16.699 (<1%)	17.40 (+5%)	**16.64 (best)**	19.95 (+18%)
max(T)	24.19 (+13%)	25.01 (+17%)	**21.39 (best)**	23.67 (+11%)
var(T)	14.31 (+93%)	15.73 (+112%)	9.039 (+22%)	**7.431 (best)**
ϵ	65.51%	62.58%	71.51%	69.26%

Table 3.1: Topology optimization solutions for various objective functions. Solutions share a common color scale for each boundary condition.

The optimizer converged to the best solution for the given objective. The computational expense using these objective functions in optimization is comparable since they share the same problem formulation and an equal number of operations are required to numerically obtain the derivatives. All four of the objective functions reached the maximum iteration limit without reaching any MMA inner-loop iteration limits. When optimizing for a particular objective, performance improvements in the range of 1-112% were observed between solutions. The potential performance improvement dwarfs any variations in computational expense that are experienced from the use of the MMA algorithm. As such, the use of proxy objective functions in an effort to reduce computational expense is not recommended.

The compliance and mean temperature optimization routines converged to nearly identical optimized topologies for all three boundary condition variations. Investigating the mathematical formulation of these objectives reveals that they are linearly proportional. The mean temperature objective is a summation of temperature values divided by a constant. For the homogeneously heated domain design problem, compliance is also the summation of temperature, but multiplied by a constant (thermal load). For this specific design problem where the load is uniformly distributed, the thermal compliance derivatives point in the same direction as the mean temperature derivatives during optimization. For general non-uniformly distributed thermal loading, the mean temperature objective should provide an alternate topology. Furthermore, it can be concluded that the compliance objective function for this problem formulation should be used in place of the average temperature objective for computational efficiency if minimum average temperature is the preferred objective.

When using an approximate maximum temperature objective function, temperature reductions of $13 - 30\%$ were observed when compared to topologies optimized by other objectives. The reduction in domain temperature may justify the use of this objective, even though the single global approximation function loses accuracy for large mesh sizes as observed by [101]. Furthermore, the inaccuracy of the p-norm approximation will propagate to the derivatives. However, the p-norm function will over-estimate the value maximum temperature and is monotonic for values greater than zero. These characteristics of the approximation function will point the derivatives towards a “lower” temperature solution, which may not represent the “lowest” temperature design, directly. An accurate maximum temperature value may be calculated directly using the maximum function after the optimization solution is obtained.

The optimized structures when using the variance objective tended to be more dendritic. Conductive material is allocated such that the non-conductive white space is more evenly distributed on the domain. Furthermore, in all three cases, the contact surface to the fixed temperature boundary seems to be smaller than those for solutions obtained when using the other objective functions. To analyze these effects, a normalized gradient is superimposed on the compliance and variance optimized topologies for the first boundary condition in Fig. 3.2.

Notice how the compliance solution (left) has higher gradient concentrations, denoted by the lighter contours, near the black structure. These are located at the end points of the conductive structure. The variance optimized solution does not feature the same magnitude of gradients at the endpoints of the structure, and its highest gradients are located near the heat sink. This may be due to the reduced presence of conductive material (i.e., inverse taper) near the fixed temperature boundary. Though this objective function can be successfully used to minimize the variance of the temperature on the domain, it does

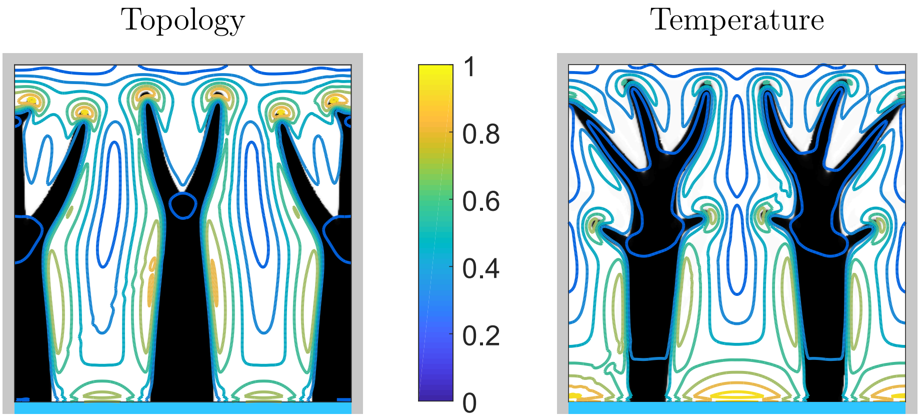


Figure 3.2: Temperature gradient comparison between compliance (left) and variance (right) minimization solutions. Normalized gradients share a color scale. Variance optimized solution moves largest variance gradients towards heat sink.

not minimize the average temperature about which the variance is calculated. In some cases, the temperature rose significantly, 30%, in order to produce a solution with a minimized variance. To use this objective function more effectively for electronics, it should be paired with some form of temperature constraint. An example formulation is presented in the next section.

3.4 Practical Formulations for Power Electronic Systems

With a practical understanding obtained from analyzing different objective function results in the previous section, findings are applied to case studies relevant to power electronics applications. Consider the 10 x 10 cm^2 design domain consisting of 8 discrete 5 mm x 2.5 mm heat sources, illustrated in Fig. 3.3. The heat sources centers are spaced evenly such that a = 2.5 cm and b=2 cm. One application that would produce such a domain is an electronics circuit board where heat is generated by electronic devices, Dede et al. [94]. This type of system has additional design considerations, including devices with distinct properties (different temperature limits, etc.). A variety of problem formulations to address this design problem class are investigated in the following sections.

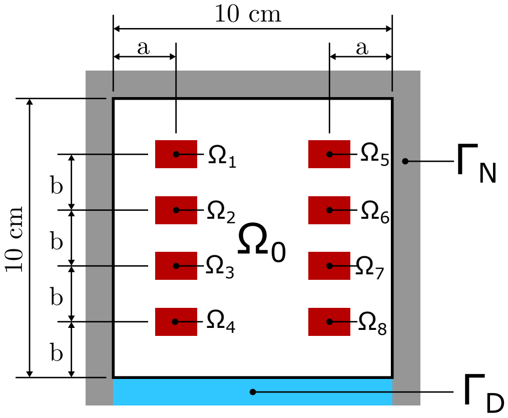


Figure 3.3: Simplified 2-D printed circuit board (PCB) domain with 8 heat generating devices.

The governing equations of this system involve heat generation prescribed in specific regions, $\Omega_{1,\ldots,8}$. In these regions, the heat generating components are assumed to be thermally conductive. The design domain, Ω_0, does not generate heat, and thermally conductive material distribution will be designed via the optimization process to extract heat. The governing equations of this system follow:

$$\nabla \cdot (\kappa \nabla T) = 0 \text{ on } \Omega_0, \tag{3.16}$$

$$\nabla \cdot (\kappa \nabla T) + f = 0 \text{ on } \Omega_{1,\ldots,8}, \tag{3.17}$$

$$T = 0 \text{ on } \Gamma_D, \tag{3.18}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{3.19}$$

For these studies an input power, $f = 1.25$ W, is applied to each of the eight rectangular devices on the domain, for a total of 10 W of loss within the domain. The Dirichlet boundary (bottom boundary) remains fixed at 0°C, and the Neumann boundary (remaining boundaries) restrict heat flux out of the domain. Using a fixed temperature boundary, where the device and heat sink will be connected by a conductive bridge, small temperature variations between device and sink are expected, and were observed for all case studies. Though this accurately represents some electronics systems, it is not true for all electronics systems. For all subsequent studies the Dirichlet boundary is retained and constraints are modified to account for this boundary condition for demonstration purposes.

3.4.1 Maximum Temperature Minimization

With the recent advances in wide-band gap devices, the temperature difference between heat source and sink may vary by 100's of degrees Celsius during steady state operation. The accuracy of the p-norm approximation is dependent on the magnitude of the norm value, and for large temperature variations on the domain numerical issues can be experienced. A similar situation is well studied in structural mechanics, were stress is an important parameter. In many cases, stress is enforced in a normalized constraint to avoid these numerical issues, Le et al. [102]. In this research, we adapt this concept in a novel way to perform a normalized temperature minimization. One strategy to normalize the temperature measurements is to optimize the maximum temperature constraint value on the domain while enforcing a temperature constraint. The problem formulation (P2) is:

$$\begin{aligned} \min_{\mathbf{d}} \quad & d_1 \\ \text{s. t.} \quad & V(\mathbf{d}) \leq V_{\max} \\ & R(\mathbf{d}) \geq R_{\min} \\ & \left(\mathbf{L}^T_{1,\ldots,8} \frac{\mathbf{U}^p}{d_1} \right)^{1/p} - 1 \leq 0 \\ & 0 \leq \mathbf{d} \leq 1, \end{aligned} \tag{P2}$$

with the design vector defined as:

$$\mathbf{d} = \left[T_{\max}, \mathbf{d}\right]^T = \left[T_{\max}, \alpha_1, ..., \alpha_n\right]^{\mathrm{T}}. \tag{3.20}$$

Here the maximum temperature constraint value, d_1 is minimized, subject to volume ($\leq 30\%$), radius (≥ 0.4 cm), and a single normalized p-norm temperature constraint. To prevent numerical oscillations resulting from the temperature constraint switching between active and inactive, a move limit of 0.01 is applied to the maximum temperature design variable. Device locations are fixed and the global temperature constraint is only applied to device nodes, assuming that other materials in the system are less sensitive to temperature. If instead all points across the domain were constrained to be less than $T_{\max}$, instead of just the devices, this problem would be mathematically equivalent to the formulation presented in Sec. 3 that minimized an approximated maximum temperature. The topology optimization solution for this problem is presented in Fig. 3.4.

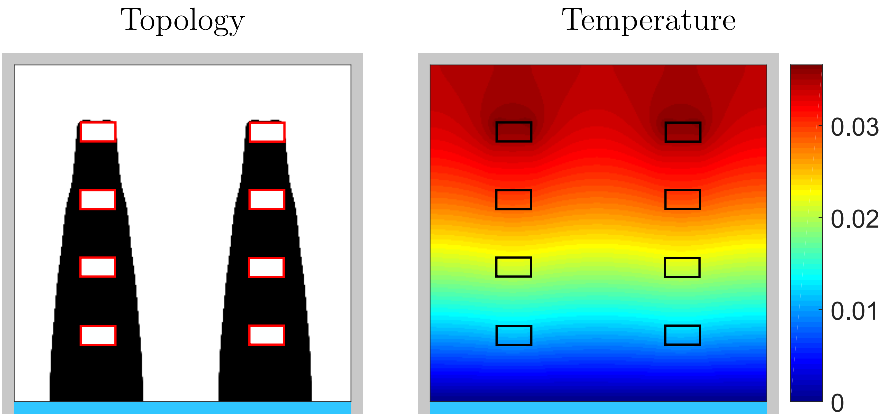


Figure 3.4: Max temperature optimization solution results in a ladder-like structure.

The MMA algorithm found a 'ladder-like' topology to reduce the maximum device temperatures. The maximum temperatures of the devices are symmetric across the central axis and are presented in order by device number: [0.037 0.030 0.022 0.012 0.037 0.030 0.022 0.012] ° C. Minimizing maximum device temperatures may make sense for electronic heat spreader design and similar problems where device lifetime and temperature are correlated.

In addition to the adaptive normalization benefits, this formulation presents increased design flexibility. For example, consider the following objective function:

$$\Theta(d) = T_{\max 1}(1 - d_1^p) + T_{\max 2} d_1^P, \tag{3.21}$$

where optimization may select an appropriate temperature constraint based on the heat spreader design task. The value for $T_{\max 1}$ is set to 0.02, which is beyond the capability of the optimal heat spreading structure, and $T_{\max 2}$ is set to 0.04, which is above the maximum temperature of the optimal heat spreading structure. This objective may serve as an abstraction of a component selection task while considering heat spreading capability. It was observed that the optimizer first tried to minimize the maximum temperature on the domain. As the optimizer could not satisfy the temperature constraint, the constraint limit was increased. The optimizer successfully pushed d_1 to a value of 1 to select the components with a higher temperature constraint. This resulted in the same topology as illustrated in Fig. 3.4.

3.4.2 Average Temperature Matching Optimization

In some electronics applications, the circuit performance may be improved by matching the temperature of specific sensitive devices. Furthermore, the optimization of electronic layouts may result in asymmetric device layout. To emulate this, this next case study was formulated where the devices nearest and furthest from the heat sink were required to have identical temperatures, while satisfying maximum temperature constraints on all eight devices. Consider the case where devices 4 and 5 (as labeled in Fig. 3.3) must operate at the same temperature for ideal performance. This formulation is appropriate for photonics systems, where device behavior is highly sensitive to temperature changes, and maintaining similar temperatures on multiple devices is required for desired performance. One approach to achieve this is to minimize a square of the difference between the average temperature of nodes corresponding to these two devices:

$$
\begin{aligned}
\min_{\mathbf{d}}.\quad & \left(\frac{\mathbf{L}_4^T\mathbf{U}}{n} - \frac{\mathbf{L}_5^T\mathbf{U}}{n}\right)^2 \\
\text{s. t.}\quad & V(\mathbf{d}) \leq V_{\max} \\
& R(\mathbf{d}) \geq R_{\min} \\
& \left(\mathbf{L}_{1,\ldots,8}^T\left(\frac{\mathbf{U}}{T_{\max}}\right)^p\right)^{1/p} - 1 \leq 0 \\
& 0 \leq \mathbf{d} \leq 1.
\end{aligned} \tag{P3}
$$

The elements of the selection vector $\mathbf{L}_i$ that correspond to nodes within device i are 1, and all other elements are zero. In addition to matching the temperature between devices 4 and 5, a general maximum temperature constraint across all eight devices must be satisfied. The maximum temperature constraint for each device is once again formulated as a p-norm applied to device node temperatures. Since the previous optimization found a minimum max temperature of 0.04° C, this formulation relaxes the requirement to a known feasible $T_{\max} = 0.06°$ C to demonstrate the capability of the procedure. This design problem is again subject to a volume constraint, where $V_{\max} = 0.3$, and a constraint on the minimum feature size, $R_{\min} = 0.4$ cm. The design variables are once again solely a function of material density parameters: $\mathbf{d} = [\alpha_1, ..., \alpha_n]^{\mathrm{T}}$. This topology optimization problem can be solved to result in the structure presented in Fig. 3.5.

The temperature of the devices are given in order as [0.525 0.474 0.042 0.059 0.510 0.038 0.262 0.157] °C. To match the temperature of a device near the temperature boundary and a device away from the temperature boundary, the optimizer allowed device 4 to heat up to match the temperature of device 5 (still satisfying the temperature constraint). The result is an asymmetric

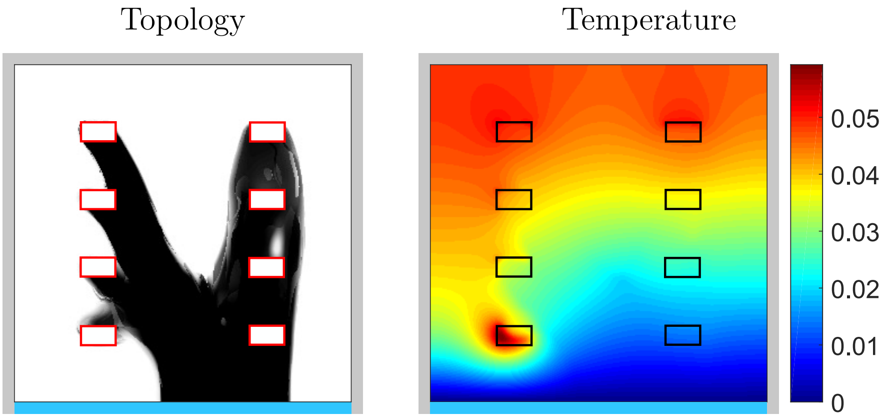


Figure 3.5: Temperature matching optimization solution produces asymmetric structure to satisfy maximum temperature constraint.

topology where device 5 is connected to the boundary with conductive material, and device 4 is minimally connected with thermally conductive material.

3.4.3 Average Temperature Maximization

In this study, we demonstrate a practical use of an average temperature optimization for an energy harvesting application subject to localized temperature constraints. This may be important for circuits with thermo-electric devices that can harvest energy [103], or for circuits placed near energy storage devices that require higher temperatures to operate efficiently. One example of this type of design problem can be formulated as:

$$\begin{aligned}
\min_{\mathbf{d}} \quad & -\frac{1}{n}\left(\mathbf{L}_2^T\mathbf{U} + \mathbf{L}_6^T\mathbf{U}\right) \\
\text{s. t.} \quad & V(\mathbf{d}) \leq V_{\max} \\
& R(\mathbf{d}) \geq R_{\min} \\
& \left(\mathbf{L}_{1,\ldots,8}^T\left(\frac{\mathbf{U}}{T_{\max}}\right)^p\right)^{1/p} \leq 1 \\
& \frac{1}{n}\left(\mathbf{L}_2^T\mathbf{U} - \mathbf{L}_6^T\mathbf{U}\right)^2 \leq 0.01 \\
& 0 \leq \mathbf{d} \leq 1.
\end{aligned} \tag{P4}$$

The objective now is to minimize the negative sum of the average temperatures on devices 2 and 6. A p-norm temperature constraint is enforced on the elements of the remaining devices to prevent failure. The temperature constraint is further relaxed here to 0.08° to demonstrate the variation in topology. In addition to these constraints, an additional temperature constraint is placed on devices 2 and 6 to ensure they operate at the same temperature. The same constraint values for volume and radius are prescribed: $V_{\max} = 0.3$ and $R_{\min} = 0.4$ cm, respectively. The results of this topology optimization are presented in Fig. 3.6.

As expected, the heat spreader design avoids devices 2 and 6 while connecting the heat sink directly to the remaining devices. The maximum temperature on devices 2 and 6 was $0.15°C$, and the maximum temperature on the remaining devices was $0.8°C$. The optimizer was able to nearly double the maximum temperature on the domain while maintaining the temperature constraint on the remaining devices. Due to layout and boundary conditions symmetry, the fourth constraint is somewhat redundant.

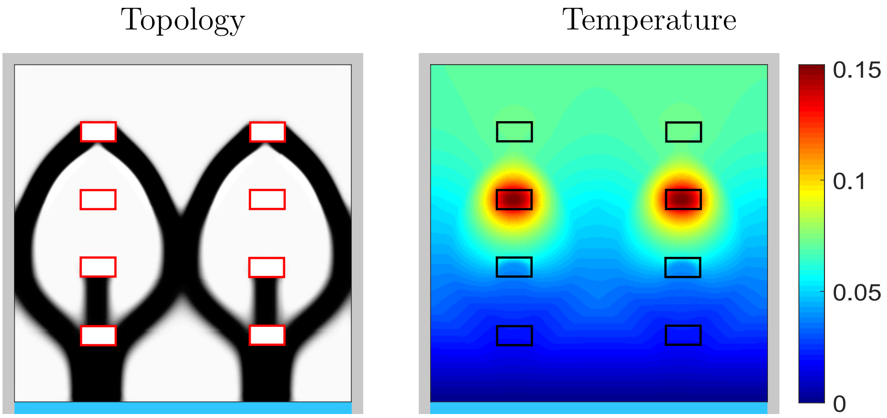


Figure 3.6: Constrained max temperature optimization solution produces a heat spreading structure that avoids specific devices.

3.4.4 Temperature Variance Optimization

In section 3, it was observed that the temperature variance objective may increase the maximum temperature on the domain to minimize the variance of the temperature on the domain. For this next problem formulation, we demonstrate a practical use of temperature variance minimization subject to a localized temperature constraint. Consider the following problem formulation:

$$\begin{aligned} \min_{\mathbf{d}} \quad & \left[\frac{1}{n} \left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n} \right)^T \left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n} \right) \right]_{1,\ldots,8} \\ \text{s. t.} \quad & V(\mathbf{d}) \leq V_{\max} \\ & R(\mathbf{d}) \geq R_{\min} \\ & \left(\mathbf{L}^T_{1,\ldots,8} \left(\frac{\mathbf{U}}{T_{\max}} \right)^p \right)^{1/p} - 1 \leq 0 \\ & 0 \leq \mathbf{d} \leq 1, \end{aligned} \tag{P5}$$

where the variance in temperature across all devices is minimized, subject to a volume, minimum radius, and a single temperature constraint across all device nodes. The maximum allowable temperature here is relaxed to 0.06 $^\circ C$ to allow room for the optimizer to rearrange material. The same constraint values for volume and radius are prescribed as $V_{\max} = 0.3$ and $R_{\min} = 0.4\text{cm}$. The optimized structure is presented in Fig. 3.7.

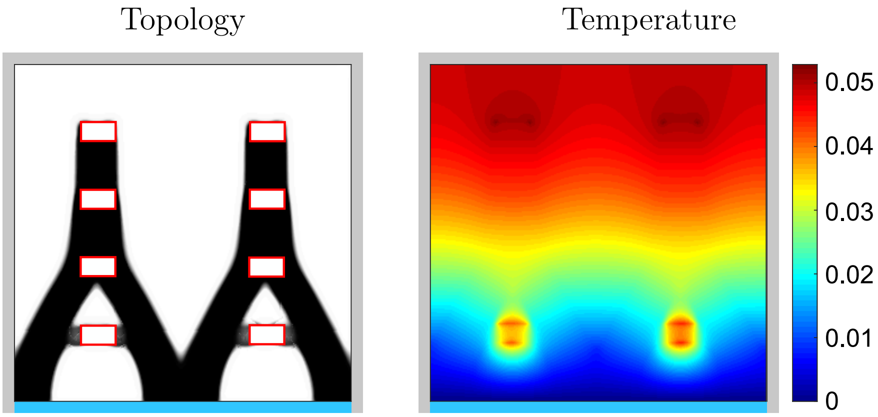


Figure 3.7: Variance-optimized solution subject to a global temperature constraint. Optimized structure maintains $0.1^\circ C$ temperature constraint.

The optimization algorithm converged to a solution which connected most of the devices together to minimize the variance while satisfying the temperature constraint. The conductive structure formed a partial-density connection to the components closest to the fixed temperature boundary to maintain the temperature constraint. This may be achieved practically by selecting a different material, with lower thermal conductivity, to interface the devices.

3.5 Discussion

Though there are some technical limitations for topology optimization as described by Dbouk [93], topology optimization may be used creatively to achieve many design tasks. Furthermore, these studies motivate a concept of designing a temperature distribution for electronics applications. This distribution can be imposed with several objective and/or constraints, instead of optimizing a single global parameter that describes performance.

In this chapter, several objective functions relevant to power electronics applications have been investigated. It was shown that thermal compliance and mean temperature optimized structures are nearly identical for a homogeneously heated design domain. This is due to the mathematical similarity between objective functions. Using a p-norm approximation of the maximum function can be used to consistently produce “lower-” temperature solutions, despite the inaccuracies in the gradient calculation. The variation-optimized solutions produced more dendritic structures with relatively small connections to the fixed temperature boundary. These solution were observed to raise the maximum temperature on the domain to enable minimization of the variance in temperature on the domain. These findings were used to design several practical formulations for power electronics systems.

The second design problem considered a discrete heating domain, corresponding to systems such as an electronics printed circuit board. For this design problem, several temperature constraints where introduced to capture realistic requirements for PCB applications. Formulations including design-dependent constraints, asymmetric loading, and temperature maximizations were considered. In utilizing various combinations of objectives and constraints, a temperature distribution on the domain was designed. Framing the optimization problem as designing a specific temperature map instead of a minimizing temperature may lead to improved system-level performance in power-dense applications.

One of the key advantages of topology optimization is the ability to represent various topologies and solve for their performance on a finite element mesh without remeshing. The design variables do not have to be restricted to material density scalars, as was demonstrated in this study (P2), to make further use of the advantage. Integrating additional system level characteristics into this gradient based optimization routine is left as a topic of future work.

3.6 Summary

In this chapter, a variety of problem formulations relevant to power electronics were investigated. First structural differences resultant from optimizing heat spreading structures for various objectives were analyzed on a benchmark problem. It is important to note that minimizing thermal compliance for a heat spreading structure may result in a large performance penalty in situations where minimized temperature is desired. Then several problem formulations, combinations of objectives and constraints, were introduced to capture the more realistic requirements for power electronics systems and demonstrate the capability of the topology optimization methods to accommodate the additional requirements. Furthermore, an alternative, more flexible, temperature minimization formulation was presented to demonstrate a strategy to normalize temperature measurements during optimization. This formulation also allowed for the concurrent design of heat spreading structure and the selection of appropriate electronic components. The observations made in this section influenced a similar investigation for reduced order convection models in Ch 5.

Chapter 4

Topology Optimization for Folded Inverter Concept

4.1 Introduction

The introduction of wide band-gap devices is changing the design of power electronic systems. Materials such as Gallium Nitride (GaN) and Silicon Carbide (SiC) enable high-frequency switching at low electrical loss to provide higher levels of efficiency at a system level. This leads to potential reductions in power electronics packaging [104]. In this chapter, a topology optimization approach [6] is used to design a heat sink for an automotive electronic application featuring GaN devices.

In recent years, there have been several practical examples of heat sink topology optimization which utilized additive manufacturing techniques. For both forced and natural convection, simplified models such as those proposed by Bruns [87] and Iga et al. [88] have been improved upon to produce manufacturable heat sink designs. Consider the work of Zhou et al. [89] where a simplified convection model was applied to optimize a heat sink design. The optimal solution had a large temperature rise reduction of 15% in simulation when compared to the baseline design. In another example, a convection model using a simplified hat function was optimized to generate a heat sink design for a jet-impingement cooling application [90]. The authors experimentally verified the performance of their additively manufactured design to be comparable to conventionally manufactured heat sink designs. Using simplified physics analyses such as these provide a computationally effective method for early-stage heat sink design.

As an intermediary step between simplified physics and 3D thermo-fluidic analysis, some authors used model-order reduction to enable design with thermo-fluidic interactions. This is implemented using multi-physics models in the design of heat sink cross sections, which are then extruded to full 3D geometry. In the work of Haertel and Nellis [91], the authors designed the cross section perpendicular to the airflow of a heat exchanger using topology optimization. They observed as high as 35% improvement in conductance per unit area when comparing optimized solutions with standard fin structures. In another example, Zeng et al. [105] designed the cross section parallel to the airflow for a heat sink design. Their heat sink design was manufactured using CNC machining and was characterized experimentally to verify performance predictions. A similar approach was used by Haertel et al. [106] to obtain airfoil-like fins to minimize the pressure drop of their heat sink designs.

Given the availability of modern computational resources, topology optimization can be used to design heat sinks using high-fidelity computational tools. For natural convection, a 3D topology optimization was performed to optimize heat sink considering buoyancy effects by Alexandersen et al. [107]. The heat sinks were fabricated and experimentally validated for LED lamps. In another example, both thermal and hydraulic performances were optimized to design a multi-pass heat sink for an electronic cooling application by Dede [108]. The performance of the multi-pass cooler has been validated for single-phase and two-phase fluid operation.

Heat sink design using topology optimization has been performed for a wide array of analysis models from simplified convection models to multi-physics simulations. Though a brief review of convection topology optimization is presented here, there are many additional developments in this area. Refer to the recent review paper by Dbouk for an assessment of these methods as they relate to industrial applications [93].

In this chapter, we present the application of a reduced-order topology optimization method for a novel electronics cooling configuration. A novel packaging scheme for multi-level inverters is presented in Sec. 4.2. Then a topology optimization method is defined to optimize a heat spreading structure in Sec. 4.3. Features from the optimized topological structure are used to define a visually adapted heat sink and performance is benchmarked using high-fidelity simulation against a baseline design in Sec. 4.4. Both the baseline and modified designs are fabricated and experimentally validated to compare their performance in Sec. 4.5. A discussion of findings and topics for future work are identified (Sec. 4.5.3) before concluding the chapter in Sec. 4.6.3.

4.2 Folded FCML Circuit Concept

The goal of this research endeavor is to design a compact 6.6kW bi-directional charger for automotive applications. The state-of-the-art inverter technology has a volumetric power density of 13kW/l at 2kW, and the stretch goal of this project is to achieve a power density of 20kW/l at 6.6kW of power. To achieve this high power density, a novel packing configuration is proposed for the following circuit topology presented in Fig. 4.1. The inverter stage of this charger uses a Flying-Capacitor Multi-Level (FCML) inverter topology to improve circuit efficiency. The FCML circuit topology was first proposed in Meynard and Foch [109]. High level count multilevel operation was not practically validated until recently, at the Google Little Box Challenge in 2015, where a 7-level

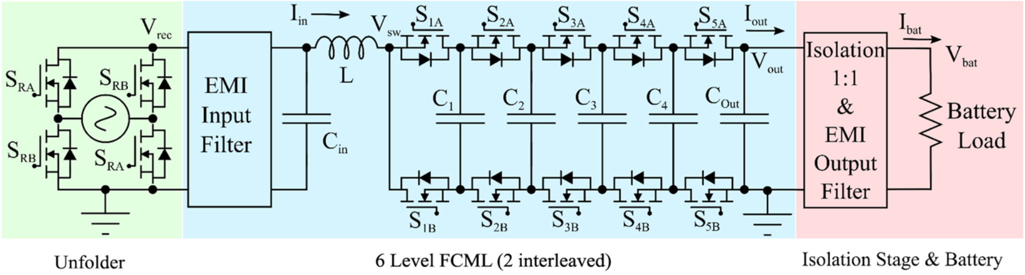


Figure 4.1: Flying-capacitor multi-level bi-directional automotive charger concept.

FCML prototype system was demonstrated at 2 kW [104]. The FCML topology distributes power processing to minimize otherwise large switching losses. This inherently increases the surface area and as such increases thermal performance. With a large surface area, minimizing packing density while satisfying temperature constraints becomes a challenge.

To increase the packing density of the inverter, a folded PCB configuration is considered here. There are several examples in the literature where flexible PCBs have been used to reduce the volume of a power electronics package [110, 111]. Since the FCML topology has been experimentally validated on a flexible PCB substrate [112], it presents a low-risk platform for development. Furthermore, the FCML circuit topology is amenable to folding due to the modularization of the switch pairs. For example, a 9-level FCML inverter would use 8 switch pairs placed into 4 switching cell modules. These can be naturally folded onto a heat sink with a square cross-section to enable heat extraction in a centralized approach, Fig. 4.2.

In the figure, the main heat generating devices are shown in red and the heat sink is shown in light gray. Cooling the FCML stage of the inverter in a planar form requires a large planar heat sink. To minimize the profile of the heat sink,

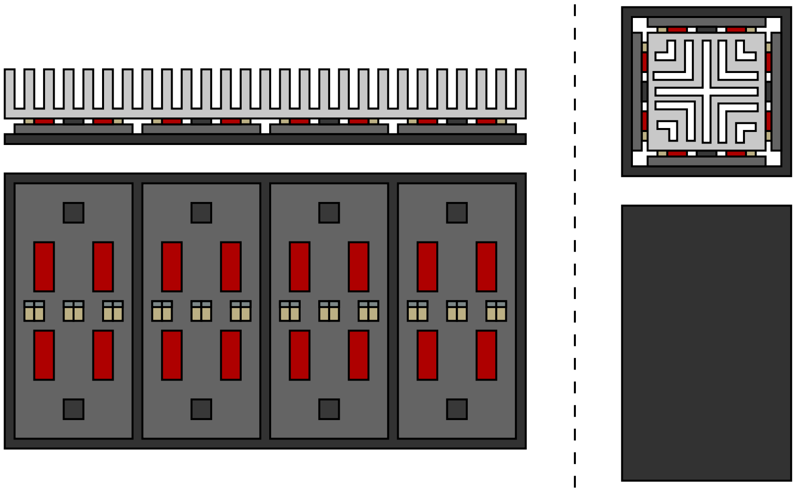


Figure 4.2: Planar PCB configuration (left). Folded PCB configuration (right).

several blower fans can be used to cool the devices. When folding the PCB in between switching cells, they can be wrapped around a much smaller heat sink structure. In this configuration, an axial fan may be used to cool the devices. The volumetric benefits of folding the PCB are presented in Fig. 4.3.

When comparing the total cooling system volume, heat sink and fan volume, a 75% reduction is obtained when using a folded circuit configuration. This large reduction in volume may be considered aggressive and an alternative, more conservative, measurement is proposed. In practice, it is common to pack the heat generating devices near the exterior of the package. This allows for the dual use of the heat sink as a cooling mechanism and as part of the enclosure to minimize electromagnetic interference (EMI). Assuming that part of the heat sink is used as the enclosure, the volume of the fins for the planar configuration will account for its heat sink volume. Under this assumption, a 40% reduction in cooling system volume is still observed when using the folded configuration.

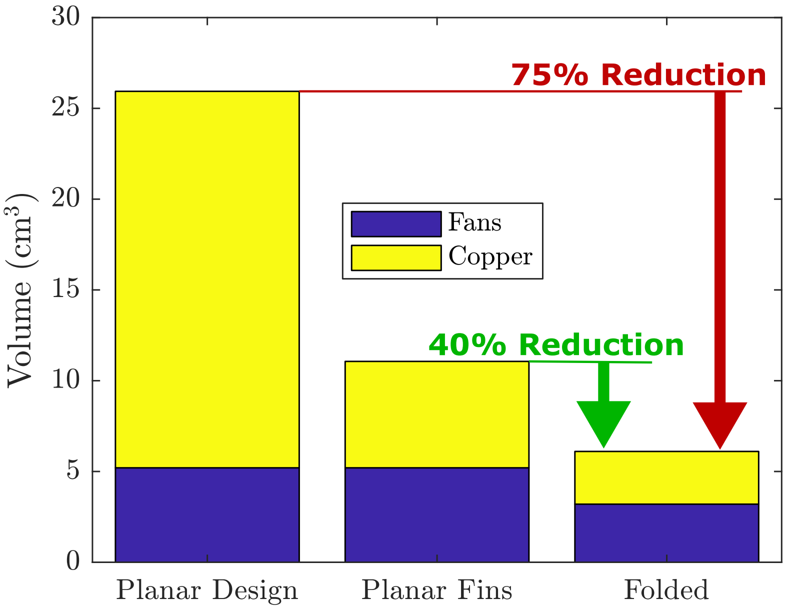


Figure 4.3: Volume reduction from folding.

In this work, topology optimization is used to generate an optimized heat sink design for the folded circuit configuration at a reduced circuit power level of 2kW. This supports low-cost validation before scaling to the 6.6kW charger design task.

4.3 Topology Optimization Methodology

At a high-level, analyzing the folded PCB configuration revealed large potential reductions in volume. This is achieved by centralizing an otherwise distributed set of heat generating devices for efficient cooling. A complementary strategy to improve the power density of the configuration is to maximize the cooling performance of the heat sink design. This will allow for additional power processing at a fixed package size. To maximize the performance of the heat sink for the folded PCB configuration, a topology optimization methodology is used. Consider the following 1/8 symmetry model of the folded PCB configuration, illustrated in Fig. 4.4.

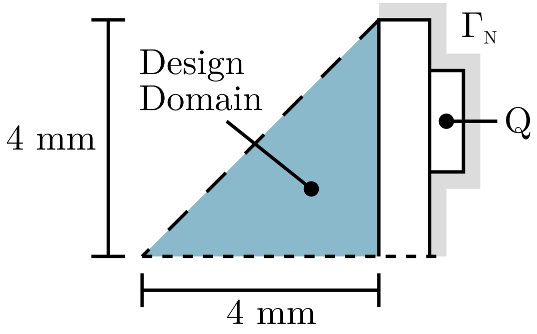


Figure 4.4: Eighth symmetry design domain for folded PCB configuration.

A total power loss of 1W is input to the device area for an effective power loss of 16W (from 16 total devices) input to the heat sink. This is comparable to the device losses of the 2.2 kW prototype for which this reduced size heat sink is designed. The device is assumed to interface directly with the heat sink and a fixed copper plate. The design domain, shown in blue, is discretized into finite elements, where each element may consist of conductive material (1) or air (0), to allow for free-form design. To make use of efficient gradient based algorithms, design vector,

$$\mathbf{d} = [\alpha_1, ..., \alpha_n]\,, \tag{4.1}$$

consist of material toggle parameters, α, which are relaxed to take intermediate values between 0 and 1 to enable the use of gradient-based algorithms. With a desire to use additive manufacturing for production, careful consideration has to be given to lengthscale. Our vendor of choice, Shapeways, offers raw silver manufacturing with a 0.6 mm minimum wall thickness constraint. To enforce a minimum wall thickness in the topology optimization procedure, a

Helmholtz filter is used. Solving the Helmholtz partial differential equation:

$$\nabla \cdot (-c\nabla \tilde{\boldsymbol{\rho}}) + a\tilde{\boldsymbol{\rho}} = \mathbf{d} \tag{4.2}$$

$$\nabla = \left[\frac{d}{dx}, \frac{d}{dy}\right], \tag{4.3}$$

will enforce a minimum feature size on the filtered element density, $\tilde{\rho}$. The parameter a is set to 1 and the parameter c controls the desired filter radius. This initial formulation biases the free-form design towards satisfying a minimum feature size. To bias designs towards one of the two material choices a power penalization is used, such as in the Solid-Isotropic Microstructure with Penalization (SIMP) method,

$$\rho = \rho_{\min} + (1 - \rho_{\min})\tilde{\rho}^{\gamma}, \tag{4.4}$$

where the density of a finite element, ρ, is a function of the filtered density, $\tilde{\rho}$, raised to a power γ. To prevent a singularity in the analysis, this function is scaled such that the density lies between a minimum density, $\rho_{\min}$, and 1. The thermal conductivity, κ of a finite elements is simply scaled by the element density,

$$\kappa = \kappa_0 \rho. \tag{4.5}$$

Since our application is dominated by convective heat transfer, this physics needs to be captured in the analysis. Performing topology optimization while solving the conjugate heat transfer problem is known to be computationally expensive. To reduce the computational expense of the optimization, a static approach, described in [88], is adopted. Consider the following governing equations for conductive heat transfer:

$$\nabla \cdot (\kappa(\mathbf{d})\nabla T) + h(\mathbf{d})A_e(T - T_{amb}) = Q \text{ on } \Omega, \tag{4.6}$$

$$(\kappa(\mathbf{d})\nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N, \tag{4.7}$$

where the conductivity, κ, and the convective heat transfer coefficient, h, are functions of the design variable on the design domain, Ω. The device power loss is input as heat generation, Q, and the boundaries of the domain are adiabatic and restrict heat flux Γ_N. Convection is input into the domain based on the area of a finite element, A_e, and an ambient temperature, T_{amb}. To perform a simplified convection topology optimization, the design variable represents three types of material: (1) thermally conductive, (0) air, and ($[\rho_{\text{low}}, \rho_{\text{upp}}]$) internal boundary. Internal boundaries are identified as material with partial density and are prescribed a unit heat flux of the following form:

$$h(\rho(d)) = \begin{cases} 0 & \text{if } \rho < \rho_{\text{low}} - \delta \\ \frac{h_0}{2\delta}(\rho - (\rho_{\text{low}} - \delta)) & \text{if } \rho_{\text{low}} + \delta < \rho < \rho_{\text{low}} + \delta \\ h_0 & \text{if } \rho_{\text{low}} + \delta < \rho < \rho_{\text{upp}} - \delta \\ \frac{h_0}{2\delta}((\rho_{\text{upp}} + \delta) - \rho) & \text{if } \rho_{\text{upp}} - \delta < \rho < \rho_{\text{upp}} + \delta \\ 0 & \text{if } \rho > \rho_{\text{upp}} + \delta. \end{cases} \tag{4.8}$$

These intermediate densities are present on all boundaries naturally when using the SIMP method with a linear filter, and thus can be used to assign the heat transfer coefficient. To solve the analysis problem, Galerkin finite element analysis is performed with linear shape functions. The dependency of the design variables in the context of the load vector, $\mathbf{P}$, temperature vector,

U, and thermal stiffness matrix, **K**, are presented in the following equation:

$$\mathbf{K}\mathbf{U} = \mathbf{P} \quad \rightarrow \quad \left[\mathbf{K}_{\text{cond}}(\mathbf{d}) + \mathbf{K}_{\text{conv}}(\mathbf{d})\right]\mathbf{U} = \mathbf{P}(\mathbf{d}), \tag{4.9}$$

where the design variables influence both the conductive and convective contributions to the thermal stiffness matrix, as well as the loading vector. The form of these effects are visualized in terms of the element density and prescribed heat flux are presented in Fig. 4.5.

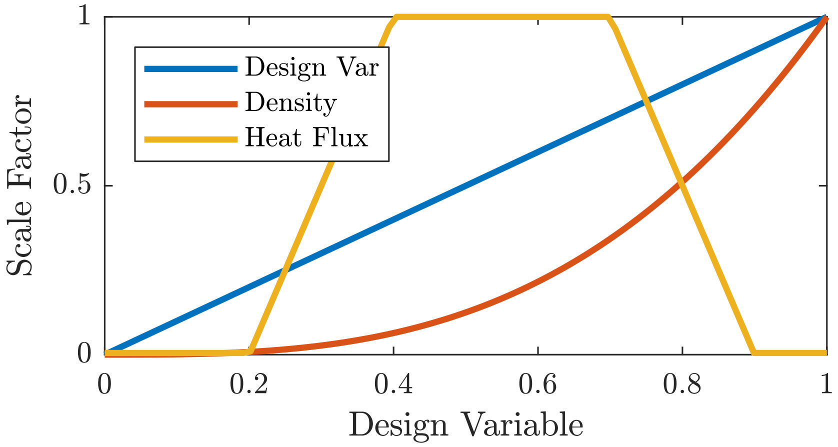


Figure 4.5: Design-dependent convective penalty.

In this study, boundary elements are defined by elements with a filtered density value between $\rho_{\text{low}} = 0.3$ and $\rho_{\text{upp}} = 0.7$ with $\delta = 0.05$ to form a ramped hat function. The density is scaled by the power law as described previously to bias elements towards a 0–1 distribution using a penalty parameter of $\gamma = 2$.

The optimization problem itself can now be represented in standard form:

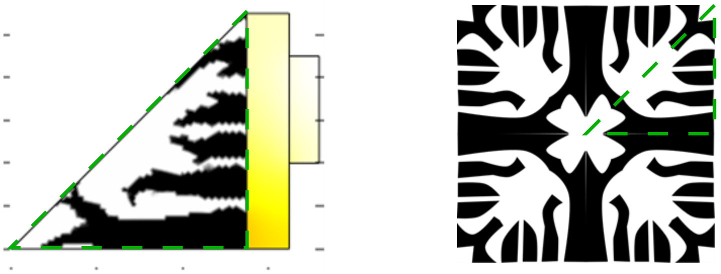
$$\begin{aligned}
\min_{\mathbf{d}}. \quad & \Theta(\mathbf{d}) = \int_{\Omega} qT dA && \text{(4.10a)}\\
\text{s. t.} \quad & V(\mathbf{d}) \leq V_{\text{max}} && \text{(4.10b)}\\
& R(\mathbf{d}) \geq R_{\text{min}} && \text{(4.10c)}\\
& 0 \leq \mathbf{d} \leq 1, && \text{(4.10d)}
\end{aligned}$$

where some objective, $\Theta(\mathbf{d})$, is minimized with respect to a volume constraint, where the volume, $V(\cdot)$, is less than a prescribed value, V_{max}, and a minimum radius constraint where the radius, R, must be greater than a prescribed value, $R_{\text{min}}(\cdot)$. The standard compliance objective is minimized and a length–scale constraint of 0.5 mm radius is enforced. To make a fair comparison between a baseline and optimized heat sink, the cross-sectional area is restricted to 50% of the design domain. The optimization itself is performed in COMSOL multiphysics using the eighth symmetry model on a triangular mesh.

4.4 Topology Optimization Results

The topology optimization formulation produced the heat sink presented in Fig. 4.6. The eighth symmetry model resulted in a structure with 5 sharp fins. Sharp fins are a common feature when using this convection formulation [88], and are a positive attribute for fin efficiency. When using this static-convection model, Eqn. 4.8, the filter fails to enforce minimum length scale. When using an additive manufacturing approach, the sharp fins may not be realizable and the eighth symmetry model is manually post-processed by hand using a vector graphics software[1]. Analyzing the full cross-section of the heat sink, several

[1] Inkscape: inkscape.org



Eighth Symmetry Model

Patterned Design

Figure 4.6: Design-dependent convective penalty results.

aspects stand out. Notably, the use of small fins along the boundaries near the heat source is apparent. In addition to these fins, there is a ring-like structure in the center of the heat sink.

The post-processed design as presented in Fig. 4.6 is not quite manufacturable. A minimum air-gap needs to be enforced in addition to the minimum material thickness constraint to make the design manufacturable by the combination of additive manufacturing and a wax casting process. This may be enforced numerically using a multi-phase approach [113], however, is left as a topic of future work. To satisfy this manufacturing constraint, the design is further post processed to account for the air gap requirement for 3D printing, the resultant design is illustrated in Fig. 4.7

The main features of the design were retained when adjusting the model. Namely, the central ring and small fin structures were included in the design. The fins were also slightly increased in size to 1 mm to improve manufacturing success. In addition to this modified optimal design, a baseline heat sink is created to benchmark the performance of the modified heat sink. This baseline design presents a straight fin heat sink for a single switching cell pair.

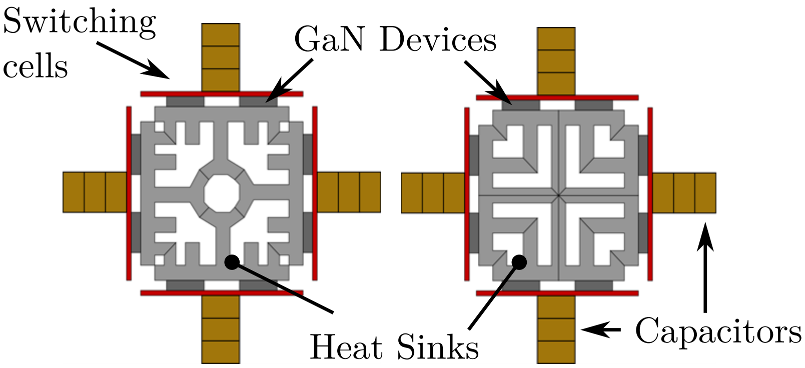


Figure 4.7: Manufacturable heat sink designs. Modified design (left). Baseline design (right).

4.4.1 Performance Prediction

To predict the performance of the heat sink designs more accurately, conjugate heat transfer simulations were performed in COMSOL Multiphysics. A quarter symmetry model of each heat sink is created to analyze the performance of the folded PCB configuration for a single switching cell unit, illustrated in Fig. 4.8.

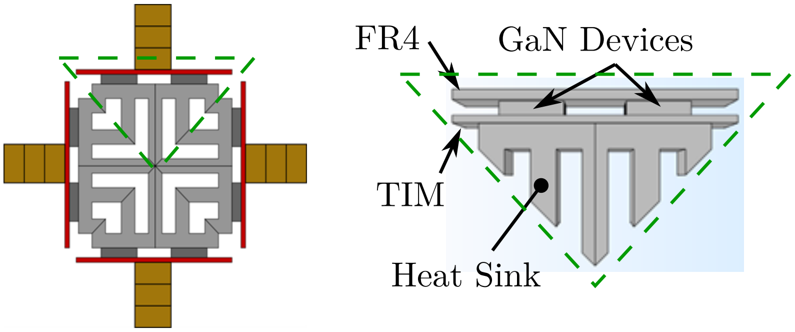


Figure 4.8: Quarter-symmetry COMSOL model for the baseline design.

From the figure, the straight-fin design of the baseline heat sink becomes apparent. Symmetry boundaries conditions were enforced on the internal boundaries of the heat sink and air volume, while heat flux is restricted on the exterior edge of the PCB. Materials were assigned to the model as prescribed and a heat generation of 1W is assigned to each device for a total of 4W of loss on a switching cell unit. The goal of these simulations is to measure the pressure drop and temperature performance of each heat sink design. As such, a coupled thermo-fluid simulation is performed solving both the Navier-Stokes equation and the Laplace equation. The Navier-Stokes equation follows:

$$\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot\left[-P\mathbf{I} + \mu_d\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) - \frac{2}{3}\mu_d(\nabla\cdot\mathbf{u})\mathbf{I}\right] \tag{4.11}$$

$$\nabla\cdot(\rho\mathbf{u}) = 0, \tag{4.12}$$

where momentum is conserved and continuity is enforced as a function of density, ρ, pressure, P, velocity, u, and dynamic viscosity, μ_d. A sample solution as obtained from COMSOL is presented in Fig. 4.9.

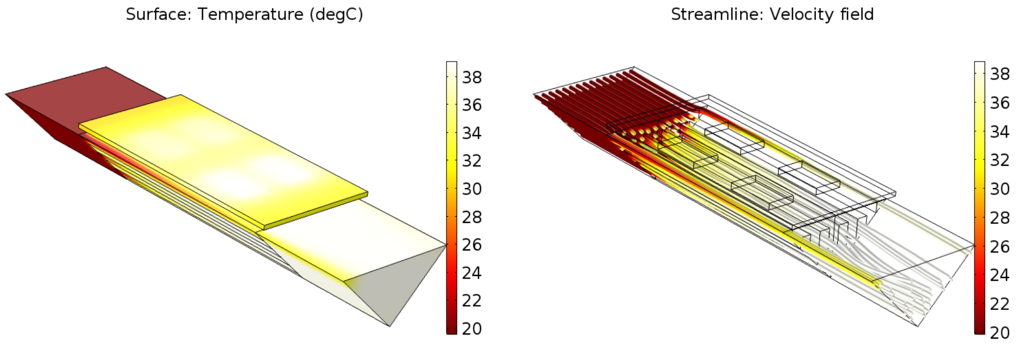


Figure 4.9: Representative quarter-symmetry COMSOL solution.

The fluid inlet is fixed to an ambient temperature of 20°C. As expected, peak hot spots are observed at the heat generating devices. The flow rate of the fan is systematically varied in a parametric sweep to fully characterize both of the heat sink designs (Fig. 4.10).

The heat sink designs possess nearly identical thermal resistance across the

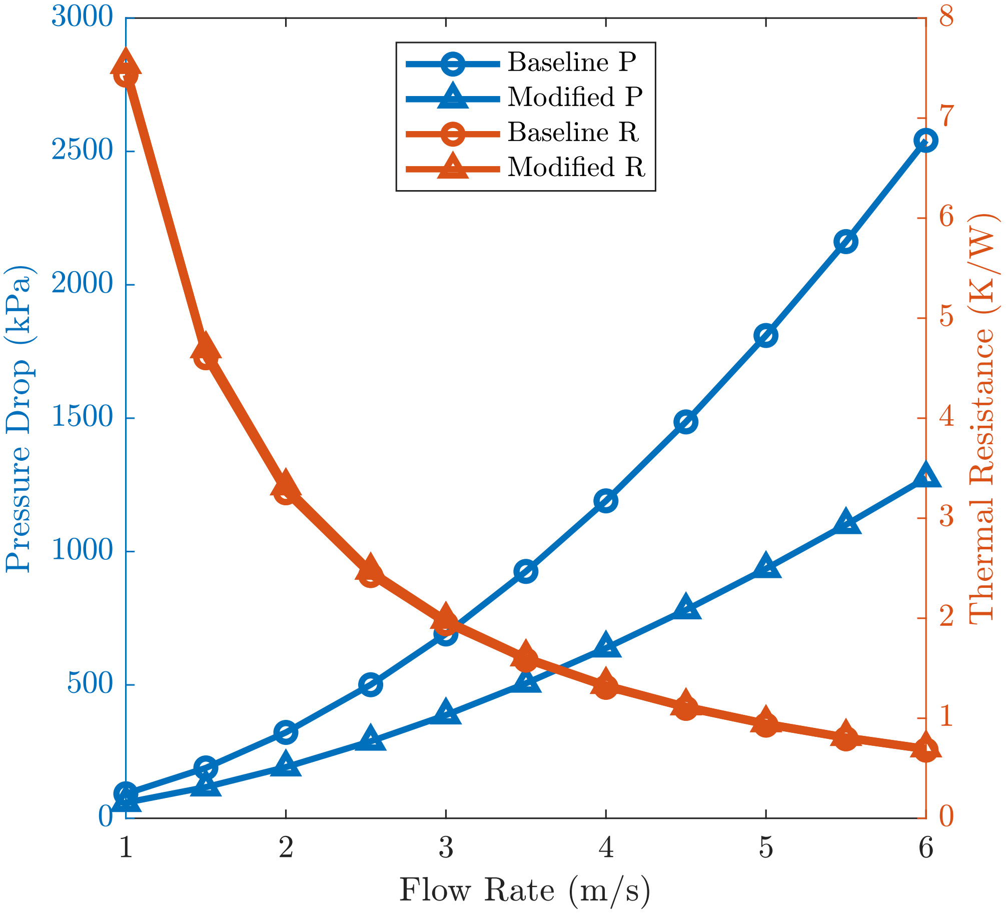


Figure 4.10: Conjugate heat transfer parametric study results.

sweep, and as such provide nearly identical maximum temperatures. The modified design presents a nearly 50% reduction in pressure drop compared to the baseline design. This is likely due to the increase in size of the airflow passages when re-orienting the conductive material. A pressure drop improvement was not explicitly designed for, however, presents a compelling use case for this heat spreading structure.

4.5 Experimental Results

To validate the performance of the heat sink designs, an experiment is performed to measure the temperature of the GaN devices in operation. The heat sinks were fabricated using a lost wax casting process out of silver using the online AM vendor Shapeways[2]; the physical heat sinks are shown in Fig. 4.11.

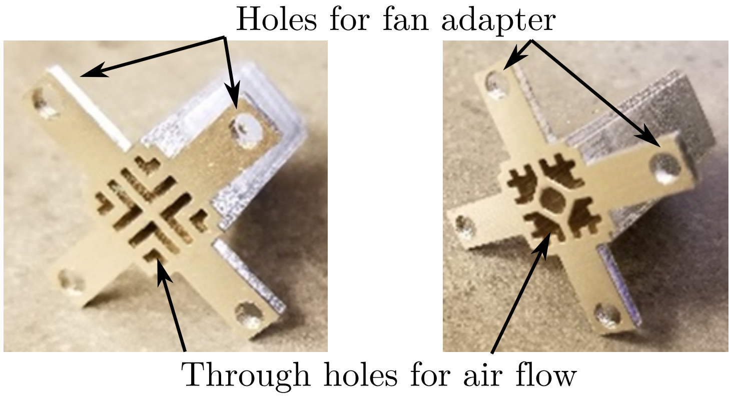


Figure 4.11: Manufactured heat sink designs. Baseline (left), Modified (right).

As mentioned earlier, the minimum feature size is increased to 1 mm to improve manufacturability. The heat sink designs were modified to interface with a custom-designed fan that ensures all of the fan airflow is directed through the heat sink.

4.5.1 Experimental Setup

To measure the temperature of the GaN devices and compare the performance of the heat sink designs, a customized testbench is made. An image of the testbench is presented in Fig. 4.12.

[2] www.shapeways.com

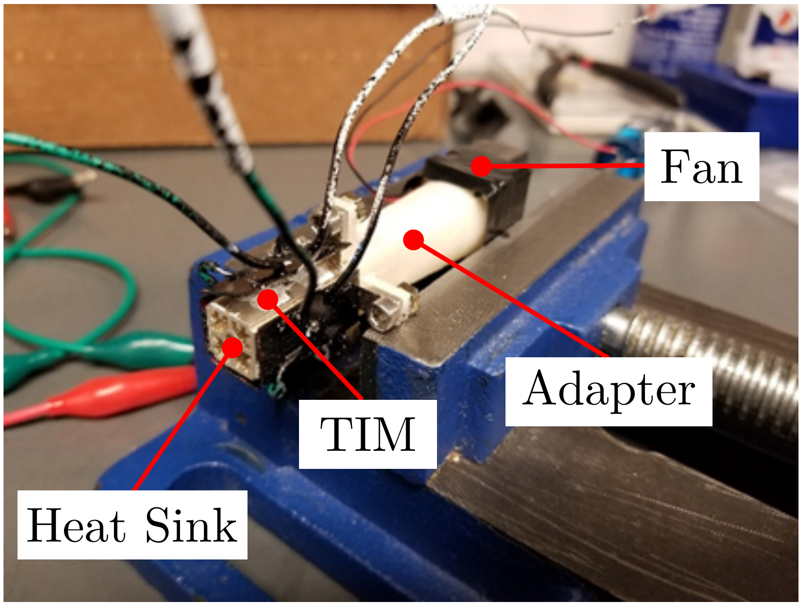


Figure 4.12: Experimental setup.

All of the following experiments share the same electronics board to minimize variations in the experimental procedure. Temperature is measured using FLIR T420 infrared camera pointed at the PCB, on the opposite side to which the device is mounted. The PCB is coated with Krylon 1618, which has a known emissivity to improve the temperature readout of the thermal camera.

The power electronics board is fixed to the heat sink using a single 1 mm screw in the center of each switching cell. The screw is fastened hand tight as required for each thermal interface material (TIM) that is tested. The heat sink is attached to the adapter using 3 mm screws. The fan is mounted to the adapter using 2 mm screws and electric tape is used to make the fan connection to the adapter air tight. A Sunon 5 V fan is used to produce a constant volumetric flow rate of 2 CFM during experimentation.

4.5.2 Thermal Interface Material Selection

An initial experiment is conducted to measure the performance of several thermal interface material options for the folded PCB configuration. As all of the devices are interfaced with the same heat sink; only electrically-isolating TIMs are under investigation. The TIMs considered include a Bergquist Gap Pad, $\kappa \approx 2W/mK$, Alphacool Eisschicht pad, $\kappa \approx 17W/mK$, and Arctic Alumina paste, $\kappa \approx 4W/mK$. The TIMs were tested using a fixed volumetric flow rate of 2 CFM and power is varied incrementally to measure the maximum temperature of the devices. Results using 1 mm thickness TIMs are illustrated in Fig. 4.13.

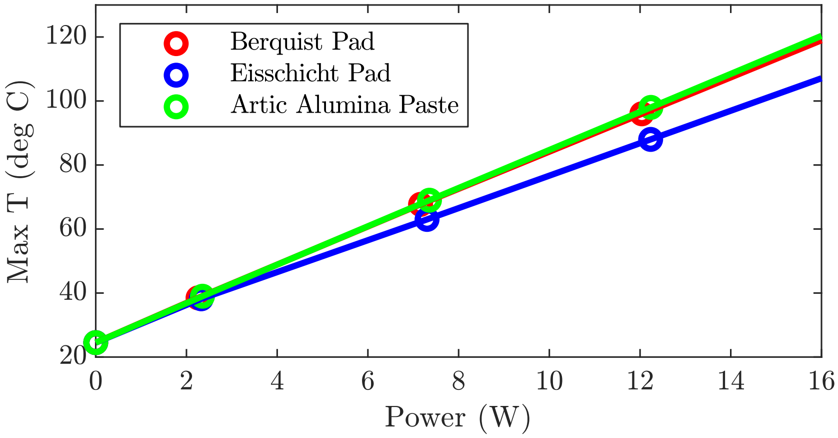


Figure 4.13: TIM usage results.

The Eisschicht Pad resulted in the lowest-temperature operation for the folded heat sink design. It provided a 10 °C decrease in temperature at ≈12W power when compared to the other two TIMs. As such, it will be used to interface the heat generation devices to the heat sinks in the following experiment.

4.5.3 Heat Sink Performance Study

This experiment is conducted to confirm the temperature performance of the two heat sink designs, Fig. 4.11. The power electronic devices were mounted onto the heat sink using the Eisschicht pad. The fan is held at a fixed flow rate of 2 CFM while device power is incrementally varied to generate the data shown in Fig. 4.14.

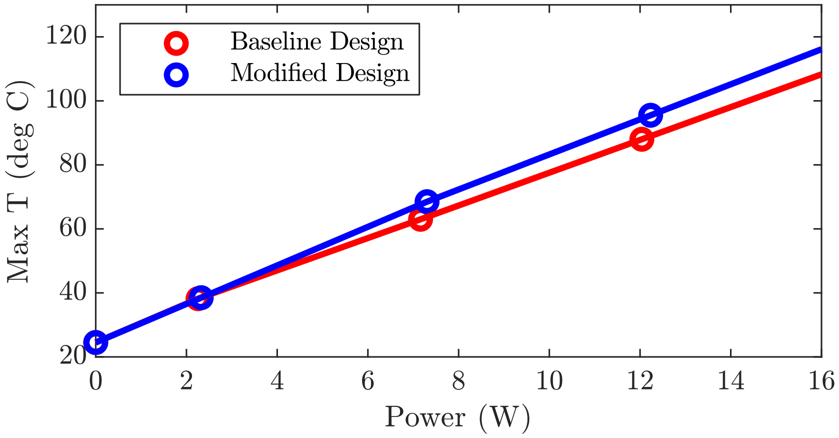


Figure 4.14: Heat sink performance comparison.

Contrary to expectation, the heat sink designs did not provide comparable performance. The baseline heat sink design had a reduced maximum temperature of 7°C at ≈12W when compared to the modified heat sink design. Though the modified heat sink did not perform comparably to the baseline heat sink in terms of temperature minimization, the modified heat sink still presents an interesting opportunity at a system-level with a greatly reduced pressure drop. Validating the pressure drop of the heat sink is left as a topic of future study.

4.6 Discussion

The results obtained from these experiments raise several questions for folded electronics cooling. A discussion of experimental and numerical optimization limitations are presented herein.

4.6.1 Experimental Limitations

In electronics cooling applications, thermal interface materials play a large role in the performance of the overall cooling solution, particularly when the TIM needs to be electrically isolating. In practice, paste TIM materials usually outperform pads due to a large reduction in the thickness of the TIM. In our experiments, we observed that the Arctic Alumina paste performed poorly. We can attribute this performance to several factors related to our experiment design. As shown in Fig. 4.15, the experiment setup involves an interface where local deformation caused by mounting processes can have a significant effect on heatsink contact. The GaN devices used in this study are particularly sensitive to mechanical stress. As such, it is important to securely fasten the heat sink without stressing the devices. Due to the thin build of the switching cell, this becomes a particular challenge. Fastening the switching cell with a single screw caused the PCB to bend slightly when fastened. For the pad

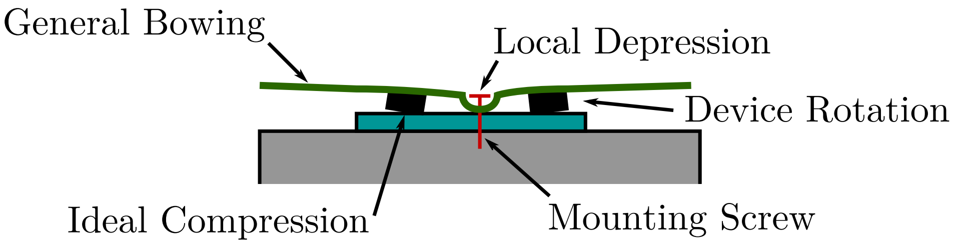


Figure 4.15: Mechanical mounting concerns.

materials, this did not cause a large issue due to their thickness. For the paste, this presented a challenge where the device was slightly rotated and did not make uniform contact with the heat sink. We believe this to be a flaw in the experiment design and will be addressed for future studies.

4.6.2 Topology Optimization Limitations

The experimental performance of the modified heat sink design also raises questions relating to the optimization procedure used. The emulation of convection through the use of a design-dependent outward heat flux is an intuitive way to parameterize the problem for gradient-based optimization. This formulation is very similar to a perimeter maximization problem subject to a prescribed volume constraint, however, there is a key issue. Consider designing a heat spreading structure by analyzing the convective heat flux:

$$q_{\text{in}} = hA_d(T_d - T_{\text{amb}}). \tag{4.13}$$

Assume the input heat flux is managed completely by convection, and the convective heat transfer coefficient and environment temperature are fixed. For an electronic device the temperature, T_d, may be fixed based on operating limits, and the device area, A_d, is designed to enforce energy balance. In the formulation presented by Ref. [88], T_d is minimized as the mean temperature and the area is effectively scaled through use of the hat function. Given that these two parameters are not completely aligned, the optimizer converges to a local optima with a high temperature. Energy is conserved, however, the domain temperature is not well reduced. This effect may be alleviated when explicitly optimizing for the maximum temperature, or by constraining the maximum temperature on the domain.

This reduced-order analysis model does not account for fluid interaction, which plays a large role at the scale of this work. There are several lower-order models which can be used to account for this coupling [91, 114], however, they come at a much greater computational expense than solving the conduction-only problem. Capturing simplified fluid interaction without solving the Navier-Stokes equations in a topology optimization would provide a great computational design tool and is left as a topic of future work.

4.6.3 Summary

In this chapter a novel "folded" packaging configuration was proposed for the FCML circuit topology. This packaging configuration resulted in a 40% reduction in cooling system volume due to the centralization of heat-generating devices and the use of an efficient axial fan (instead of a conventional centrifugal blower fan). The heat sink for the folded PCB configuration was topologically optimized using a simplified convection model. The optimized heat sink did not satisfy all manufacturing constraints and was used to influence an interpretation of the heat sink into a modified design. When using high-fidelity simulation, the modified heat sink design offers comparable temperature performance with a 50% reduction in pressure drop when compared to a baseline design. Both heat sink designs were manufactured and experiments were conducted to identify the best TIM, and to assess the temperature performance of the heat sink designs. The Eisschicht pad was found to perform the best, and the modified heat sink design resulted in a higher maximum temperatures when compared to the baseline design. This demonstrates the lack of accuracy in conjugate heat transfer simulation and motivates the use of experimentation to validate heat sink performance.

Chapter 5

Topology optimization for chip-scale heat spreaders using reduced-order convection models

5.1 Introduction

Advances in high-band gap devices have enabled the design of high temperature power electronic systems. These systems present new challenges for thermal engineers where devices with both low- and high-temperature sensitivity are placed within a common package. This heterogeneous mix of electronic devices motivates heat removal at a device level to prevent the unintentional failure of nearby devices.

Researchers have developed several methods to address the challenge of heat removal using a heat spreading structure for using fixed-mesh strategies. These include voxel representations [81], explicit geometric representations [115], and generative representations [116]. These have been coupled to a variety of reduced-order convection models to reduce the computational expense of heat spreader optimization [87–89]. A general review of topology optimization methods for heat transfer systems is presented in [93].

Yet there is very limited experimental validation for topology optimization methods for heat spreader design. When using conduction models, [117] verified the performance of topologically optimized heat trees in 2D. As another example, [94] used 3D conduction topology optimization to design a thermal bridge for an automotive power control unit. Thermo-fluid physics have also been used to optimize heat spreaders for gaseous and liquid fluid flow models using reduced order 2D [105, 118] and full 3D [119] designs. There have not been many examples of optimization using the simplified convection models that are still being published today. One example is the work of Dede et al. [90], where 3D topology optimization was performed with a simplified convection model to design a heat spreader for a jet impingement flow. The authors observed comparable performance when compared to commercially available heat sink designs.

In this chapter, topology optimization using reduced order methods is investigated for a power electronics application. First the test sample for experimental validation is motivated in Sec. 5.2. Then an initial set of experiments is conducted to select a heat sink material in Sec. 5.3. A variety of heat spreader design problems and solved to produce a selection of optimized structures in Sec. 5.5. Several of the optimized designs are fabricated and their performance is experimentally quantified in Sec. 5.5. A discussion of findings is presented in Sec. 5.7 and a summary of the chapter is presented in Sec. 5.8

5.2 Device Level Heat Sink Design

The increased power density of wide-bandgap devices motivates the use of localized cooling to improve system-level efficiency. Consider the following diagram of a bottom-cooled device utilizing thermal vias to extract heat through the PCB with an individualized heat spreading structure, shown in Fig. 5.1. A wide-band-gap device, such as a Silcon-Carbide MOSFET, is mounted on a copper pad using solder. A series of copper coated vias are used to transfer electrical power between the top and bottom surfaces of the PCB. These vias also act as a thermal pathway to improve heat transfer through the PCB. A

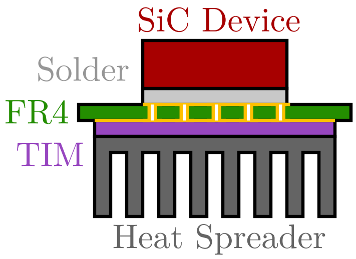


Figure 5.1: Bottom-cooled chip-scale heat sink.

heat spreading structure may be attached to the bottom side of the PCB to improve heat transfer to the environment and decrease the maximum temperature of the device on the top side of the PCB. To minimize the thermal resistance of the interface between the copper vias and heat spreading structure, a thermal interface material (TIM) may be used.

This packaging configuration is not yet commonly used and presents a unique set of challenges. The key features of this configuration are summarized in Table 5.1.

Key Features	Advantage	Disadvantage
Individual device heat spreaders	· Electrical isolation · Simple interface	· Increased assembly complexity
Thermal via heat transfer	· High conductivity · Low mechanical stress	· Poor manufacturing tolerances · Reduced PCB area
Optimized Heat Sink	· Higher temperature capacity	· Increased manufacturing complexity

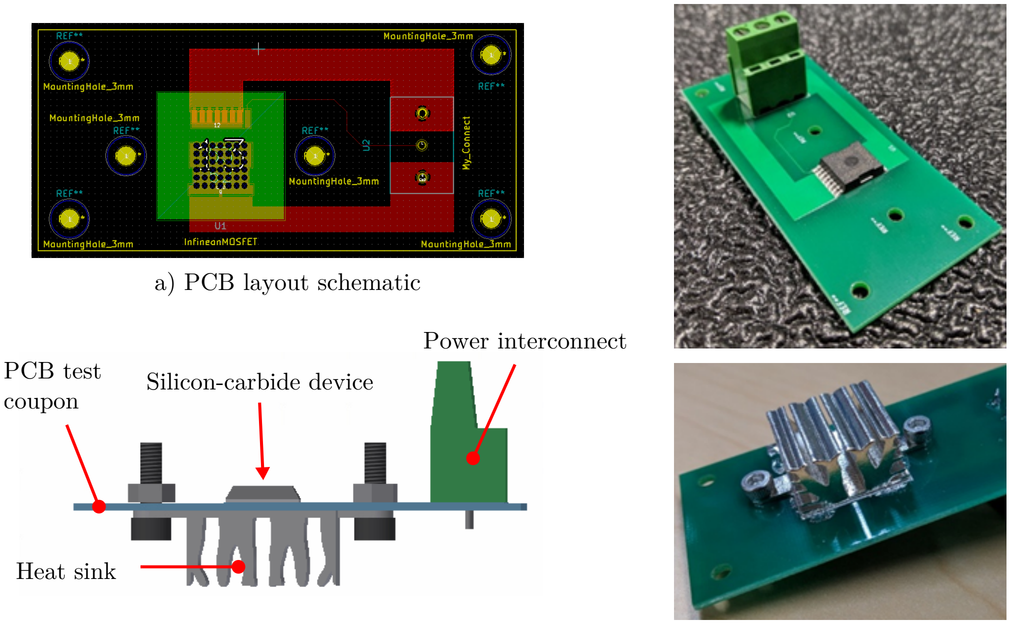
Table 5.1: Key considerations for through-PCB cooling.

The use of individualized heat sinks removes electrical isolation concerns for using a single heat spreader to cool multiple devices. As such, the interface between the heat spreader and PCB is also simplified since there is a single surface elevation. With these advantages comes an increase in assembly complexity, time, and cost. Using a bottom-cooled device and thermal vias for heat transfer may increase reliability by eliminating the mechanical stress that is typically present when mounting a heat spreading structure directly to an electronic device. Mechanical stress is typically introduced to help minimize thermal resistance between the heat spreading structure and device by reducing the distance between the two surfaces. This may lead to issues with the soldered connection point between device and PCB and is a common point of failure. Combining the thermal vias with individualized heat spreaders allows for the use of high-conductivity thermal interface materials (typically electrically conductive) which can fill air gaps in the vias. This greatly increases the effective thermal conductivity through the PCB. The proposed advantages come at a cost of reduced usable PCB surface area.

The focus of this chapter is the optimization of the heat spreading structures for the packing configuration shown in Fig. 5.1. This configuration presents a potential increase in the power capacity for the circuit, which will come at an expense of manufacturing complexity. Additive manufacturing processes will be used to alleviate concerns related to manufacturing complexity.

5.3 Initial Heat Spreader Experiments

To realize the through-PCB cooled concept, a custom 76x35 mm PCB was designed and fabricated, shown in Fig. 5.2. A single SiC device, Infineon IAUT300N10S5N015ATMA1, is soldered to the top of surface of the PCB. Power is routed to the device through an interconnect on the top surface of the PCB. Electrical power enters and exits the PCB through the top surface of the PCB, represented as the red wiring path in Fig. 5.2a. A total of 255 thermal vias with a hole diameter of 0.15 mm are placed on the copper mounting pad as per manufacturer specification for the PG-HSOF-8-1 footprint. The role of



a) PCB layout schematic

b) PCB CAD rendering

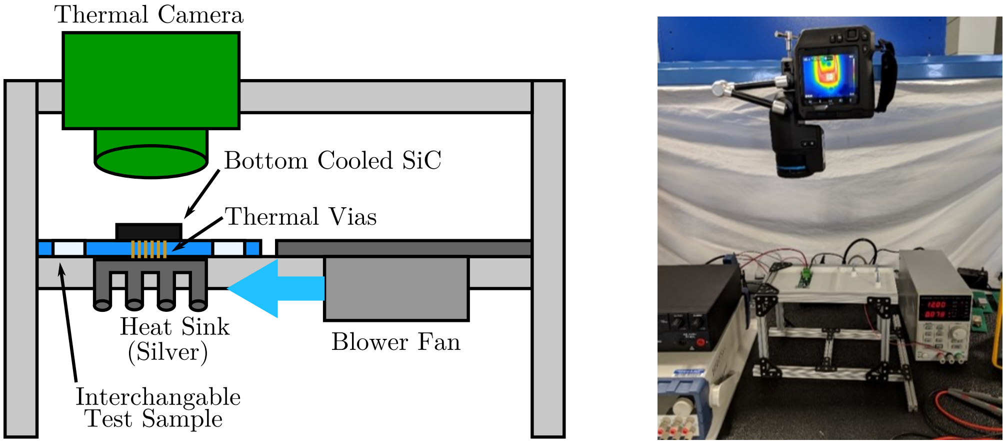
b) Manufactured test sample

Figure 5.2: PCB Test Sample.

the copper vias for this circuit is solely for heat transfer, as there is no electrical connection between the top and bottom surfaces of the PCB. A heat sink is mounted on the reverse side of the PCB and is secured with 3mm screws, as shown in Fig. 5.2b. The fabricated PCB is presented in Fig. 5.2c.

To test the performance of the heat sink design on this test sample, an experimental apparatus is created, Fig. 5.3. The PCB test board is placed onto a structure with the SiC device facing up. A FLIR T450 thermal camera is fixed above the device to measure spot temperature at the center of the device. A Sunon 12V blower fan, Sunon MF60151V1-B00U-A99, is used to force air through the heat spreading structure on the bottom side of the PCB. An image of the experimental setup is presented in Fig. 5.3b.

With the intent of testing unconventional heat sink designs, additive man-



a) Diagram of experimental setup

b) Experimental setup

Figure 5.3: Experimental setup.

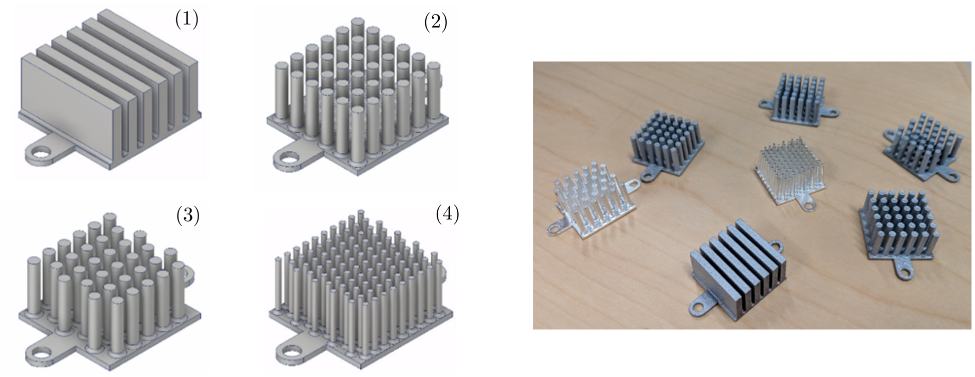
ufacturing (AM) is investigated as a manufacturing process. The online manufacturing service, ShapeWays.com, offers two candidate materials, Table 5.2. The first material, silver ($\kappa \approx 400W/mK$), is fabricated through a lost wax casting process. This process restricts the minimum feature size to 1 mm and produces a smooth surface texture. The second material is an aluminum alloy ($\kappa \approx 130W/mK$) formed using a selective laser melting (SLM) process. This restricts the minimum feature size to 1.5 mm and results in a rough surface texture. The silver process has a minimum lead time of 8 days and the aluminum process has a minimum lead time of 15 days. Prior to numerical optimization, a heat spreading structure material must be selected for model

Material	Silver	Aluminum
Process	Lost wax casting	Selective laser melting
Thermal conductivity	400 W/mK	130 W/mK
Minimum feature	1mm	1.5mm

Table 5.2: Candidate materials from ShapeWays.com

calibration.

A series of experiments are conducted to select a material for future studies using the following benchmark heat spreading structures, shown in Fig. 5.4. These include a straight fin design (a1), a pin fin design (a2), and a staggered pin fin design (a3, a4) as shown in the figure. The heat sink designs are printed in both materials at their rated minimum feature size. Several silver heat sinks are fabricated at 1.5 mm to directly compare performance with the aluminum designs.



a) Benchmark heat spreader designs

b) Fabricated heat sinks

Figure 5.4: Benchmark heat spreading structures for material selection.

5.3.1 Experimental Procedure

The blower fan is set to a fixed voltage, 12V, to provide a constant rated airflow of 5.7 CFM during experimentation. The SiC device used as a diode to generate heat. The input current to the SiC MOSFET is set to a fixed point and the system is allowed time to reach steady state. Once the system reaches steady state, the spot temperature at the center of the MOSFET is recorded using the FLIR camera. The current passing through the device, I_{in}, is recorded by calculating the voltage drop across a shunt resistor, and the voltage drop across the device, V_{device}, is measured as the average voltage between the source pins and drain bar. The voltage measurements are taken using a FLUKE multimeter. The voltage drop and current measurements can be used to obtain the power loss of the device:

$$P_{\text{loss}} = I_{\text{in}} V_{\text{device}}. \tag{5.1}$$

The input current is then increased at uniform intervals to produce a sequence of power loss and temperature measurements. The experiments are halted when the device reaches a maximum spot temperature greater than 100 $°C$.

5.3.2 Experimental Results

First, an experiment was conducted to measure the performance variations between the two manufacturing processes. Namely, to determine whether the higher thermal conductivity of silver overcomes the improved surface texture of the aluminum from the melting process. Staggered pin heat sinks of a 1.5 mm minimum feature size are used for this experiment, and results are presented for a step input power loss of 10 W in Fig.5.5.

At a minimum feature size of 1.5 mm, the aluminum heat spreader outperforms the silver by a margin of 6°C (17%). This suggests the increased surface roughness of the aluminum heat sinks is more influential than the increased thermal conductivity of the silver heat sink. However, testing the silver staggered pin fin design with a 1 mm minimum feature size reveals significant improvement of 6°C when compared to the aluminum design at 1.5 mm for a total improvement of 26%. Aluminum is a more practical material for engineering systems

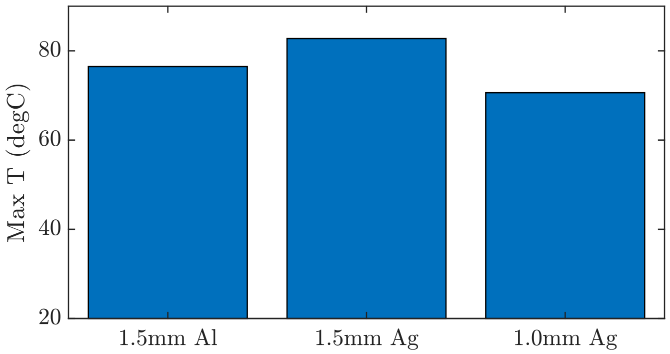


Figure 5.5: Results from material selection experiment.

due to its low cost. However, SLM manufactured aluminum is quite brittle, and several heat sink fixtures were destroyed when mounted to the PCB. As such, the silver material is selected for future experiments. The results in this thesis using silver heat sinks may be scaled to predict performance when using an the SLM manufactured aluminum.

In the process of testing different heat sink materials, several thermal interface materials were tested, as shown in Table 5.3.

Material	Thermal Conductivity	Type	Electrically Isolating?
Bergquist Gap Pad	≈ 2 W/mK	Pad	Yes
Alpha Cool Eisschicht	≈ 14 W/mK	Pad	Yes
Arctic Silver	≈ 8.9 W/mK	Paste	Yes
Solder	≈ 60 W/mK	Paste	No
Liquid Pro	≈ 82 W/mK	Liquid	No

Table 5.3: Candidate Thermal Interface Materials.

When using the liquid or paste materials, the heat sinks can be securely tightened to the PCB to minimize the thickness of the TIM and hence thermal resistance. This resulted in some TIM entering the voids in the vias and excess TIM being squeezed out from under the heat sink. Testing the same heat sink on multiple PCBs revealed large variations in performance (greater than 20°C, or 20%). This reveals poor manufacturing tolerances for the vias where they may not be properly coated with copper and/or the vias are filled with FR4. As such, the remaining experiments in this dissertation are conducted using the same PCB to ensure a fair performance comparison. Some experimental results, at 10W of power loss, using the different TIM materials are presented in the Fig. 5.6. The largest performance improvement was observed when changing from single to double-digit thermal conductivity. Using a TIM with larger thermal conductivity has diminishing effect of reducing the maximum temperature of the device. Furthermore, the reduced interface thickness when using Arctic Silver overcomes the higher thermal conductivity of the AlphaCool Eisschicht pad. It is important to note that the poor performance of the solder can be attributed to its application (by hand). Using specialized installation machinery, performance should be comparable to Arctic Silver and Liquid Pro. These observations were drawn after testing many of the optimized heat sinks with several TIMs. The most consistent and repeatable data was obtained when using the Arctic Silver TIM and reusing a single PCB for all experiments. As such, experimental results utilizing Arctic Silver are pre-

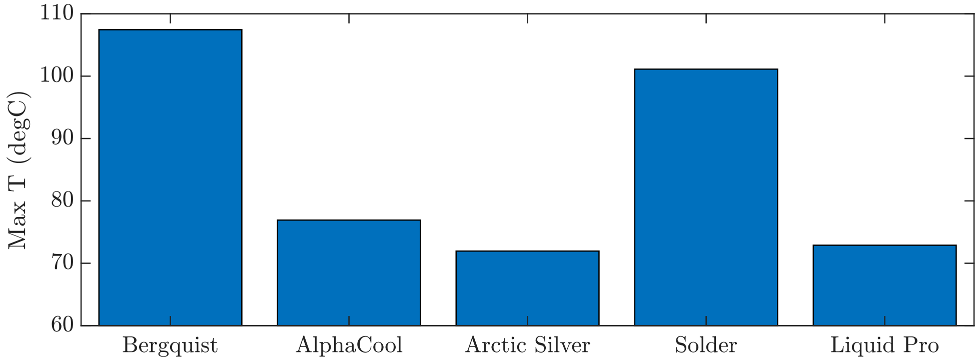


Figure 5.6: Results from testing various TIMs.

sented. The maximum temperature of a device using a different TIM may be predicted by scaling the experimental data according to these findings.

5.4 Design problem formulation

The heat spreader design domain is constrained to the air volume above the heat generating device, shown in Fig. 5.7.

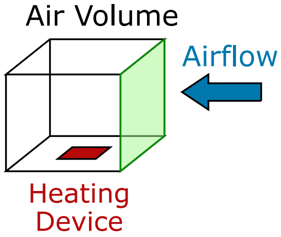


Figure 5.7: Heat spreader design abstraction.

An input airflow, blue arrow, is considered on one of the faces of the air volume box and an input heat flux, shaded red square, is applied to a portion of the bottom surface representing the device. For the subsequent studies, a 2D design domain is considered utilizing the face of the bounding box that is perpendicular to the airflow (the shaded green face). The design problem is summarized in the following problem formulation,

$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) \tag{5.2}$$

$$\text{s.t.} \quad 0 \leq \mathbf{d} \leq 1 \tag{5.3}$$

$$V(\mathbf{d}) \leq V_{\max} \tag{5.4}$$

$$R(\mathbf{d}) \geq R_{\min}, \tag{5.5}$$

where some objective, Θ, is minimized subject to a set of design variables,

d. The volume of the heat spreading structure, $V(\mathbf{d})$, is restricted to occupy a maximum volume, $V_{\text{max}} \leq 50\%$, and the minimum feature size, $R(\mathbf{d})$, is restricted to $R_{\text{min}} \geq 1$ mm based on manufacturing constraints of the silver process. To generate a wide variety of candidate structures the governing physics, boundary conditions, objective function, and design representations will be varied.

5.4.1 Physics models

First, consider the use of several reduced-order physics models to generate optimized topological structures for heat sinking. Four variations of governing physics and boundary conditions are presented in Fig. 5.8.

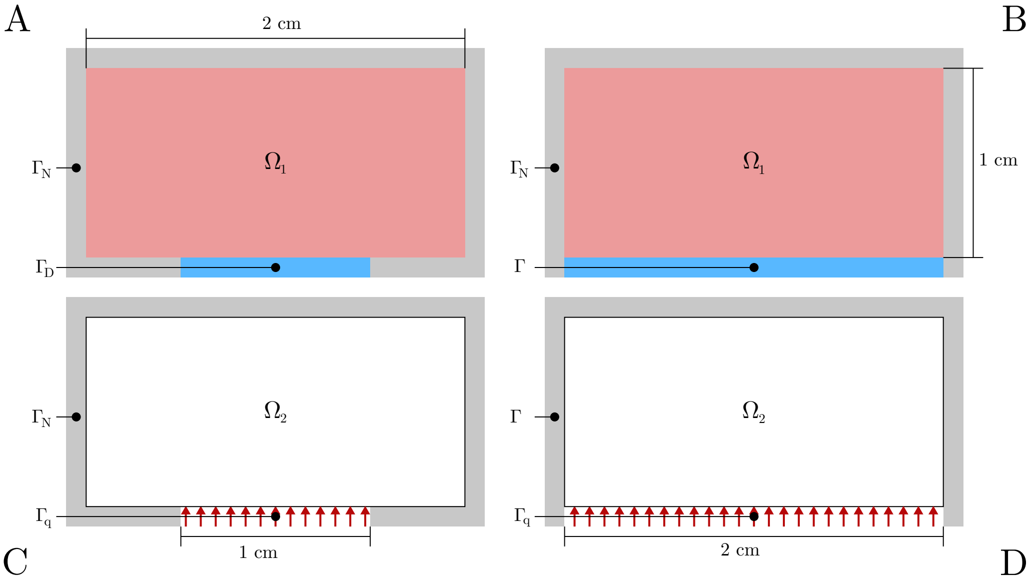


Figure 5.8: Candidate boundary conditions.

Conditions A and B consider thermal conduction governed by the following equations:

$$\nabla \cdot (\kappa \nabla T) + f = 0 \text{ on } \Omega_1 \tag{5.6}$$

$$T = 0 \text{ on } \Gamma_D \tag{5.7}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{5.8}$$

The heat spreader is designed considering a 10 W of loss in the power electronic device. This is loss is input to the finite element analysis as a feat flux term, f. At the interface between the device and heat sink, a fixed temperature condition of 0°C is applied on the boundary, Γ_D. For Case A, the true dimensions of the device are considered. For Case B, the fixed temperature boundary is set to the entire bottom boundary. The remaining boundaries are adiabatic and restrict heat flux, Γ_N. These analysis domains may be used as an abstraction of the design problem where the heat extraction problem is running in reverse. Namely, heat is taken from the domain and funneled to the boundary by designing the thermal conductivity, κ, distribution on the domain, Ω_1.

The second set of boundary conditions, C and D, consider convection and is governed by the following equations,

$$\nabla \cdot (\kappa \nabla T) + hA_e(T - T_a) = 0 \text{ on } \Omega_2 \tag{5.9}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = q_d \text{ on } \Gamma_q \tag{5.10}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{5.11}$$

Here the same 10W of power loss is input to the interface boundary between the heat sink and device, Γ_q, in the form of an input heat flux, q_d. Cases C and D are similarly differentiated as Cases A and B, namely, by the size of the boundary. The remaining boundaries, Γ_N, are adiabatic. For this problem, heat extracted out-of-plane is a function of the convective heat transfer coefficient, h, and ambient temperature, T_a. The implementation of the convection term will vary depending upon the physics simplifications utilized in the two methods:

- A constant convective force applied to the entire domain,
- A constant convective force applied to the interface between solid and void elements.

For both cases, material density scales both κ and h. In the first case, the convection is applied uniformly throughout the entire domain. The second case is an extension of the approach presented in [88], where a smoothed-out hat function is used to apply the convection term at elements with partially-defined material boundaries through the scaling of the parameter h.

Solving topology optimization problems with these reduced-order physics representations results in 3 different types of structure, illustrated in Fig. 5.9.

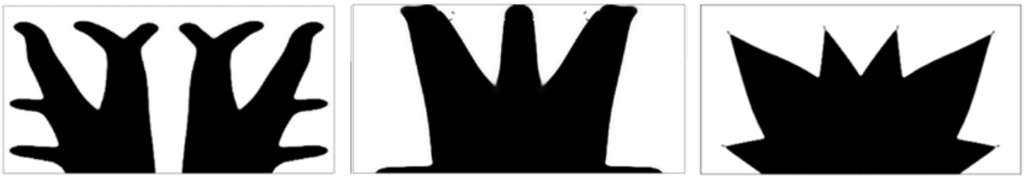


Figure 5.9: SIMP topology optimization solutions for thermal compliance objective. Pure conduction (left), conduction with uniform convection (center), conduction with boundary-dependent convection (right).

The optimized solution for the conduction physics (left topology) presents a dendritic structure as typically seen in the literature. When including the constant convection term, a straight fin structure is preferred (middle topology).

When restricting the convection force to the partially defined material at the solid-void interface, pointed fins are generally developed (right topology).

5.4.2 Objective function

Next, the structural variations that result as a function of changing the objective function are considered. The following three objective functions presented in Table 5.4 are investigated.

$\Theta_C(\mathbf{d}) = \mathbf{U}^T\mathbf{P}$	$\Theta_N(\mathbf{d}) = \left(\mathbf{L}^T\mathbf{U}^p\right)^{1/p}$	$\Theta_V(\mathbf{d}) = \frac{1}{n}\left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n}\right)^T\left(\mathbf{U} - \frac{\mathbf{L}^T\mathbf{U}}{n}\right)$

Table 5.4: Candidate objective function.

This includes the thermal compliance, Θ_C, defined by the product of the temperature vector, $\mathbf{U}$, and thermal load vector, $\mathbf{P}$, which is a common objective function in density based topology optimization. Also, a norm-approximated maximum temperature, Θ_N, as the product of a vector of ones, $\mathbf{L}$, the temperature vector, $\mathbf{U}$, and a norm order, p. Finally, the temperature variance, Θ_V, is also considered as the average of the squared difference of the nodal and average temperature on the domain. The selection of objective function causes large changes in the optimized structure. Consider the following solutions obtained by optimizing a conduction problem using the SIMP method, shown in Fig. 5.10.

When moving from left to right, the optimized structure becomes less bulky and the interface material to the fixed temperature boundary is reduced. Furthermore, the number of secondary branches increases. This may improve performance for convective heat transfer where the amount of heat flux extracted correlates with the surface area of the heat spreader.

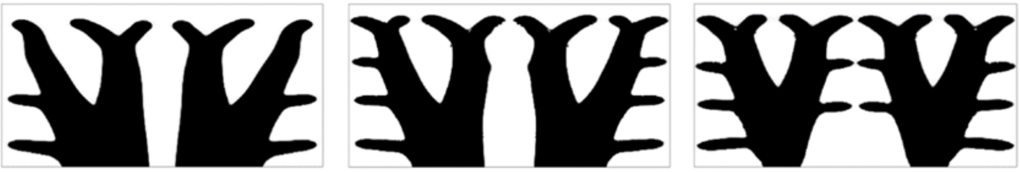


Figure 5.10: Topology optimization solutions for various objective functions. Thermal compliance (left), maximum temperature (center), temperature variance (right).

5.4.3 Design Representations

The use of three different design representations are considered here.

5.4.3.1 SIMP Method (SIMP)

The first method uses a voxel-based design representation. This method is adapted from Ref. [6], and structured for heat transfer applications. The design vector using this method is given by:

$$d = [\alpha_1, \alpha_2, ..., \alpha_n], \tag{5.12}$$

where each finite element (voxel) is assigned a density scalar, α_i. The minimum radius constraint may be enforced implicitly using a density filter [120]. The volume constraint is calculated as the sum of element individual finite element densities, and is enforced explicitly using an optimization constraint. This problem is solved using the Method of Moving Asymptotes (MMA) algorithm.

5.4.3.2 Geometric Projection Method (GPM)

This next method is based on the use of discrete geometric bars to represent the heat sink design. A differentiable geometric projection is used to map the bars onto the finite element mesh, as defined in Ref. [115]. This differentiable projection enables representation of the heat sink by simple geometry without the need to remesh the structure. The smooth projection also naturally accommodates the simplified convective boundary conditions used in this research. The composition of the design vector for this method follows:

$$d = [xi_1, yi_1, xf_2, yf_2, \alpha_1, ..., xi_n, yi_n, xf_n, yf_n, \alpha_n], \tag{5.13}$$

where each bar has a total of 5 design variables to capture the initial bar node, $[xi_1, xi_2]$, final bar node $[xf_1, xf_2]$, and bar density parameter, α. In this study, a varied number of bars are projected onto the domain. Each bar has a minimum thickness to satisfy the minimum radius constraint. The volume constraint is be satisfied by removing and/or overlapping bars during the optimization routine. The optimization using this design representation is also driven by the MMA algorithm.

5.4.3.3 Space Colonization Algorithm (SCA)

This final method used for heat sink design is based on the use of generative algorithm as an indirect representation of the heat sink design. The generative algorithm used here is the Space Colonization Algorithm, which is used to emulate venation. This algorithm has shown promise in previous heat sink design studies [8], and the readers are referred to this document for details on the algorithm implementation. The composition of the design vector follows:

$$d = [n_\mathrm{s}, n_\mathrm{g}, l_\mathrm{g}, ax_1ay_1, ax_2ay_2, \ldots, ax_n, ay_n], \tag{5.14}$$

where the optimizer will decide the number of initial stem nodes, n_s, the number of venation growth stages, n_g, the length of the vein growth per iteration,

l_{g}, and the auxin coordinates $[ax_1, ay_2]$. For this study, the number of stems is varied from 1:10, the number of growth stages is varied from 1 to 3, the vein growth length is between 5–10% of the domain width, and 40 auxins are introduced on the domain. The same differential geometric projection as mentioned for the geometric projection method is applied to project the topology here on the finite element mesh with elements of a fixed width. A nested optimization loop is run for each design candidate to increase the projected bar widths uniformly until the minimum volume constraint is satisfied using a bisection algorithm. The outer loop topological design is driven by a genetic algorithm.

5.5 Optimization Results

Optimization was performed for each of the physics models, boundary conditions, optimization methods, and objective functions to result in several dozen optimized heat spreader designs.

5.5.1 Selected optimized solutions

To demonstrate the differences in structures produced by each design method, a selection of optimized designs produced by each algorithm is presented in the following figures. Consider first some designs produced by the SIMP method, shown in Fig. 5.11. Based on the selection of objective, physics, and boundary conditions, a variety of structures emerge. These include tree-structures, straight fin structures, pointed structures, and a variety of structures across these spectra.

A selection of optimized designs using the GPM are presented in Fig. 5.12.

Depending on the number of bars used as design variables, the GPM produces somewhat simplified versions of the SIMP optimized designs. With a limited number of bars, simple straight-finned patterns emerge. As the number of bars increases, designs related to tree-like patterns emerge.

A selection of designs searched using the SCA are presented in Fig. 5.13. The optimizer here was allowed to choose the number of initial stem nodes and complexity of the resulting structure. The better performing designs featured more initial stem nodes and conductive paths that covered a majority of the design space. Though the algorithm targeted tree-like structure, heuristics were not included to guarantee the manufacturability of the design. As such, many designs features narrow regions of void which prevent manufacturing.

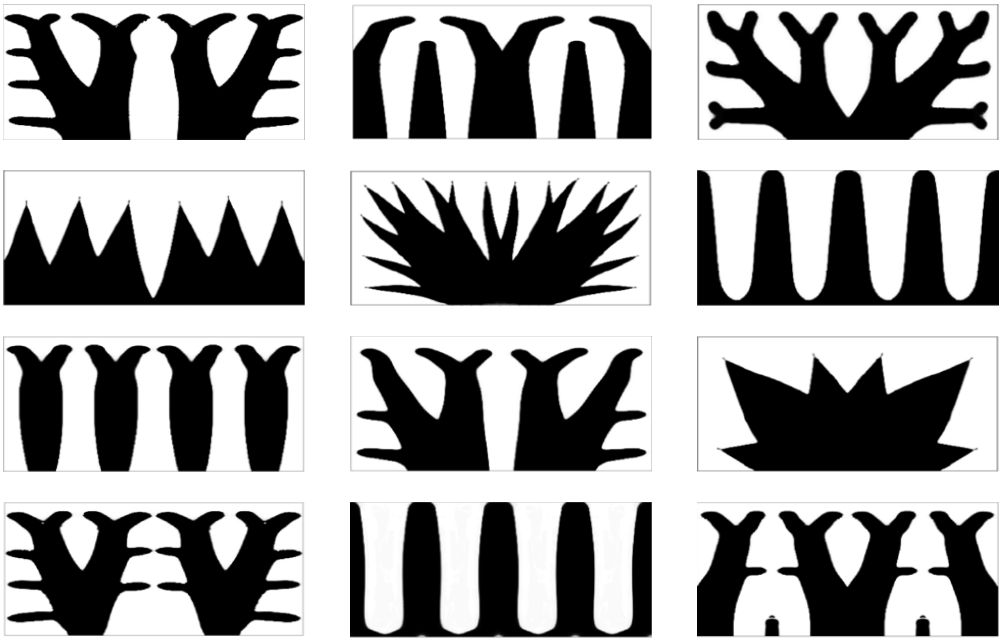


Figure 5.11: Selected designs produced by SIMP method.

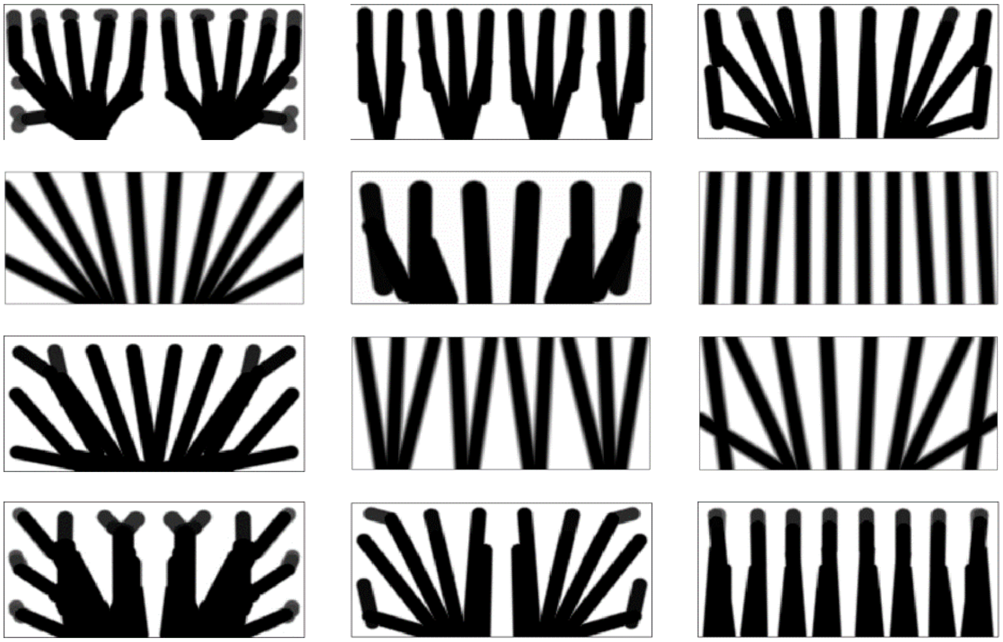


Figure 5.12: Selected designs produced by the GPM.

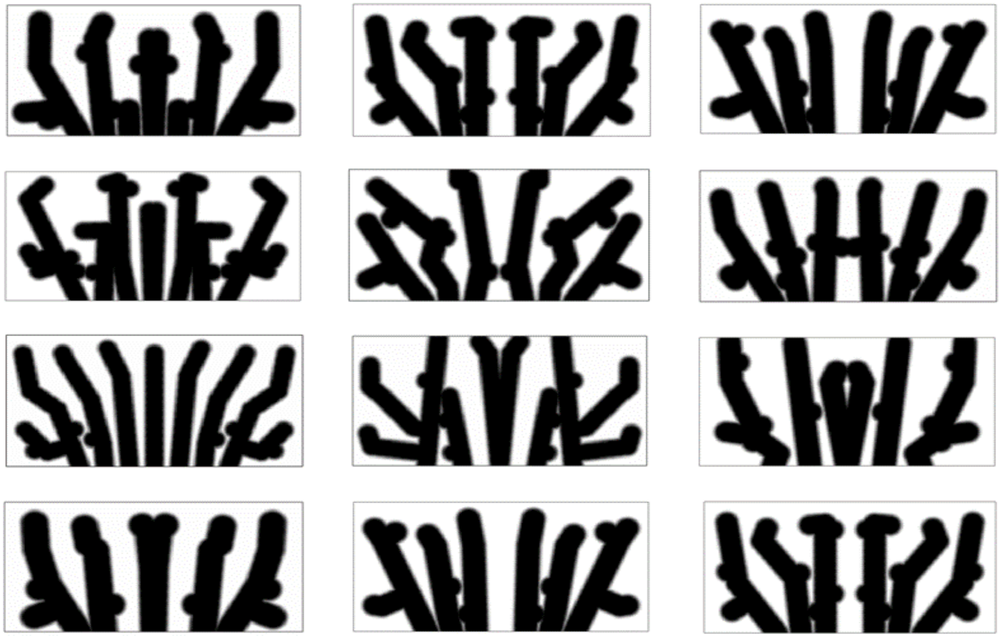


Figure 5.13: Selected designs produced by space colonization algorithm.

5.5.2 Fabricated Heat Spreading Structures

Of the numerous optimized designs, 12 sufficiently unique designs were selected and fabricated using a lost wax casting approach from the online vendor ShapeWays^0 . The heat sink cross-sections and physical realizations are shown in Fig. 5.15.

All of heat spreader structures share a uniform baseplate such that they can be secured to the PCB. The 2D cross-sections of the selected optimized designs were traced in Autodesk Inventor and extruded along the baseplate to form heat sink designs. Within this selection of heat sink designs, a variety of tree and straight-fin structures were chosen for experimental testing. The performance of these optimized designs will be compared to a more conventional heat straight fin sink design that may be found off the shelf, shown in Fig. 5.14. This heat spreading structure features 1 mm straight fins and small fillets near the baseplate to improve print quality. A similar design to this can be easily obtained using an explicit geometric representation and the simplified boundary conditions.

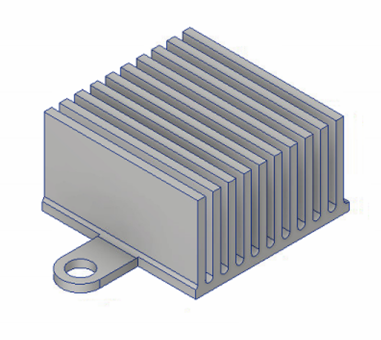


Figure 5.14: Reference heat spreader design.

[2] www.shapeways.com

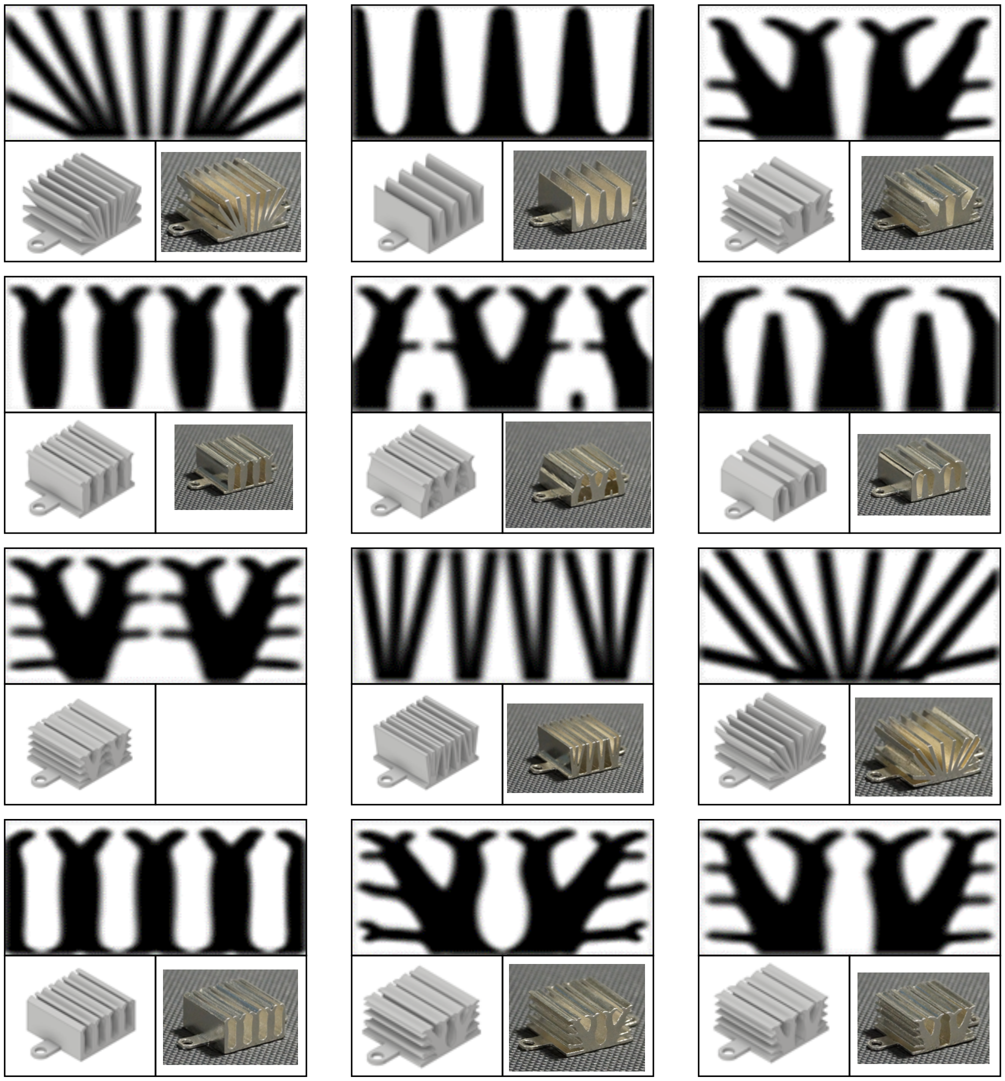


Figure 5.15: Selected designs produced by space colonization algorithm.

5.6 Experimental Evaluation

Experiments were conducted to measure the heat extraction capability of the heat spreading structures from 0 to 100°C. The experimental data was then used to extrapolate the maximum temperature of the heat sink given a 20 W

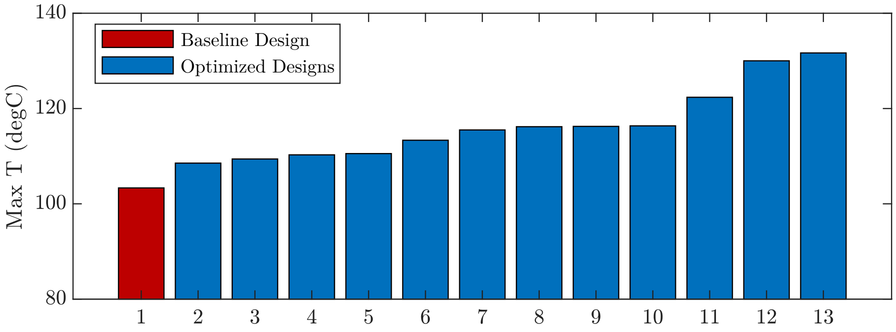
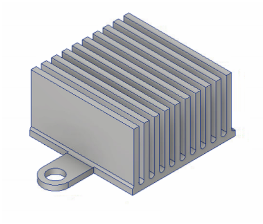
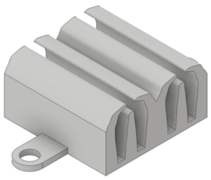


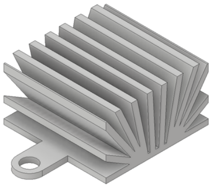
Figure 5.16: Maximum device temperature for each heat sink designs for 15 W of heat loss.



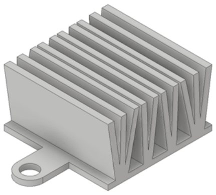
1



2



3



4

Figure 5.17: Best performing heat sinks designs.

heat loss input, as illustrated in Fig. 5.16. The reference ‘conventional’ heat spreader design outperformed all of the heat sink designs obtained from reduced order-optimization. The maximum temperature of the optimized designs ranged from 5-27% larger than the reference design.

Heat spreader performance as a function of problem formulation variation is presented in Fig. 5.18. There does not appear to be any correlation between experimental performance and the optimization objective, reduced-order physics, or boundary style. The solutions obtained using the GPM tended to perform better on average. This suggests that the best performing design may have simple geometric features.

The designs with the four lowest maximum temperature are presented in Fig. 5.17. The best performing design (1) was the straight fin structure. The next three designs had nearly identical performance and all feature simple geometric elements. Design (3) is a simple radial heat sink and Design (4) is a v-fin heat sink that has been investigated in literature [121]. Design (2) is more interesting in that it features both straight fins and some curved fins at a larger fin thickness.

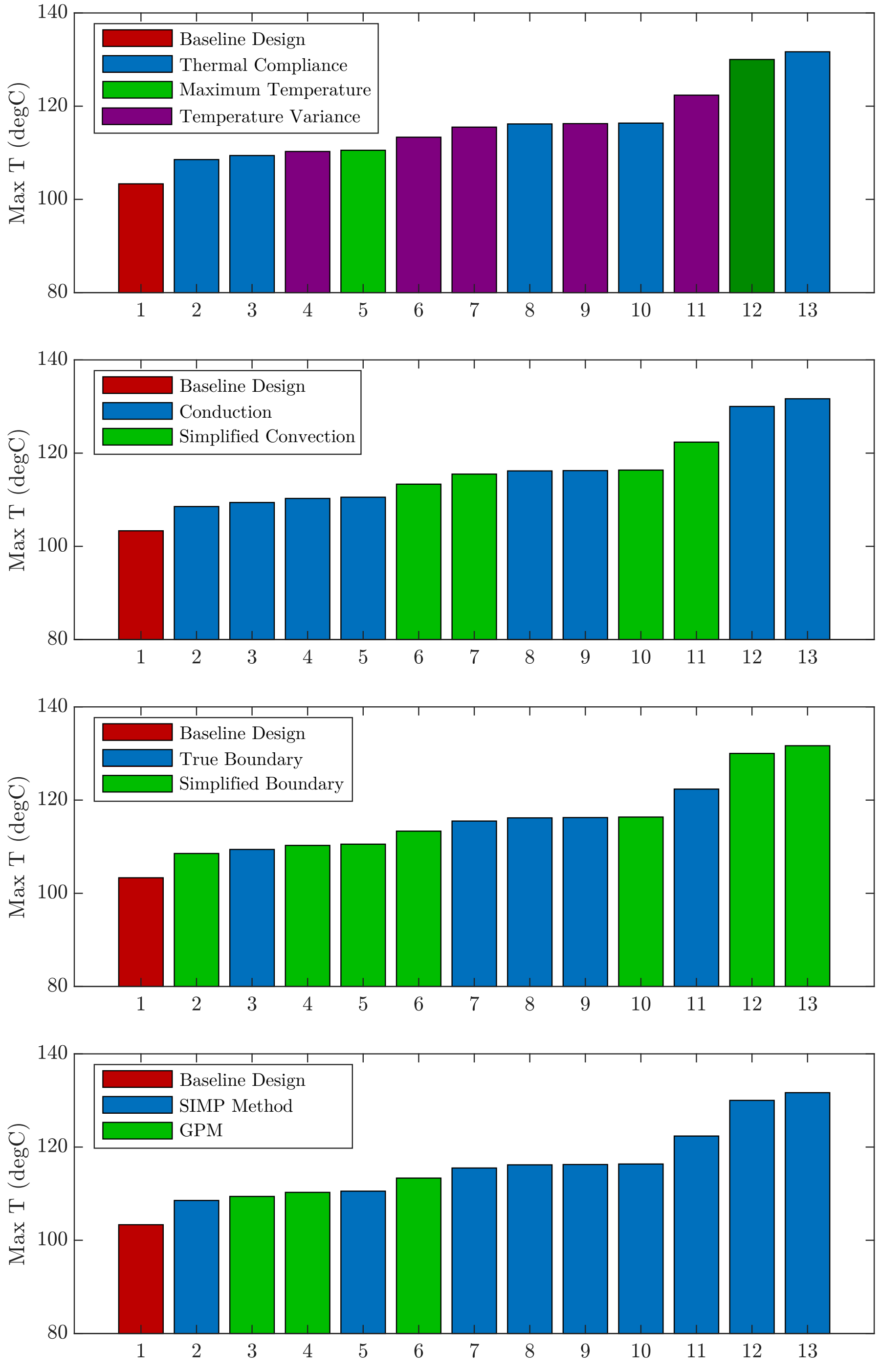


Figure 5.18: Maximum device temperatures for each heat sink design differentiated by problem formulation variations.

5.7 Discussion

Based on the numerous design studies and experimental validation, several observations can be made.

- Copper coated vias are a viable mechanism for heat transfer through the PCB given careful manufacturing. It was found that experimental results varied as much as 30% when using an alternative PCB. Furthermore, the packing configuration was able to maintain a temperature near 100°C for a 15 W power loss input. Extrapolating to the temperature limit of the device, the setup should manage 28W of loss. For a device with $\approx 99\%$ efficiency, a power level of 3 kW can be processed.

- The reduced order physics models were unable to reliably bias designs towards good performing structures. The best heat sink style was achieved using an explicit geometry representation with simplified boundary conditions. However, increasing the design flexibility of the GPM converged to similar topologies as SIMP.

- Reliably designing a heat spreading structure will require the solution of more advanced physics. This is been achieved by several researchers using conventional designs strategies described in this chapter, however, is not common with topology optimization methods. Methods utilizing a fixed analysis mesh require further enhancement before they can be used effectively for heat spreader design.

5.8 Summary

Based on the observations in Ch. 5, a more comprehensive study was conducted to identify a good reduced order method for heat spreader topology optimization. Heat spreader topology optimization was performed to enhance the performance of a through-PCB cooling circuit configuration. It was observed that using reduced order-models does not reliably produce structures which perform well experimentally and the topology optimization representation needs to be enhanced to account for complex boundary interactions.

A variety of design representations, reduced-order physics models, objective functions, and boundary condition simplifications are tested. A selection of promising and/or unique designs are selected for experimental validation. It was observed that simple designs outperformed more complex designs. Furthermore, conventional designs such as straight fin, radial fin, and v-shaped fin structures performed well. These structures were simply achieved using an explicit geometric representation, though their optimization convergence for this application was more coincidence. The best performing design consisted of straight fin structures with fillets. The current design representations were unable to reproduce these features. An analysis of the trade-offs between model-representations and suggested strategies to accommodate this best feature style are discussed in the next chapter.

Chapter 6

A comparison of structural representations for heat spreader design

6.1 Introduction

Researchers have developed a variety of problem representations for structural optimization. These vary from direct voxel representations [6], to indirect rule-based representations [7]. These methods vary in both the size of the representation, number of design variables, and access to the design space. In this chapter, several types of design representations are investigated in more detail as they relate to heat spreader structural optimization.

A method to optimize a completely free-form design representation on a finite-element mesh with gradient-based algorithms was first proposed by Bendsoe and Kikuchi [6], where the authors used direct optimization of element density to design optimal material distributions. This methodology was extended through abstraction to optimize level-set functions within a finite-element framework [2]. This abstraction restricted portions of the design space, however, it provided an explicit boundary representation for a given structure.

More recently, researchers have developed differentiable projections to optimize discrete geometric elements in both the density field [3] and level-set framework [122]. These methods allow for the explicit parametric representation of geometric elements during the optimization routine. The geometric elements can vary in complexity from simple bars, as in the aforementioned papers, to curved bars [4], and supershapes [123]. Furthermore, the geometric elements can be tuned independently, or linked into pre-defined layouts before optimization [124].

In the literature, the various styles of structural representation have been applied to heat spreader design problems. The first paper on heat conduction SIMP topology optimization utilized the finite volume method [81]. These definitions were extended to a finite element framework where the effect of boundary definition was analyzed [85]. A more recent paper challenges the optimality of dendritic structures and proposes lamellar structures as optimal through several examples [83]. There are also several examples of heat spreader designs generated from algorithmic abstraction, such as through the use of cellular automata [125], bionic growth [126], and space colonization algorithms [127].

In this chapter, three structural optimization methods are analyzed as they apply to the design of heat spreading structures. The heat spreader design problem is formulated in Sec. 6.2. First, the SIMP method is analyzed in Sec. 6.3. Then, the GPM method is used in Sec. 6.4. Finally, an implementation using the space colonization algorithm is presented in Sec. 6.5. These results are compiled and discussed in Sec. 6.6.

6.2 Heat Transfer Design Problem Formulation

In this section, the heat spreader design problem is presented. Consider the rectangular, 2×1 cm,^{2} design domain presented in Fig. 6.1. There is a 1 cm fixed temperature boundary, Γ_D, on the bottom surface and the remaining adiabatic surfaces, Γ_N, restrict heat flux. The entire domain area, Ω, generates heat and symmetry is used across the vertical axis to reduce the design

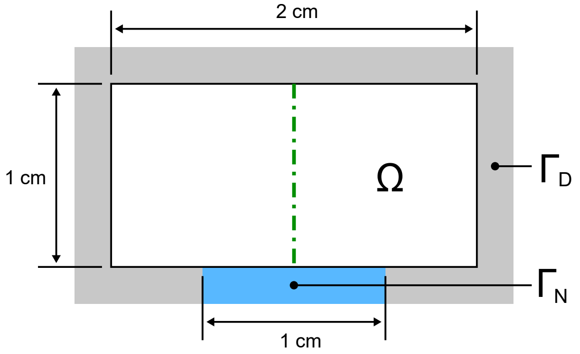


Figure 6.1: Homogeneously heated design domain.

and analysis problem dimension. This domain is governed by the following equations describing conductive heat transfer:

$$\nabla \cdot (\kappa \nabla T) + f = 0 \text{ on } \Omega, \tag{6.1}$$

$$T = 0 \text{ on } \Gamma_D, \tag{6.2}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{6.3}$$

A fixed input power of $f = 10$ kW is applied to the design domain, and the fixed temperature boundary is set to 0°C. Material will be allocated on the domain to adjust the thermal conductivity, κ, of a finite element between maximum and minimum values. The analysis problem is solved using a custom finite element code written in MATLAB. A finite element mesh of 200×200 elements is used for all the subsequent examples. The goal of the heat spreader as posed here is to extract heat from the domain to the fixed temperature boundary and hence reduce the maximum temperature on the domain. This heat spreader design task can be formulated as follows:

$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) = ||T||_p \tag{6.4}$$

$$\text{s.t.} \quad 0 \leq \mathbf{d} \leq 1 \tag{6.5}$$

$$\rho_e \in \{\rho_{\min}, \rho_{\max}\} \tag{6.6}$$

$$V(\mathbf{d}) \leq V_{\max} \tag{6.7}$$

$$R(\mathbf{d}) \geq R_{\min}, \tag{6.8}$$

where a maximum temperature minimization is performed using a p-norm objective, $p = 8$, with respect to a set of design variables, $\mathbf{d}$. The design variables differ between design representations, and will be described separately for each algorithm. The designs variables are normalized here between 0 and 1. Each finite element must consist of either a low thermal conductivity material, $\rho_{\min} = 1$ W/mK, or a high thermal conductivity material, $\rho_{\max} = 400$ W/mK. The amount of high conductivity material used is restricted by a volume constraint where the volume, $V(\mathbf{d})$, must be less than or equal to a prescribed amount, $V_{\max} = 0.5\%$. For the 2D design problem presented here, this constraint restricts the area of the optimized structure. The final constraint restricts the minimum feature size such that the radius of high conductivity material, $R(\mathbf{d})$, is greater than a prescribed value, $R_{\min}$. These constraints summarize the requirements for the heat spreader design task. The enforcement of these constraints will vary between design methods and the associated optimization problem formulations will be presented separately for each method.

6.3 SIMP Method

The first design method under investigation is the Solid-Isotropic Microstructure with Penalization (SIMP) method [81]. The design representation considers the density of individual voxels, or pixels in 2D, on a domain. The design variables are given by scalars parameters, α:

$$\mathbf{d} = \left[\alpha_1, ..., \alpha_n\right], \tag{6.9}$$

that tune the existence of material for each finite element. To enforce the minimum feature size requirement, Eq. (6.8), a linear density filter of the following form is applied:

$$\tilde{\rho} = \sum_j w_{ij} \alpha_j, \tag{6.10}$$

where a weighting function, w, is defined based on the centroid distances of neighboring elements:

$$w_{i,j} = \begin{cases} \frac{R-d(i,j)}{\sum(R-d(i,k))} & j \in N_i \\ 0 & j \notin N_i. \end{cases} \tag{6.11}$$

The filter weight, $w_{i,j}$, is scaled based on the difference between a prescribed minimum radius, R, and the distance between element centers $d(i,j)$, and a summation of neighboring element distances, $d(i,k)$. The filtered element densities are then modified by the following expression to produce the density, ρ, used by the finite element analysis.

$$\rho = \rho_{\min} + \left(\rho_0 - \rho_{\min}\right) \tilde{\rho}_i^{\gamma}. \tag{6.12}$$

To prevent singularity in the stiffness matrix, the elemental densities are allowed to vary between $\rho_{\min}$ and ρ_0. The designs are biased towards a binary distribution by using the penalization to the power, γ, which lessens the effect of partially-defined material.

6.3.1 Standard Implementation

The design optimization problem can be formulated as:

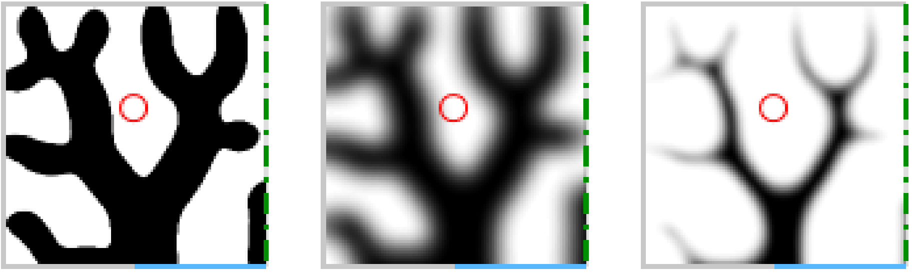
$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) = \max(T) \tag{6.13a}$$

$$\text{s.t.} \quad V(\mathbf{d}) \leq V_{\max} \tag{6.13b}$$

$$0 \leq \mathbf{d} \leq 1, \tag{6.13c}$$

where a single volume constraint is imposed and the minimum feature size requirement is overcome using the density filter. The SIMP method, as formulated in Eq. (6.13), can be used to solve the true design problem, Eq. (6.4). The adjoint method is used to obtain the derivatives for the objective and constraint functions, and the method of moving asymptotes (MMA) algorithm is used to solve the optimization problem. Since the SIMP penalization greatly decreases the effect of partially defined material, the filter radius was set to the minimum feature size. This biases the optimized design vector towards a minimum feature size of nearly half the filter radius. This is demonstrated by the solution to an optimization routine using a density filter [120], illustrated in Fig. 6.2.

Three versions of the optimized topology are shown, and finite element analysis is used to calculate the maximum temperature for each design. The design vector itself is nearly binary, satisfies the volume constraint, nearly satisfies minimum feature size requirements, and has a maximum temperature



Design vector
41.22 °C

Density filtered
19.58 °C

Penalized
135.00 °C

Figure 6.2: Optimized SIMP topology using a density filter. Minimum radius requirement size shown by red circle.

of 41.22°C. The density-filtered design satisfies the radius and volume constraints, however, does not satisfy the binary distribution requirement. This partially-defined boundary improves the performance of the heat spreading structure with a maximum temperature of 19.58°C. The optimization routine itself actually optimized the penalized and filtered topology to produce a structure with a maximum temperature of 135.00°C. A criticism of this strategy is the consistent use of a structure with partially defined boundaries to assess the performance of the structure and calculate derivatives. Though the optimization routine was being guided by the performance of the filtered and penalized structure, the resultant design vector produced a reasonable design. The minimum feature size requirement, Eq. (6.8), is satisfied throughout most of the domain excluding the bottom boundary. This is a product of the filter application and can be alleviated by decreasing the size of the design domain such that it is slightly smaller than the analysis mesh. This will enable the projection of elements densities outside the design domain within the analysis mesh and retain the minimum feature size.

6.3.2 Robust Implementation

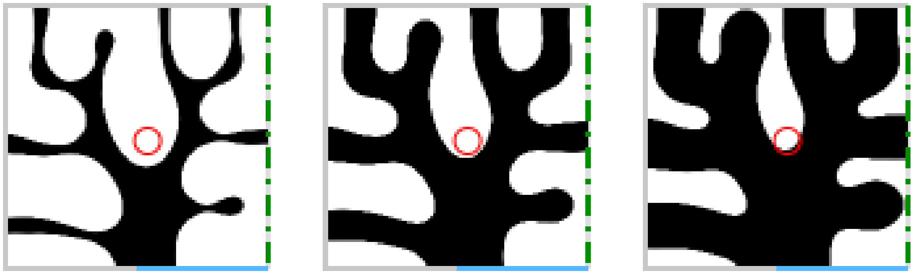
The boundary definition can be improved during the optimization by applying the following trigonometric filter after the density filter:

$$\tilde{\rho}(x) = \frac{\tanh(\beta - \nu) + \tanh(\beta(\rho - \nu))}{\tanh(\beta\nu) + \tanh(\beta(1 - \nu))}. \tag{6.14}$$

This function will minimize the transition between 0 and 1 as the parameter β increases at a transition density anchor point ν. Choosing a ν parameter of 1 results in a heaviside filter [128]. Using the heaviside filter for heat conduction topology optimization does not guarantee a binary solution. To obtain a near binary final material density, a robust formulation is adopted as proposed in Ref. [129]:

$$\begin{aligned}
\min_{\mathbf{d}} \quad & \max\left(||T^e||_p, ||T^i||_p, ||T^d||_p\right) && \text{(6.15a)}\\
\text{s.t.} \quad & \mathbf{K}(\rho^e)\mathbf{u}^e = \mathbf{f} && \text{(6.15b)}\\
& \mathbf{K}(\rho^i)\mathbf{u}^i = \mathbf{f} && \text{(6.15c)}\\
& \mathbf{K}(\rho^d)\mathbf{u}^d = \mathbf{f} && \text{(6.15d)}\\
& V(\mathbf{d}) \leq V_d^* && \text{(6.15e)}\\
& 0 \leq \mathbf{d} \leq 1. && \text{(6.15f)}
\end{aligned}$$

The anchor position, ν, for the eroded, intermediate, and dilated designs is chosen as 0.25, 0.5, and 0.75, respectively. A filter radius of 0.1 cm is chosen to result in a minimum feature size of 0.1 cm for the intermediate design. A beta continuation scheme is used where beta is doubled every 50 iterations for a total of 400 iterations. The optimized structure using the robust formulation is presented in Fig. 6.3.



Eroded Design
max(T) = 126.53 °C

Intermediate Design
max(T) = 81.54 °C

Dilated Design
max(T) = 50.81 °C

Figure 6.3: SIMP solution using robust formulation.

The eroded design is nearly binary and satisfies the volume constraint, however, does not satisfy a minimum feature size. The dilated design does satisfy the minimum feature size constraint, however, no longer satisfies the volume constraint. The intermediate design satisfies all constraints to produce a feasible design with a maximum temperature of 81.54°C. Optimization with an improved boundary definition for this test problem came at a cost of structural performance. Though the robust formulation as presented here provides a feasible solution at the end of the optimization, the intermediate iterations are still guided by structures with partially defined boundaries. This challenge may be addressed using a level-set formulation, however, changes in topological structure are restricted. The density based SIMP method typically produces improved designs, independent of these observations.

6.4 Geometric Projection Method

The geometric projection method (GPM) is based on parametrically-defined shapes which are projected onto a finite element mesh using a differentiable projection. This allows for simple calculation of gradients on a finite element mesh and eliminates the need to remesh the updated designs. The framework presented here is based-off the work of Ref. [115]. The design optimization formulation for this method is presented by the following equations:

$$\begin{aligned} \min_{\mathbf{d}} \quad & \Theta(\mathbf{d}) = \max(T) && \text{(6.16a)} \\ \text{s.t.} \quad & V(\mathbf{d}) \leq V_{\max} && \text{(6.16b)} \\ & 0 \leq \mathbf{d} \leq 1. && \text{(6.16c)} \end{aligned}$$

The design variables used to parametrically define a bar $\mathbf{d}_q$ have the following form:

$$\mathbf{d}_q = \left[x_0, y_0, x_f, y_f, t, \alpha\right]_q, \tag{6.17}$$

where the initial node coordinates, x_0 and y_0, and endpoint node coordinates, x_f and y_f, can be varied to adjust the position and length of the bar. A parameter, t, adjusts the thickness of bar and a parameter, α, can be used to scale the density of the bar from solid to void, similar to the SIMP method. As an extension of this work, a SIMP penalization is also applied to the entire density field after projections to improve the convergence properties of the algorithm.

The GPM formulation as presented here nearly satisfies the problem formulation in Eq. (6.4). To make the projections differentiable, the boundaries of the bars are smeared from 0 to 1. In this implementation, 5% of the bar edge is transitioned from 0 to 1 smoothly. The minimum feature size constraint can be biased by constraining the minimum width of the bars to the minimum feature size. This does not guarantee satisfaction in practice. The optimization problem is solved using the MMA algorithm, and the derivatives are calculated using the adjoint method. Refer to Ref. [115] for a detailed derivation.

One advantage of using explicit geometric representations is the ability to adjust the complexity of the final design [123, 130]. Consider the following four variations on the geometric design variables for a bar, q:

$$\mathbf{d1}_q = [x_0, y_0, x_f, y_f]_q \tag{6.18}$$

$$\mathbf{d2}_q = [x_0, y_0, x_f, y_f, t]_q \tag{6.19}$$

$$\mathbf{d3}_q = [x_0, y_0, x_f, y_f, \alpha]_q \tag{6.20}$$

$$\mathbf{d4}_q = [x_0, y_0, x_f, y_f, t, \alpha]_q \,. \tag{6.21}$$

In the first two cases, $\mathbf{d1}_q$ and $\mathbf{d2}_q$, the bars are simply moved across the domain to form a structure. In the second two cases, $\mathbf{d3}_q$ and $\mathbf{d4}_q$, the bar densities are considered design variables. Representations $\mathbf{d1}_q$ and $\mathbf{d3}_q$ use constant bar widths, and representations $\mathbf{d2}_q$ and $\mathbf{d4}_q$ consider the individual bar widths as design variables. Solutions using all of the design representations for a 12 bar structure are presented in Fig. 6.1. When using constant density bars, $\mathbf{d1}_q$ and $\mathbf{d2}_q$, the optimizer struggles to merge bars which are nearly touching. This can be alleviated using the bar density as a design variable, $\mathbf{d3}_q$ and $\mathbf{d4}_q$. The optimizer then struggles to obtain a solution with binary material properties when using the bar density as a design variable. Adding the ability to scale the width of each individual bar increases the design freedom,$\mathbf{d1}_q$ and $\mathbf{d3}_q$. However, in practice the optimizer converges to a worse performing design. For the remaining design problems, $\mathbf{d3}_q$ will be used as the design vector.

Another strategy to scale the complexity of the design is simply to vary the number of bars used in the optimization. Four different bar number initializations and their resultant optimized structures are presented in Fig. 6.2. As the number of bars increases, there is a potential improvement in performance. This can be seen from the optimization results as the maximum temperature dropped from 54.16°C using 6 bars to 37.1°C using 18 bars. However, the optimized structure using 24 bars has a temperature increase of 2% when

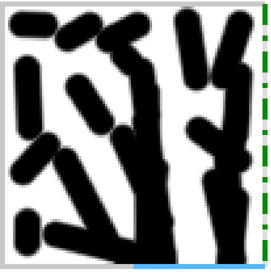
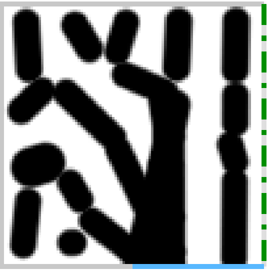
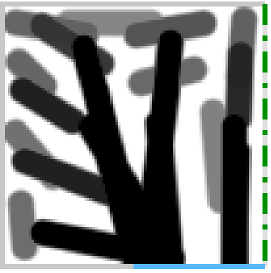
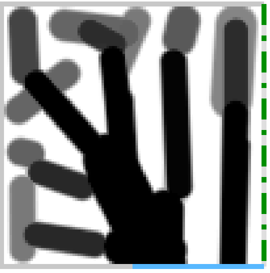
 **d1**	 **d2**
max(T) = 42.94°C	max(T) = 45.08°C
 **d3**	 **d5**
max(T) = 37.99°C	max(T) = 40.49°C

Table 6.1: Complexity tuning by increasing number of representation parameters.

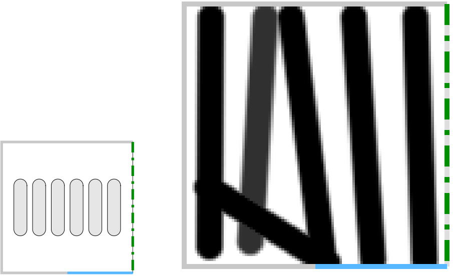
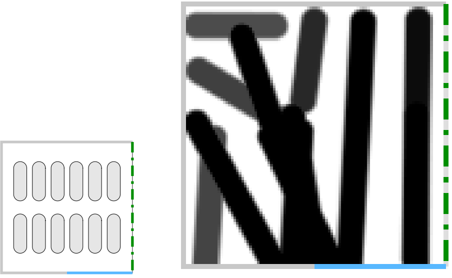
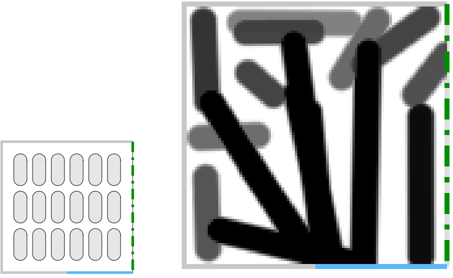
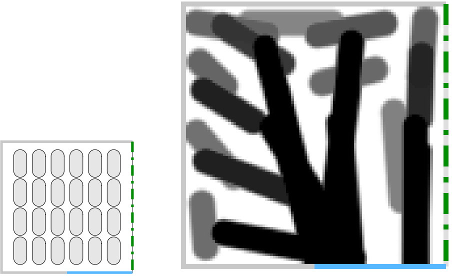
 max(T) = 54.16°C	 max(T) = 44.57°C
 max(T) = 37.10°C	 max(T) = 37.99°C

Table 6.2: Complexity tuning by varying bar numbers.

compared to the 18 bar solution. In the limit of introducing more bars into the design problem, smoother SIMP-like solutions can be obtained. Consider the optimized structure when using 100 projected bars, shown in Fig. 6.4.

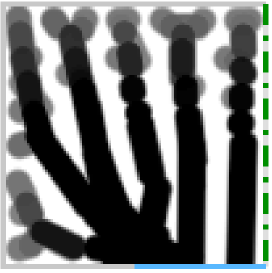


Figure 6.4: GPM solution with 100 design variables.

Here, the optimal structure has a maximum temperature of 29.04°C, and the shape of the structure is smoothed out. The optimizer struggled to obtain a binary solution and there are still instances where the minimum feature size requirement is not satisfied. As such, it may be worthwhile to use GPM as an exploratory tool and rely on SIMP methods for detailing. A similar strategy using a generative algorithm as an exploratory tool was shown to consistently improve structural performance after SIMP detailing [8].

When using uniformly distributed bars locations for a smaller number of bars, such as 24, the initial configuration of the method plays a large role in the optimized structure obtained. There are many styles of initialization for geometric projections that have been investigated in literature, whether straight [115], square [124], or crossed-bars [122]. When uniformly distributing these bar configurations with 100 elements, the domain is nearly solid and the optimized solutions resemble each other. Here, four different starting configurations using 24 discrete geometric elements, a common denominator for all initial conditions, are considered. The initial bar configurations and the optimized density fields are presented in Table 6.3.

A variety of radial fin structures are obtained from the different starting positions. The maximum temperature of these structures varies as does the amount of partially density bars present in the converged solution. As such, it is unfair to compare their performance directly. However, the structural variance offers insights into performance as the designs converged to local optima. For example, the cross-bar initialization resulted in an optimized design with cross bars at the branch extremities. The horizontal bar initialization resulted in an optimized design with horizontal bars at the branch extremities. The GPM provides a simple reduced-order representation that can be used to define unique feasible initial starting conditions with tunable representation complexity. Overcoming the observed challenges would make the GPM a valuable tool for heat spreader structural optimization.

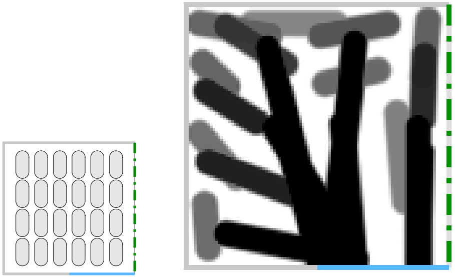
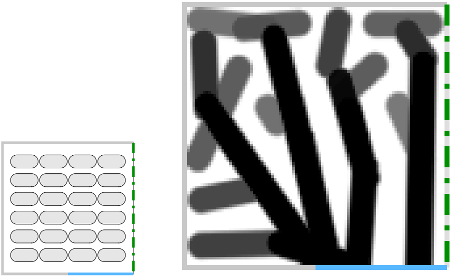
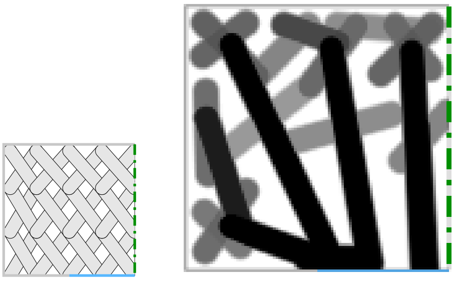
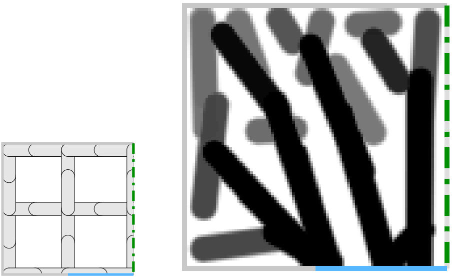
	
max(T) = 37.99°C	max(T) = 35.61°C
	
max(T) = 36.53°C	max(T) = 36.31°C

Table 6.3: Projection solutions based on different starting points.

6.5 Space colonization algorithm

The final method under consideration is based on the use of a low-order design representation to perform a targeted search of the design domain using a generative algorithm. Here, we significantly extend the search capability of the algorithm as presented in Ref. [127] to allow for a diverse search in the structural space. A sketch of the algorithm is illustrated in Fig. 6.5.

The algorithm is initialized with a set of root nodes, shown in red. Auxins, open blue circles, are introduced into the domain to influence the growth of the branches. Each auxin is paired with its nearest branch node and the branch nodes grow in the average direction of its auxins by a fixed step. The algorithm iterates through this process until all auxins have been reached by a branch node. At this point, nodal information is used to create a series of bar segments

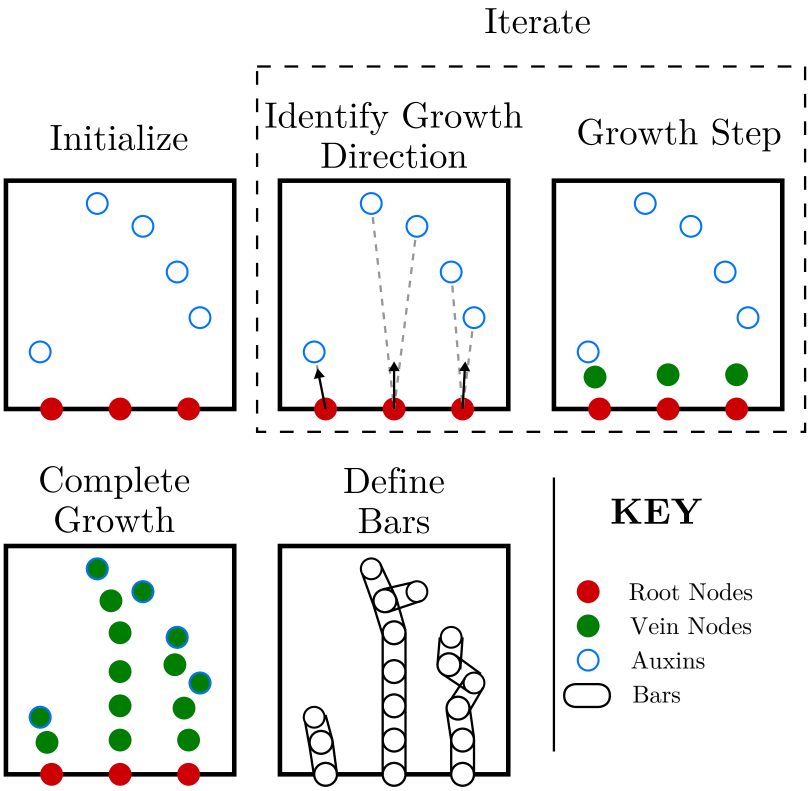


Figure 6.5: Space colonization algorithm.

to represent the structure in 2D. Note that the same differentiable projection is used here as with the GPM in order to reuse the finite element mesh. A fast force-directed meshing strategy is proposed as alternative meshing strategy in Reg. [8]. The design variables using the space colonization algorithm are given by the following parameters,

$$\mathbf{d}_i = [n_s, n_g, l_g, ax_1, ay_1, ..., ax_n, ay_n]\,, \tag{6.22}$$

including the number of initial stems, n_s, the number of growth stages, n_g, the length of the growth step, l_g, and the x and y locations of the auxins, $[ax_i, ay_i]$. The inclusion of the numbers of stems, growth stages, and growth length greatly increases the flexibility of the method and allows the optimizer to tune the complexity of the design. In our previous work, [101], we design the algorithm to target dendritic structures specifically. However, the space colonization algorithm is much more flexible, and can target a wider variety of structures. Consider the following variations in design given a fixed set of auxins as illustrated in Fig. 6.4. Increasing the number of initial stem nodes naturally creates fin-like structures. The same effect can be achieved by introducing the auxins in stages, for example starting with the first two auxins initially, then introducing the second two after the first growth stage terminates. In the implementation of the algorithm here, both the number of initial nodes and the growth stages are used as design variables to allow the optimizer to tune the complexity of the optimized solution. A genetic algorithm can be

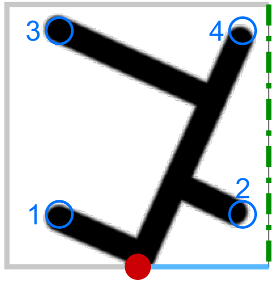
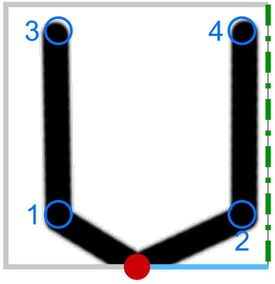
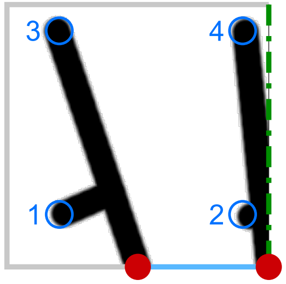
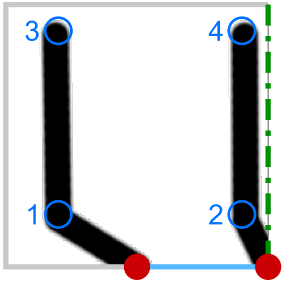
	1 stage	2 stages
1 stem		
2 stems		

Table 6.4: Algorithm variations.

used to solve the following unconstrained numerical optimization problem:

$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) = \max(T) \tag{6.23a}$$

$$\text{s.t.} \quad 0 \leq \mathbf{d} \leq 1. \tag{6.23b}$$

The minimum radius constraint is inherently satisfied here using discrete geometric projection of bars in a layout form, i.e., endpoints are always connected. However, the volume constraint may not be satisfied. For initial projections that do not satisfy the volume constraint, a multiplication penalty factor of 1.5 is applied to the maximum temperature. For initial conditions that are below the volume constraint, the following single variable optimization problem is solved to enforce equality:

$$\min_{t} \quad V(t) - V_{\max} \tag{6.24a}$$

$$\text{s.t.} \quad 0 \leq t \leq 1, \tag{6.24b}$$

where a tunable parameter, t, is adjusted until the volume of the projected structure is within 1% of the volume constraint value. To solve this optimization problem, a simple bisection algorithm is implemented. This inner loop optimization problem is solved to provide a fair comparison between topologies. Both the SIMP method and GPM will find optimized structures at the volume constraint.

Since the outer-loop optimization is driven by a meta-heuristic algorithm, it is important to understand the implications of the optimization parameters. A series of studies is conducted by varying the initial population size and maximum number of generations; the results are summarized in Table 6.5.

The number of development stages is allowed to vary between 1 and 4. The size of the growth step is allowed to vary between 4% and 9% of the domain, the number of stem nodes is allowed to vary between 1 and 4, and a total of 40 auxins are introduced on the domain.

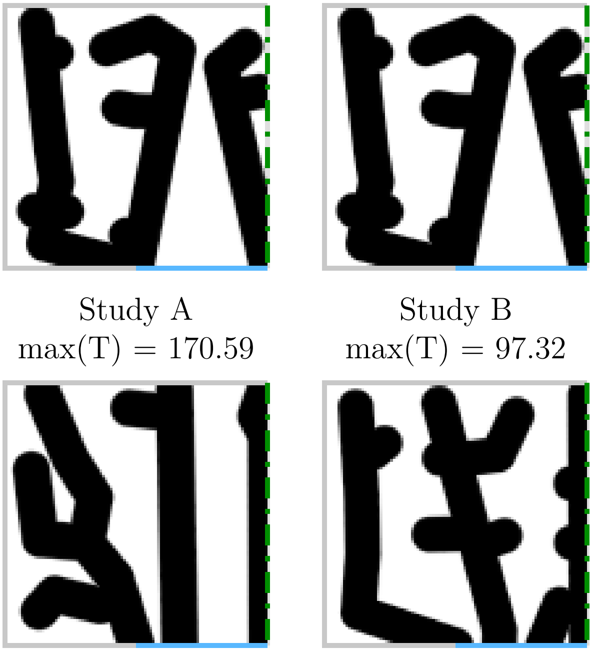
Study	GOP	GEN	max(T)	Var(T)
A	10	10	170.59	21.66
A	100	10	97.32	10.31
B	100	100	73.93	7.51
C	1000	10	81.96	6.82

Table 6.5: GA search sensitivity, results presented are the best of 10 trials.

As the total number of evaluations increases, the optimizer begins to have more consistent convergence. Given a fixed number of evaluations, and varying the population and generation size, the exploitation characteristics become evident as better designs are found using more generations. The best obtained designs from each set of studies are presented in Fig. 6.6.

Each execution of the genetic algorithm was able to produce a sufficiently unique design. Since the optimization is driven by a genetic algorithm with a small number of generations, convergence to a local optimum can not be guaranteed. Many of the optimized structures performed comparably to that obtained by the robust SIMP formulation, however, not as well as the standard SIMP formulation. This may be due to the simple constant thickness projection of bars on the design domain. The better performing designs were observed to have bulk material located near the heat sink and smaller features at the edges of the domain. The performance of the SCA implementation may be improved by including the thickness of individual bars as designs variables, as similarly implemented in previous research [8].

Using the number of development stages and initial stems as designs vari-



Study A
max(T) = 170.59

Study B
max(T) = 97.32

Study C
max(T) = 73.93

Study D
max(T) = 81.61

Figure 6.6: Best designs obtained from various studies.

ables in this implementation enabled tuning the complexity of the designs. The best performing designs all started with initial stem nodes and featured 2 development stages. Note that the branches did not always propagate from every stem node. This design variables can be fixed in subsequent optimization runs to perform a targeted search of likely well-performing designs.

6.6 Discussion

The best topology obtained by each design method is presented in Fig. 6.7. The best SIMP method optimized topology presents a structure with a maximum temperature of 41.22°C (left image). This topology resembles a branching pat-

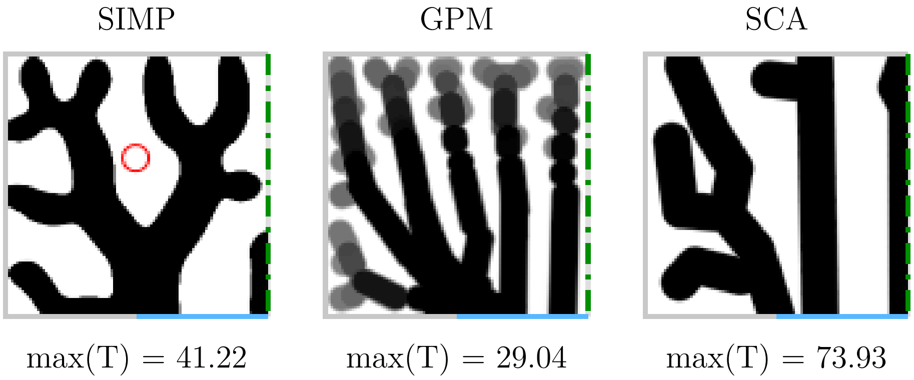


Figure 6.7: Best topologies.

tern as typically found starting from a homogeneous material distribution [81]. The best GPM optimized topology has thin and long fins extending from the heat sink (center image). These resemble those obtained in Ref. [83] when using SIMP with an alternate initial condition. This topology has a maximum temperature of 29.04°C, which may be enhanced by the presence of bars with partially defined material density. The best topology obtained using the space colonization algorithm (right image) has a maximum temperature of 73.93°C. This structure satisfies both minimum radius and volume constraints, however, suffers from poor performance.

The SIMP optimized topology clearly presents the most refined structure. The structure naturally varies in width and the connection between branches has smooth fillets. The GPM topology produces a straight-fin structure, which is likely the more optimal, but the optimized structure could use some refinement. There are several bars near the ends of the branches which are partially defined. Additionally, some of the bars appear to be minimally connected, resembling checkerboard phenomena when using SIMP without a filter. Furthermore, using this explicit representation, it is difficult to reproduce the smooth fillets that are present when using the SIMP method. The optimized topology using the SCA does not suffer from the same checkerboard phenomena since the layout of the topology is implicitly constrained. Obtaining a more refined topology using this representation can be achieved by decreasing the step growth size of the algorithm and allowing the thickness of the bars to change as design variables. Refer to Ref. [8] for an implementation with circular nodes projected.

When using the two gradient based methods, SIMP and GPM, the actual topological changes are obtained within the first few optimization iterations. The remaining iterations are simply improving the boundary representation and/or performing size optimization to stretch the features. The initial condition for the SIMP method defines what optimized structure will emerge since the designs are mesh-independent with the filter. The complexity of the optimized structure can be somewhat tuned using the GPM representation as bars are increased. After a certain point, designs begin to converge to a similar structure. The space colonization algorithm produces large variations between topologies in every generation. This is closer to an actual topology optimization method. However, some form of size optimization needs to be implemented to produce a fair comparison between topological structures. This can be achieved as with simple sizing parameters, or by feeding a candidate topology into either the SIMP method or the GPM for fine tuning.

6.7 Summary

In this chapter, a practical investigation of several structural optimization representations for heat spreader design was performed. The SIMP-voxel representation provided the highest design flexibility, which was reflected in the smooth structures resultant from optimization. The SIMP method converged to the best performing design. The GPM explicit geometric representation provided a lower dimension design representation and enabled tuning of solution complexity. In this study, it was shown that the GPM requires some reformulation to produce feasible structures. Using the generative space colonization algorithm, the optimizer was able to tune the complexity of the optimized structure to produce both simple fin and dendritic structures. Due to the projection of uniform bars, this method was not as successful as in past implementations [8, 127]. Through these studies, it was noted that the complexity of the optimized structures varied greatly based on method and initialization. No single method allowed for the direct tuning of complexity in the optimized structure to produce well-refined designs. Initial work in measuring and tuning complexity was proposed in Ref [130], however, there are still open challenges left to address before these methods can be practically adopted.

Chapter 7

Explicit Geometric Projection Methods for Heat Spreader Design

7.1 Introduction

In Chapter 6 it was observed that the geometric projection method (GPM) did not produce feasible structures for heat spreader design. When the GPM is used for solid mechanics, the optimized structures tend to produce feasible designs [115, 124]. In this chapter, the behavior of geometric projection methods in the context of heat spreader design is analyzed.

The use of explicit geometric elements within a fixed finite element mesh framework was first proposed in Ref. [3]. A SIMP-like penalization was later included on the density of each geometric element to enhance this geometric projection [115]. A second projection strategy was developed in parallel where explicit geometric elements are projected using a level-set representation [122] using the extended finite element method to analyze the structure. Both methods offer the potential to achieve topological changes with the advantage of having parametrically defined elements.

The simple geometric representations originally used in these methods have been advanced to include curved bars [4] and supershapes [123] to produce more natural structures. Another advantage of this class of methods is the ability to control the complexity of the final solution by changing the number of geometric elements used in the optimization. This is not achievable using

SIMP or level-set methods when minimum feature size is required.

However, an explicit feature size requirement is not naturally satisfied when using discrete geometric elements. Consider the following situation presented in Fig. 7.1.

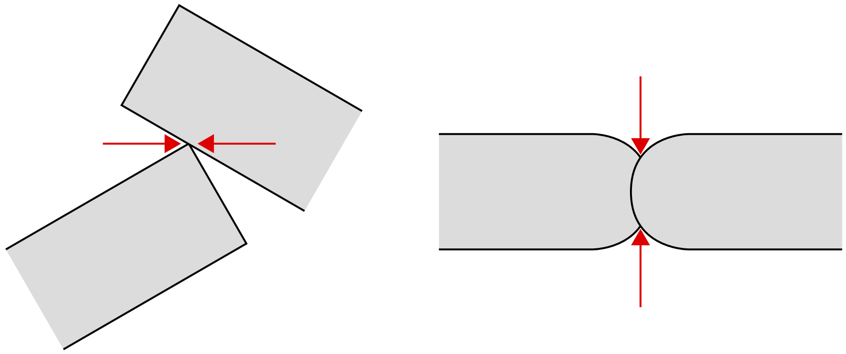


Figure 7.1: Bar intersection situation.

A situation may occur when two geometric elements are minimally intersecting, thus creating a topological element with a new thickness. This issue manifests itself frequently when solving a heat conduction topology optimization problem using discrete geometric elements. One strategy to overcome this challenge is to constrain the layout of the geometric elements such that they are connected at endpoints, then optimize the shape of the resultant structure [124]. This guarantees that a topology maintains a minimum feature size implicitly, however, restricts the design space.

In this chapter an alternative strategy for satisfying the minimum feature size constraint is proposed to allow more access to the design space. First a sketch of the geometric projection is provided in Sec. 7.2. The spreader design problem is defined in Sec. 7.3 and the challenges with the implementations are outlined. Two strategies to remedy the shortcomings for the algorithm as presented in Sec. 7.4 and the results are discussed in Sec. 7.5. The chapter is then concluded in Sec. 7.6.

7.2 Method

In this section, a sketch of the geometric projection method is provided. The geometric projection method used here is modeled after the implementation of Ref. [115] with some modifications. In this section, bars with uniform width, t, and semi-circular ends are considered. The geometric bars are parameterized with the coordinates of the endpoints, $\boldsymbol{x}_{q0}$ and $\boldsymbol{x}_{f0}$, and the width of the bar, t. The signed distance, d_q, from a point, $\boldsymbol{p}$, to the edge of a bar, q, is given by the following equation:

$$\boldsymbol{\phi}_q(d_q(\boldsymbol{x}_{q0}, x_{qf}, \boldsymbol{p}), t) := d_q(\boldsymbol{x}_{q0}, \boldsymbol{x}_{qf}, \boldsymbol{p}) - \frac{t}{2}, \tag{7.1}$$

as the difference between the distance from the central axis of the bar segment, d_q, and half the bar width. The distance between the central axis of the segment and a point can be calculated using the following conditions:

$$d_q(\boldsymbol{x}_{qf}, \boldsymbol{x}_{qf}, \boldsymbol{p}) := \begin{cases} ||\boldsymbol{b}|| & \text{if } \boldsymbol{a} \cdot \boldsymbol{a} \geq 0 \\ ||\boldsymbol{g}|| & \text{if } 0 < \boldsymbol{a} \cdot \boldsymbol{b} < \boldsymbol{a} \cdot \boldsymbol{a} \\ ||\boldsymbol{e}|| & \text{if } \boldsymbol{a} \cdot \boldsymbol{b} \geq \boldsymbol{a} \cdot \boldsymbol{a}, \end{cases} \tag{7.2}$$

depending on whether the candidate point is closer to the endpoints of the central axis or endpoints of the bar. These conditions can be checked using the following equations:

$$\boldsymbol{a} := \boldsymbol{x}_{qf} - \boldsymbol{x}_{q0} \tag{7.3}$$

$$\boldsymbol{b} := \boldsymbol{p} - \boldsymbol{x}_{q0} \tag{7.4}$$

$$\boldsymbol{e} := \boldsymbol{p} - \boldsymbol{x}_{qf} \tag{7.5}$$

$$\boldsymbol{g} := \left[I - \frac{1}{||\boldsymbol{a}||^2} \boldsymbol{a} \otimes \boldsymbol{a} \right] \boldsymbol{b} =: \mathbb{P}_a^{\perp} \boldsymbol{b}. \tag{7.6}$$

To enable the reuse of the finite element mesh and allow for the calculation of smooth derivatives for the projected bars, the edges of the bars are smeared using the following relationship:

$$\bar{\boldsymbol{\rho}}_q(d_q(\boldsymbol{x}_{qf}, \boldsymbol{x}_{x0}, \boldsymbol{p}), r) = \begin{cases} 0 & \text{if } \phi_q > 0 \\ \frac{1}{\pi r^2} \left[r^2 \text{arccos} \left(\frac{\boldsymbol{\phi}_q(d_q)}{r} \right) \right. & \\ \left. - \boldsymbol{\phi}_q(d_q) \sqrt{r^2 - \boldsymbol{\phi}_q(d_q)^2} \right] & \text{if } -t < \phi_q \leq t \\ 1 & \text{if } \phi_q < -r, \end{cases} \tag{7.7}$$

At this point the implementation differs from that presented in Ref. [123]. The penalized projected density field of a bar, $\hat{\boldsymbol{\rho}}_q$ is given by the following equation:

$$\hat{\boldsymbol{\rho}}_q = \rho_{\min} + \bar{\boldsymbol{\rho}}_q (1 - \rho_{\min}) \alpha_q^{\gamma} \tag{7.8}$$

where the bar projected element density, $\bar{\rho}_q$ is multiplied by a design variable, α, raised to the power γ. To prevent a division by zero in the sensitivity, the minimum density fraction is added to this quantity. The bar projected element densities are then merged using the following norm function:

$$\hat{\boldsymbol{\rho}} = \left[\sum_{q=1}^{N} \hat{\boldsymbol{\rho}}_q \right]^{1/p}, \tag{7.9}$$

to form a single projected density field, $\hat{\boldsymbol{\rho}}$, for all bars. For a finite norm value, the merged density value lies between $\rho_{\min}$ and some value > 1. To limit the maximum material density, a maximum function is used,

$$\rho = \max(\hat{\rho}, 1) \tag{7.10}$$

This creates some non-smoothness in the derivative, however, in practice works well. Before solving the finite element problem, another SIMP penalty is applied to the material density to scale the thermal conductivity, κ, of each finite element. This simply lessens the effect of the partially-defined edges, minimizes the void projections, and improves convergence. The projected density is used to scale thermal conductivity using the following equation:

$$\kappa = \kappa_{\min} + (\kappa_0 - \kappa_{\min})\rho^{\gamma}, \tag{7.11}$$

Since the void elements have near zero value in ρ after powered aggregation, a minimum value of $\kappa_{\min}$ is added to scale the thermal conductivity. The same penalization parameter, γ, is used for both the bar penalization and finite element density penalization. To improve the effect of the penalization, the bar penalty for the constraint, γ_v, is set to 1 as suggested by Norato et al. [115].

7.3 Heat spreader topology optimization.

In this section, the heat spreader design problem is formulated for the GPM design representation. Consider the following 1×1 m^2 design domain, shown in Fig. 7.2.

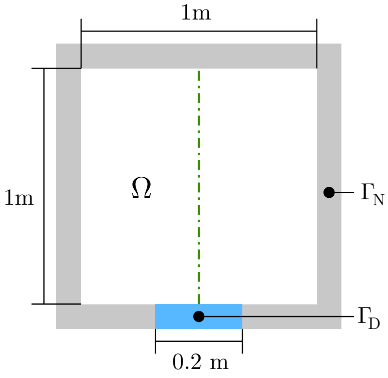


Figure 7.2: Heat conduction design domain.

There is a 0.2 m fixed temperature boundary, Γ_D, set to a fixed temperature of 0°C as shown in the figure. The remaining boundaries, Γ_N, are adiabatic and restrict heat flux. Symmetry is enforced on the vertical central axis to reduce computational expense, and the system is governed by the following equations for steady state conductive heat transfer:

$$\nabla \cdot (\kappa \nabla T) + f = 0 \text{ on } \Omega, \tag{7.12}$$

$$T = 0 \text{ on } \Gamma_D, \tag{7.13}$$

$$(\kappa \nabla T) \cdot \mathbf{n} = 0 \text{ on } \Gamma_N. \tag{7.14}$$

A fixed input power of $f = 1$ kW is applied to the design domain and the goal of the heat spreader is to direct this heat towards the fixed temperature boundary. The thermal conductivity, κ, of a finite element is allowed to vary between 4 and 400 W/mK.

The optimization problem formulation is given by the following expression:

$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) = \int_{\Omega} qT \tag{7.15}$$

$$\text{s.t.} \quad 0 \leq \mathbf{d} \leq 1 \tag{7.16}$$

$$V(\mathbf{d}) \leq V_{\max}. \tag{7.17}$$

The objective, $\Theta(\cdot)$ is to minimize the thermal compliance of the system, defined as the sum of the product of heat flux, q, and temperature, T. This is subject to a volume constraint where the volume of the structure, $V(\mathbf{d})$ must be less than a prescribed value, $V_{\max} = 50\%$. It is desirable that the structure retains a minimum feature size and so the bars are set to a fixed width of t.

Solving the heat conduction problem with a low penalization parameter, $\gamma = 3$, results in designs with partially-defined geometric elements as shown in Figure 7.3. The partially-defined geometric elements are similar to the gray-scale boundary observed when using the SIMP method. The gray-scale boundaries can be alleviated by adopting a continuation strategy for the SIMP penalization, or by simply choosing a larger penalization value, $\gamma = 6$, as depicted in Fig. 7.4.

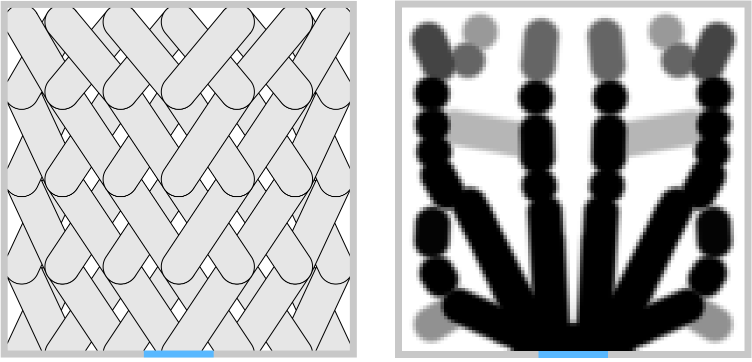


Figure 7.3: Heat spreader optimization with a SIMP penalty of 3. Initial condition (left). Optimized design (right).

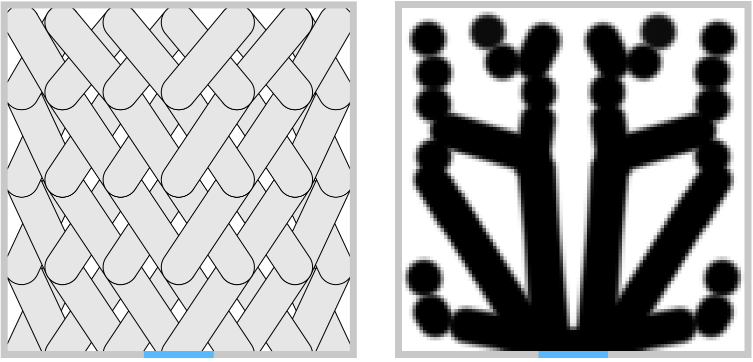


Figure 7.4: Heat spreader optimization with a SIMP penalty of 6. Initial condition (left). Optimized design (right).

Using a larger penalization successfully eliminated bars with partially-defined material properties and produced a structure. Though the material density is now well defined, the minimum structural thickness is no longer bounded by the thickness of the bar. In several locations, the bars are minimally interfaced, resulting in thin thermally conductive paths. This is very similar to the checker-boarding patterns that result when using the SIMP method without a filter. These partial connections create a minimum feature size less than the desired value of t. Two candidate strategies to alleviate this issue are presented in the following section.

7.4 Layout-Constrained Optimization

Constraining the layout of the bar elements is a strategy to alleviate the checkerboard patterns that result when using explicit geometric elements for heat spreader optimization. There are two general types of constraints that can be applied: implicit and explicit constraints. Constraining the layout of the geometric elements reduces access to the solution space, however, it guarantees feasible optimized designs with respect to the constraints. Optimization results using both types of constraints are presented in the following subsections.

7.4.1 Implicitly constrained

One strategy to guarantee a minimum feature size is to fix the topology of the projected bars and simply perform shape optimization. This strategy was first implemented using bar projection in Ref. [124], where the authors used a ground structure comprised of squares. Here, a cross-bar ground structure is used and the design vector is collapsed to only consist of unique nodes, ie., bar endpoints that lie at the same coordinates share a single, [x,y], design variable. Consider this following example to create a square structure, Fig. 7.5.

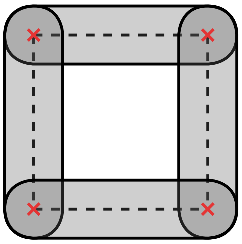


Figure 7.5: Implicitly constrained layout of a square.

The structure can be represented by 8 total design variables given by the x and y-coordinates of the intersecting nodes. The design optimization problem formulation is given as such,

$$
\begin{aligned}
\min_{\mathbf{d}_u} \quad & \Theta(\mathbf{d}_u) = \int_\Omega qT && (7.18)\\
\text{s.t.} \quad & 0 \leq \mathbf{d}_u \leq 1 && (7.19)\\
& V(\mathbf{d}_u) \leq V_{\max}, && (7.20)
\end{aligned}
$$

where the design variables now consist of only the unique nodal coordinate pairs, $\mathbf{d}_u$, to implicitly enforce the layout. The optimized structure using an implicitly constrained layout is presented in Fig. 7.6.

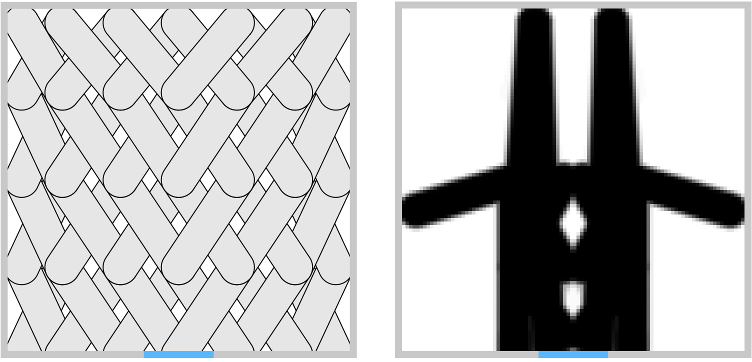


Figure 7.6: Optimized solution with fixed layout. Initial layout (left), optimized layout (right).

The optimized structure clearly avoids the checkerboard patterns that were obtained when not enforcing a layout. In explicitly enforcing the layout, the performance of the optimized structure decreased significantly to 14,310 WK. Using this definition, checkerboard patterns might still arise since the bar existence parameters, α, are used. Feasibility can be guaranteed by removing the design variables and assigning full density to all bars. Though this optimization implementation produced a feasible design, a large performance degradation was observed. This may be due to the restrictions in design space resultant from the fixed layout.

7.4.2 Explicit Optimization Constrained

An alternative strategy is to constrain the intersection of local bars in the optimization problem. If the center axis of the two bars do not overlap, then the minimum distance between the two bars can be calculated by checking the distance between each bar and associated endpoint, Eqn. (7.2). If the center axes of the two bars overlap, this condition is detected separately. One strategy to do so is to extend the line segments to infinity and calculate the intersection of the lines. If this intersection point lies within the bar segment, then the distance between the line segments is zero. A computer implementation of this algorithm is adopted from publicly available geometric algorithms^0 . This distance calculation can be used to numerically enforce intersection between all bars during the optimization. Consider the following constraint function applied between all bar pairs:

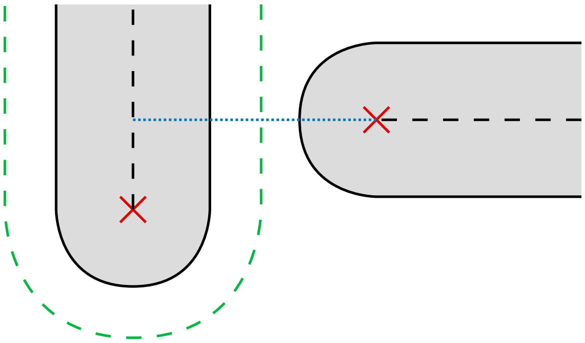


Figure 7.7: Bar snap constraint representation.

[2] http://geomalgorithms.com/a07-_distance.html

$$O(\mathbf{d}) = \begin{cases} -1 & \text{for } d_q \leq \frac{t}{2} \\ d_q & \text{for } \frac{t}{4} \leq d_q \leq \frac{3t}{2} \\ -1 & \text{for } \frac{3t}{2}, \end{cases} \tag{7.21}$$

where bars that lie within a minimum distance of $\frac{3t}{2}$ are forced to intersect within a tolerance of $\frac{t}{2}$. If two bars are intersecting within a tolerance, the constraint is inactive with zero derivative. If two bars are outside of the activity range, the constraint is inactive with zero derivative. Bars that lie within the two boundaries are biased towards each other. This provides an additional degree of freedom in the optimization when compared to the implicitly constrained layout where bars can now slide along each other and still satisfy constraints. The same square layout can be constructed using this formulation, shown in Fig. 7.8.

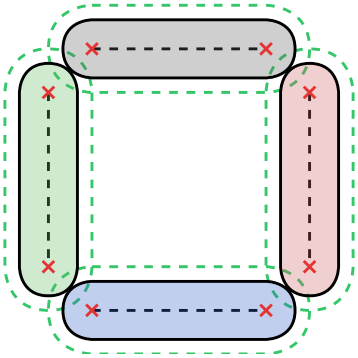


Figure 7.8: Explicitly constrained layout optimization of a square.

Here a total of 16 design variables and 6 intersection constraints are required to enforce the layout. The design optimization problem formulation using explicit constraints is given as such:

$$\min_{\mathbf{d}} \quad \Theta(\mathbf{d}) = \int_\Omega qT \tag{7.22}$$

$$\text{s.t.} \quad 0 \leq \mathbf{d} \leq 1 \tag{7.23}$$

$$V(\mathbf{d}) \leq V_{\text{max}} \tag{7.24}$$

$$\mathbf{O}(\mathbf{d}) \leq 0, \tag{7.25}$$

where the overlap constraint, $\mathbf{O}(\cdot)$, is simply added to the optimization formulation. This constraint implementation works well when starting with a feasible layout, such as the cross-bar pattern shown in Fig. 7.9. The optimization algorithm produced a solution with a thermal compliance value of 2,402 WK. The design is completely binary and does not feature any floating bars. There are small thickness reductions at some of the bar interfaces, however, these can be alleviated by tightening the intersection constraint bounds.

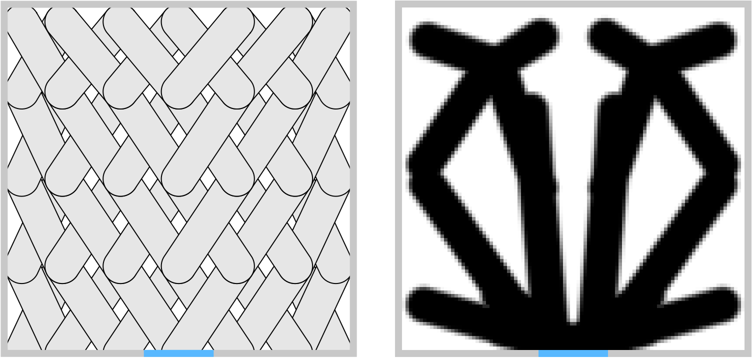


Figure 7.9: Explicitly constrained layout optimization. Initial condition (left), optimized topology (right).

One advantage of the explicit constraints is the ability to toggle the constraint during optimization. For example, consider allowing the optimizer to run naturally and toggling the intersection constraint after 15 iterations. The result of this approach is shown in Fig. 7.10. The optimized structure has a

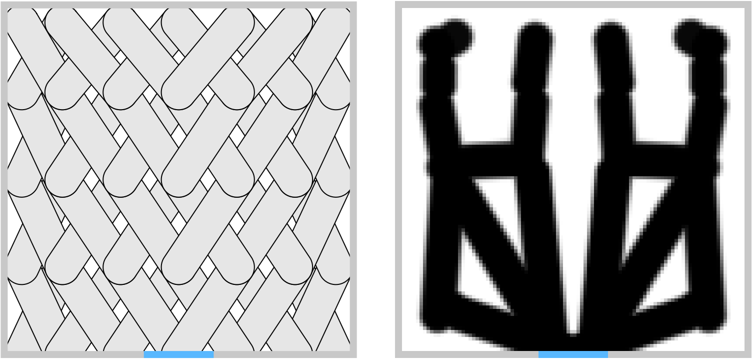


Figure 7.10: Explicitly constrained layout optimization with constraint toggle. Initial condition (left), optimized topology (right).

thermal compliance value of 2,193 WK, which is improved from the previous implementation. The optimizer was allowed to naturally adjust the position of the bars towards a local optima. At the 15th iteration, the intersection constraint was toggled on to enforce a feasible structure. The 15th iteration was selected since the structure started to settle into a reasonable form and floating elements were not yet present. It is important that no floating elements are present as the constraint will repel floating elements past a certain distance. Defining a more effective intersection penalty is left as a topic of future work.

7.5 Discussion

Using projected explicit geometric elements for fixed-mesh topology optimization presents challenges similar to those present when using the SIMP method. The optimization results have compiled into the following table to facilitate visual comparison.

These challenges become evident when performing heat spreader topology

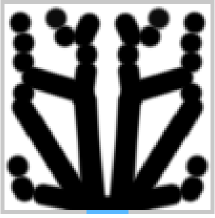
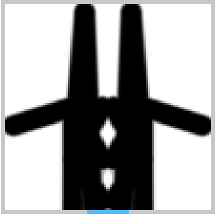
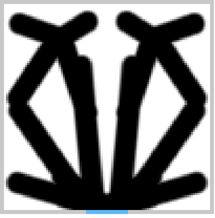
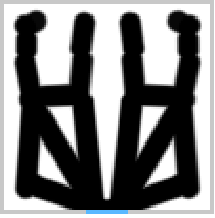
			
unconstrained	implicit	explicit	toggle

Table 7.1: Compiled projection results.

optimization. When using design variables to represent the density, or existence, of geometric elements, partially defined bars are present in the optimized structure. Using a large value to penalize the material density of the bars alleviates this issue, but presents checkerboard patterns. When using the SIMP method, filters can be used to prevent this occurrence and produce mesh independent designs. For the geometric projection method, two alternative strategies based on the use of layout constraints are presented. It was shown that using an implicit constraint produced poorly performing structures when compared to an explicit constraint approach. The use of the explicit constraint was also shown to be more flexible as it can be toggled ON at a later point in the optimization to produce feasible structures. Furthermore this design resembles the unconstrained optimization solution. Automating the toggle detection point and defining a more stable intersection constraint are left as topic of feature work.

7.6 Summary

In this chapter, an alternative optimization formulation was presented to overcome some challenges with the explicit geometric projection methods for topology optimization. This investigation was motivated by the comparative study and sequential fix for the GPM presented in Chapter 6. Using explicit intersection constraints allowed the optimizer sufficient freedom to find well performing structures and satisfy minimum feature size requirements. Using these differentiable projections on a fixed-finite element mesh opens opportunities for advances in applications with fixed topologies, such as packing and layout optimization [131].

Chapter 8

Maximum Approximation Functions

8.1 Introduction

In power electronic applications, assessing the maximum temperature of a power device is critical to ensure safe operation. Several of the design methods investigated in this dissertation take advantage of gradient-based algorithms to quickly converge to optimized solutions. The maximum function itself is not differentiable, and as such, can not be used with gradient-based algorithms. There are several smooth approximations for the maximum function that have been used for topology optimization [83, 132, 133]. Though these functions have been used to produce optimized structures, the implications of their use have not been thoroughly studied.

In this chapter an assessment of maximum approximation functions as they apply to topology optimization is conducted. Four maximum approximation functions are introduced in Sec. 8.2. Then, the tuning parameters of these functions and their respective derivatives are analyzed in Sec. 8.3. How the approximation functions scale with respect to the number of temperature values and the temperature distribution is investigated in Sec. 8.4. Some topologically optimized temperature distributions are presented and suggestions for improving maximum approximation accuracy are detailed in Sec. 8.5. A discussion of findings is presented in Sec. 8.6, and a summary of the chapter is presented in Sec. 8.7

8.2 Maximum approximation functions

In this section, four maximum approximation functions are under investigation as candidate replacements for the maximum function. The first approximation function used is the p-norm function. It is given by the following equation:

$$T_{\text{norm}} = ||\mathbf{T}||_p = \left(\sum_{i=1}^{n} |T_i|^p \right)^{1/p}. \tag{8.1}$$

The maximum approximation is taken as the p-root of the sum of nodal temperatures, T_i, raised to a power p. This function approximates the max function as $p \rightarrow \infty$. This approximation function has been used in heat transfer topology optimization for a similar problem detailed in Ref. [83].

The next approximation function is a smooth maximum function using exponentials:

$$T_{\text{s,e}} = \frac{\sum_{i=1}^{n} T_i \exp\left(\upsilon T_i\right)}{\sum_{i=1}^{n} \exp\left(\upsilon T_i\right)}, \tag{8.2}$$

where a scalar, υ, can be chosen to scale the smoothness and accuracy of the approximation. As $\upsilon \rightarrow \infty$, the maximum function is obtained.

The next approximation function is an alternative smooth maximum approximation using a power function that is investigated in Ref. [132]:

$$T_{\text{s,p}} = \frac{\sum_{i=1}^{n} T_i \sigma_i^T}{\sum_{i=1}^{n} \sigma_i^T}. \tag{8.3}$$

Here the parameter, σ, can be tuned to improve accuracy.

The final maximum approximation is based on a logrithmic exponential sum

(LSE) [133]:

$$T_{\text{LSE}} = \frac{1}{N} \ln \left(\sum_{i=1}^{n} \exp\left(NT_i\right) \right), \tag{8.4}$$

where the scalar parameter, N, can be used to tune the accuracy of the function. This approximation function has been used to predict the maximum stress in stress-constrained topology optimization [130].

To illustrate the behavioral differences between the four approximation functions, consider the following test problem:

$$T_{\max} = \max\left(T_1 \in [0,2], T_2 = 1\right), \tag{8.5}$$

where the maximum approximation function must determine the maximum value between a temperature, T_1 (varied from between 0 and 2), and a second temperature, T_2 (fixed at 1). The predicted maximum temperature approximations for all four functions are presented in Fig. 8.1. Note that the tuning parameters were selected to help visualize the behavioral differences.

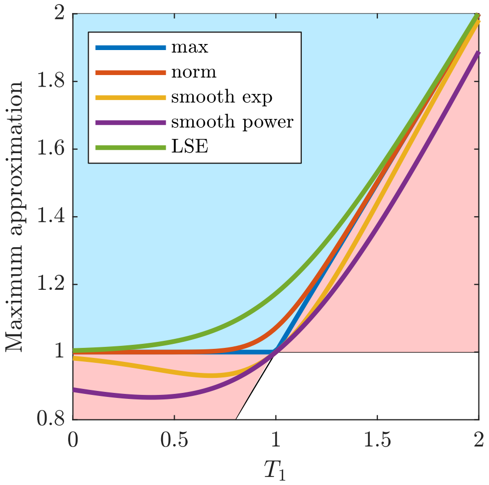


Figure 8.1: Maximum approximation functions. The blue region is an overapproximation, the red regions are under approximations.

From the figure, two classes of approximation functions can be observed. The norm and LSE maximum functions consistently overestimate the maximum value within the range of data. The two smooth maximum approximation functions consistently underestimate the maximum temperature. When $\upsilon = 0$ or $\sigma = 1$, Eqn. (8.2) and Eqn. (8.3) simply equal the mean temperature. Increasing the value of these parameters slowly transitions this mean function to the max function, hence always underestimating the maximum value. As such, these two functions are not appropriate to be used for constraints. However, all four of the maximum approximation functions can be used as an optimization objective.

8.3 Derivatives

In this section, the maximum approximation functions are analyzed in more detail as a function of their scaling parameters and derivatives. The same test case as defined in the previous section,

$$T_{\max} = \max\left(T_1 \in [0,2], T_2 = 1\right), \tag{8.6}$$

will be analyzed. To visualize the search direction in 1D, a parameter, S, is defined as the ratio of the partial derivative of the temperature with respect to each temperature value as shown below,

$$S = \frac{\frac{\partial T}{\partial T_2}}{\frac{\partial T}{\partial T_1}}. \tag{8.7}$$

The derivatives are normalized to 1 to allow for simple visual comparison of trends.

8.3.1 P-norm

First, consider the p-norm function. The derivative of the p-norm is defined by the following equation:

$$\frac{dT_{\text{norm}}}{d\mathbf{T}} = \frac{\mathbf{T} \circ |\mathbf{T}|^{p-2}}{||\mathbf{T}||_p^{p-1}}. \tag{8.8}$$

The predicted maximum values for the 2, 10, and 50 norm are presented in Fig. 8.2. As the order of the norm increases, the error in the approximation function tends towards zero. The derivative of the norm function is sufficiently smooth and transitions into the shape of the maximum step as p increases.

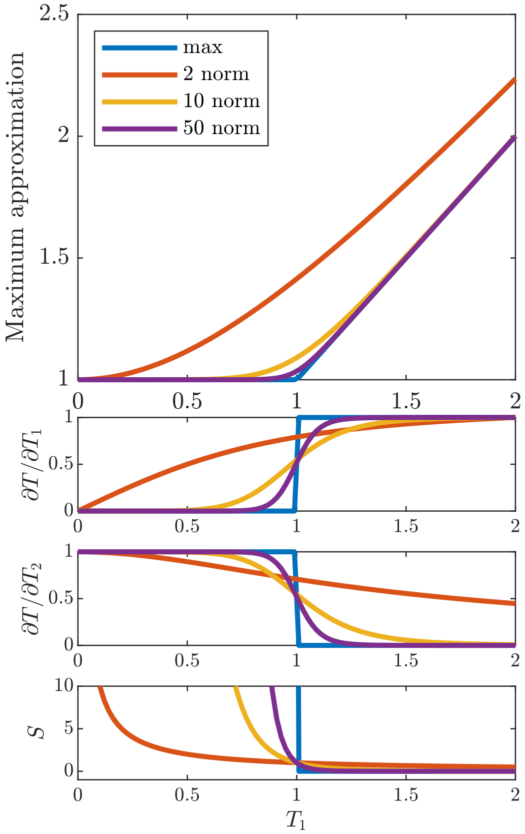


Figure 8.2: Scaling and derivatives for the norm maximum approximation function.

The search direction remains positive across the range of values which leads the optimization towards “lower” temperature solution.

8.3.2 Smooth Max (exponential form)

Next, consider a smooth maximum approximation of the maximum function using exponentials. The derivative of this function is given by the following equation:

$$\frac{dT_{\mathrm{s,e}}}{d\mathbf{T}} = \left(\sum_{i=1}^{n} e^{\upsilon T_i} \right)^{-1} \left[e^{\upsilon \mathbf{T}} \circ (\mathbf{1} + \upsilon \mathbf{T}) \right]. \tag{8.9}$$

The predicted maximum values for $\upsilon = 2$, 8, and 20 are presented in Fig. 8.3. As the temperature scalar parameter, υ, increases the error in approximation decreases. As mentioned in the previous section, the function always under predicts the true maximum. For the prescribed υ values, the derivative with respect to T_1 has an asymmetric shape at 1. Despite this non-ideal behavior, the search direction, S, looks similar to that of the p-norm.

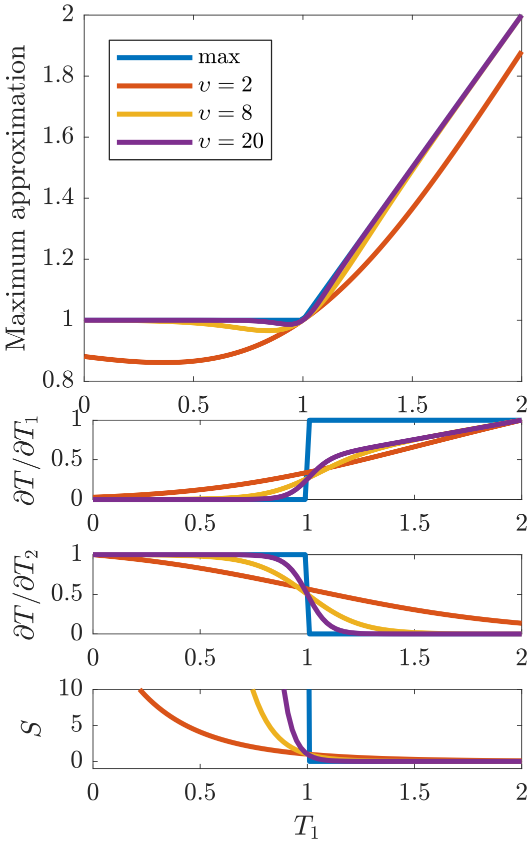


Figure 8.3: Scaling and derivatives for the exponential smooth maximum approximation function.

8.3.3 Smooth Max (power form)

The derivative of the smooth max in a power form as derived in Ref. [132] is presented in the following equation:

$$\frac{dT_{\text{s,p}}}{d\mathbf{T}} = \sigma^{\mathbf{T}} \circ \left(\sum_{i=1}^{n} \sigma^{T_i}\right)^{-1} [\mathbf{1} + \mathbf{T} ln(\sigma)] - \left(\frac{\sum_{i=1}^{n} T_i \sigma^{T_i}}{(\sum_{i=1}^{n} \sigma^{T_i})^2}\right) \left[ln(\sigma)\sigma^{\mathbf{T}}\right]. \quad (8.10)$$

This particular function is not as sensitive to the scalar parameter, σ, and as such, large values of 1×10^3, 1×10^5, and 1×10^7 are used for this parameter. The predicted maximum values for this function are presented in Fig. 8.4. As

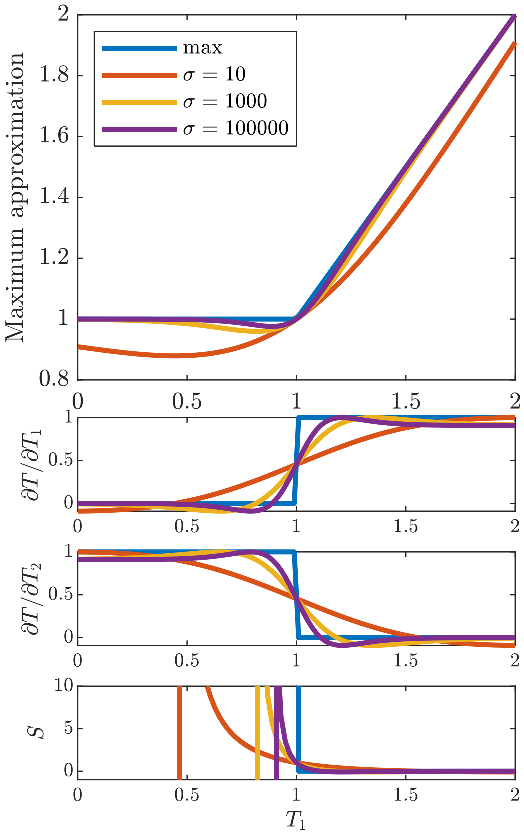


Figure 8.4: Scaling and derivatives for the power smooth maximum approximation function.

expected, the approximation function increases in accuracy as the parameter $\sigma \to \infty$. For this function, the derivatives switch sign in certain situations, as evident near 0.8 for $\frac{\partial T}{\partial T_1}$ and 1.2 for $\frac{\partial T}{\partial T_2}$. This sign change may cause the optimizer to converge to the incorrect local minimum since the derivative crosses the zero axis more than once. This is more evident in the plot for the search direction, S. Given that the approximation function has multiple roots and the derivative switches sign, it should not be used in optimization.

8.3.4 Log Sum Exponential Max

The final approximation function is the log-sum-exponential approximation. The derivative of this function is given by the following equation:

$$\frac{dT_{\text{LSE}}}{d\mathbf{T}} = \left(\sum_{i=1}^{n} e^{NT_i}\right)^{-1} \left[e^{N\mathbf{T}}\right]. \tag{8.11}$$

The temperate scalar parameter is set to values of $N = 2$, 8, and 20, and the predicted temperature values are presented in Fig. 8.5. As the parameter

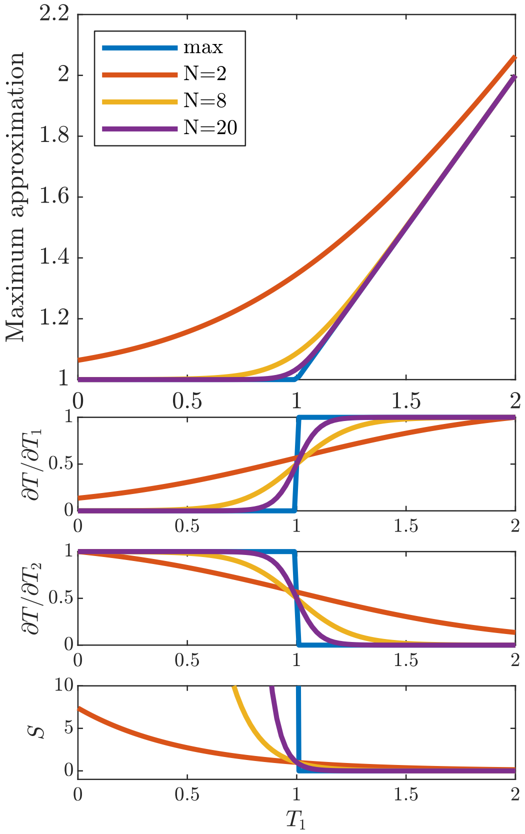


Figure 8.5: Scaling and derivatives for the log-sum-exponential maximum approximation function.

N increases, the error in approximation decreases and the derivatives for this function transition smoothly from a linear ramp to a step. The derivatives remain monotonic and behave similarly to the p-norm with larger values of p.

8.4 Prediction Error

Based on the analysis presented in the last section, the p-norm and log-sum-exponential maximum approximation functions are the most appropriate for optimization. To better understand the behavior of these functions, the accuracy as a function of mesh size (Sec. 8.4.1) and temperature distribution shape (Sec. 8.4.2) are presented.

8.4.1 Number of finite elements.

First, consider the effect of increasing the number of finite elements in a mesh. This will increase the total number of temperature measurements that are aggregated in the maximum approximation functions. Here the temperature is bounded between 0 and 1, and with linearly interpolated temperature values of an increasing discretization size. The accuracy of the norm and LSE functions for various set sizes is illustrated in Fig. 8.6.

From the plot, it can be deduced that the norm function loses accuracy more quickly as the total number of aggregated elements increases when compared to the LSE function when $p = N$. This can be attributed to the magnitude of the elements which are being aggregated. As p increases for the norm function, the temperature values are being scaled towards 0. As N increases, the temperature values entered to the exponential are greater than 1. In effect, the norm function for temperature values between 0 and 1 tries to minimize small temperatures to make the maximum temperature stand out, and the LSE function tries to increase the magnitude of the largest temperature such

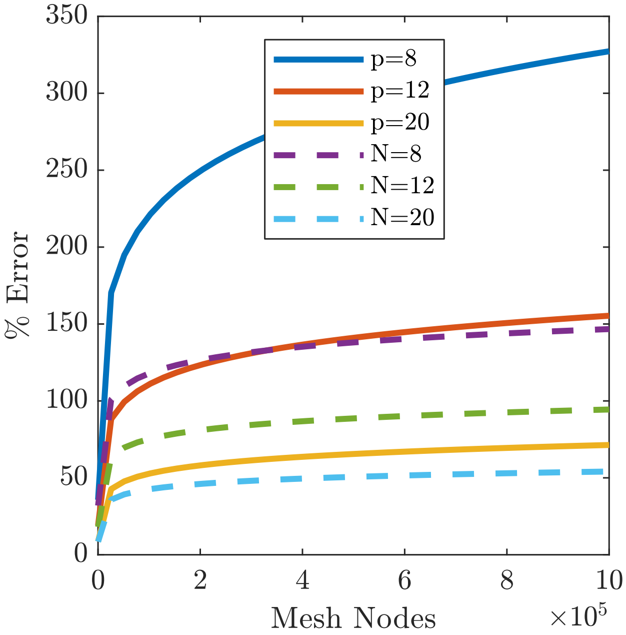


Figure 8.6: Norm and LSE maximum approximation error as a function of the total number of nodal temperature values.

that it stands out. For temperature values constrained between 0 and 1, the LSE is more effective if $p = N$, however, a norm value can be chosen to achieve comparable performance for a given N, for example: $(p = 12, N = 8)$. For these particular values, the norm function is more accurate for a mesh with less than 2×10^5 temperature nodes.

8.4.2 Distribution Shape

Given the fundamental difference between the algorithms for normalized temperature values, the accuracy of these functions will be sensitive to the distribution of temperature values. When optimizing a heat spreading structure, the ordered temperature distribution across a domain may not be linearly distributed. Consider scaling the temperature values by the following trigonometric function:

$$\mathbf{T}_d = \frac{1}{\beta}\tanh^{-1}\left(\mathbf{T}_0\left[\tanh(\beta\nu) + \tanh(\beta(1-\nu))\right] - tanh(\beta - \nu)\right) - \nu. \tag{8.12}$$

The parameter, β, controls the shape of the temperature distribution, and the parameter, ν, shifts the average value. A vector, $\mathbf{T}_0$, of values linearly interpolated between 0 and 1 is used to create the new temperature distribution vector, $\mathbf{T}_d$. When performing topology optimization with a fixed temperature boundary, the ordered set of temperature values can fit to this equation. The effects of changing individual parameters, β and ν, are presented in Fig. 8.7.

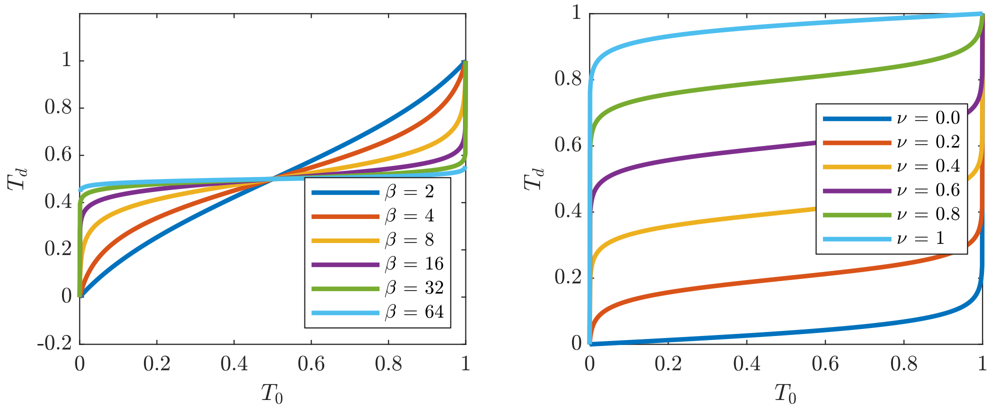


Figure 8.7: Effect of trigonometric function.

As the parameter β increases, the temperature values become clustered about the parameter, ν. For the parameter ν, the temperatures start mostly high for $\nu = 1$ and become mostly low for $\nu = 0$. The sensitivity of the norm maximum approximation function with respect to the distribution shape is presented using a heat map, illustrated in Fig. 8.8.

It is important to note that the function becomes undefined for certain combinations of β and ν; these combinations have been identified by the pink patches on the domain. The norm function more accurately predicts the max-

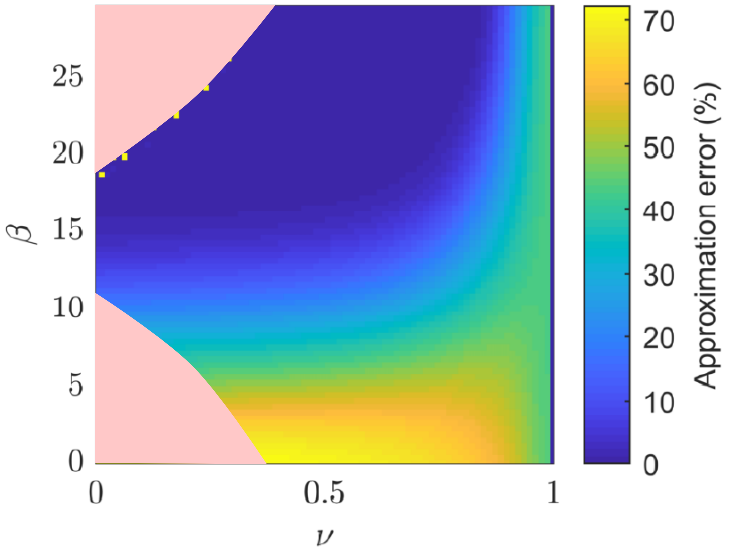


Figure 8.8: Approximation function sensitivity to temperature distribution.

imum temperature when there is a single large value in the set, $\nu \to 1$ or $\beta \to 30$. As the distribution becomes more uniform, the error levels off near 75%.

8.5 Topology optimized temperature distributions

A given heat spreading structure will provide a similar sorted temperature distribution to these discussed in Sec. 8.4.2. Consider the normalized sorted temperature distributions for a variety of temperature optimized solutions, shown in Fig. 8.9. The dashed black line presents a fitted curve with $\beta = 4$ and $\nu = 0.45$. This fitted curve can be used to understand the error in approximation prediction for a problem with a similar temperature profile. Consider the approximation error as a function of the number of mesh nodes for the fitted temperature distribution, illustrated in Fig. 8.10.

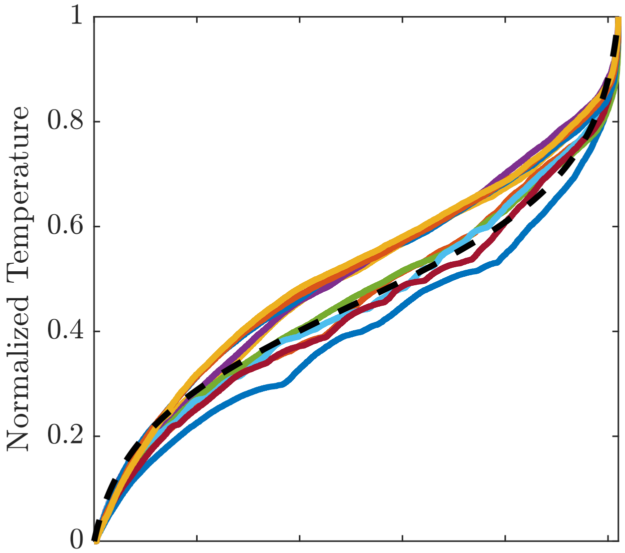


Figure 8.9: Approximation function sensitivity to temperature distribution.

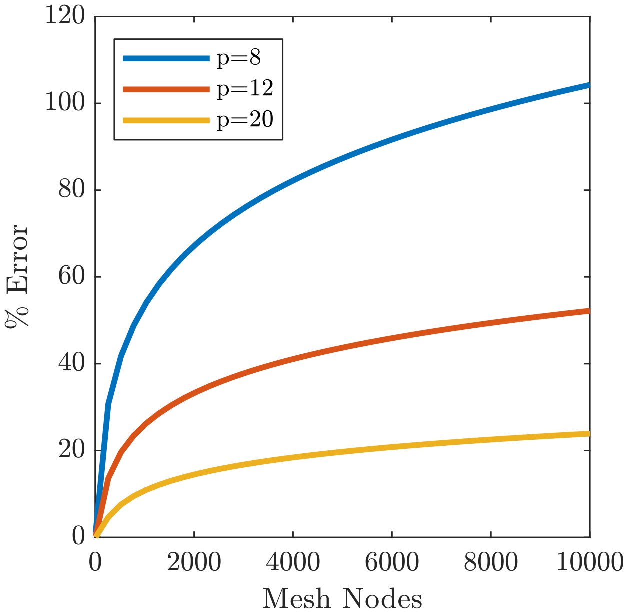


Figure 8.10: Approximation function sensitivity to temperature distribution.

It is preferable to choose the highest norm order to minimize the error in maximum approximation. However, large p values lead to numerical oscillations in the optimization. If performing temperature-constrained optimization, an alternative strategy to minimize constraint prediction error is to scale the constraint value by the predicted error. This will allow the optimizer to correctly assess when the constraint is active, while still using a smooth maximum approximation (low p).

8.6 Discussion

Differentiable maximum approximation functions will continue to be investigated for gradient-based topology optimization for their ability to predict parameters of interest (e.g., maximum temperature, maximum stress, etc.), while preserving differentiability. In this chapter, four approximation functions that have been utilized in the literature for topology optimization are analyzed in detail. The smooth maximum approximation functions have been shown to consistently underestimate the maximum value, and as such are not appropriate for use as a constraint. Furthermore, the derivative of the exponential form of the smooth max changes sign during optimization and may cause the optimizer to converge to an incorrect local optima. For this reason, this approximation function should not be used in optimization. The norm and LSE maximum approximations offer comparable performance. The norm function is more accurate when the temperature set is small, and the LSE function is more accurate when using a large set of temperature values. Either maximum approximation function can be used confidently for optimization as their derivatives are comparable.

An investigation was also conducted to determine the effect of temperature distribution on maximum prediction capability for the norm function. For topology optimization problems with a fixed temperature boundary, the optimized temperature can be fitted by a trigonometric function. This function can be used to predict the error of the p-norm temperature. For a maximum temperature constraint, this error can be used to scale the constraint value such that constraint activity is correct.

8.7 Summary

In this chapter, several maximum approximation functions were analyzed in the context of topology optimization. Smooth maximum approximation were observed to under-predict the true maximum value and as such are not suitable for a maximum temperature constraint approximation. Furthermore, the smooth maximum approximation in power form has poorly behaved derivatives, with multiple roots, as derived in Ref. [132] and should not be used for optimization. The norm function was found to be more accurate than a log-sum-exponent maximum approximation function for sets with a relatively small number of temperature values. A method to scale the maximum temperature constraint was proposed to predict constraint activity while maintaining a sufficiently smooth derivative.

Chapter 9

Conclusion

9.1 Summary

Topology optimization methods are a key enabling technology in driving forward performance improvements. In this dissertation, several structural topology optimization methods were enhanced for heat spreader design.

In Ch. 2, a framework was presented to classify general topology optimization problems. This general class of topologies can be represented by vertex labeled graphs and applies to several application domains (e.g., electrical circuits, hybrid-powertrains, structural components, chemical processes, etc). The framework unified language across these application domains to draw comparisons between design methods. Key distinctions here are emphasizing the role of the modeler, defining relative expense metrics, and clarifying the use of data in nested and simultaneous forms.

The following chapter, Ch. 3, focused on some practical heat conduction formulations for heat spreader topology optimization. This chapter included an investigation into objective functions and constraint combinations that capture power electronic requirements. Unique to this chapter is the use of a temperature-constrained minimization objective formulation, presented in Sec. 3.4.1, which normalizes values during optimization. Furthermore, this formulation was also able to capture a component selection task within the topology optimization framework.

In Ch. 4, a novel packaging scheme was presented for the FCML inverter topology. This packaging solution resulted in a 40% reduction in cooling system volume for the unfolding and inversion stages. A reduced-order convection model was used to topologically optimize a heat sink for this application. The optimization solution did not satisfy manufacturing constraints, and aspects of the optimized solution were used to design a modified heat sink. It was shown in simulation that the modified heat sink should have comparable performance to a baseline design. However, experimentation revealed a discrepancy with the simulation results and the baseline heat sink outperformed the modified design. This motivated an investigation into reduced-order design methods for forced-convection heat sink design.

A series of topology optimization studies were performed in Ch. 5 to generate a wide variety of heat spreader designs. These studies involved changes in problem formulation aspects such as the physics models, boundary conditions, objective functions, and design representations used to generate heat spreaders. The heat spreader designs were experimentally qualified on a simple test platform. Simple heat spreader designs were found to maintain the lowest device temperature during experimentation. The best design consisted of simple fins with fillets. This feature combination was unlikely, or unachievable, using the proposed design representations and motivated an investigation into structural design representations for topology optimization.

In Ch. 6, optimization using the SIMP, GPM, and SCA were analyzed. It was observed that the SIMP optimization produced the best performing design when optimizing a structure with partially defined boundaries. The GPM solution produced optimized topologies with long fin structures, however, failed to converge to a feasible design. The SCA was able to produce a wide variety of non-dendritic structures, however, failed to converge to the simplified projection of constant thickness bars. Of the three methods, the SCA method was the only one that actually tested new topologies between iterations. Both the SIMP method and GPM converged to a topology within a few iterations and spent the remaining majority of iterations performing size optimization.

The geometric projection method was further investigated in Ch. 7 to address challenges emergent for heat spreader topology optimization as discovered in the previous chapter. The GPM struggled to converge to a binary solution, and this was simply resolved by an increase in the SIMP penalization. Addressing this issue revealed the emergence of checkerboard patterns in the optimized design, which disqualified the manufacturability of the structure. To address the checkerboard patterns, two layout constraint formulations were proposed. The first considered an implicit constraint formulation as proposed by in Ref. [124], and the second considered an explicit constraint formulation enforced through optimization. It was observed that the explicitly-constrained layout allowed the optimizer to converge to better performing feasible designs.

In Ch. 8, four maximum approximation functions were analyzed in detail for use with topology optimization methods. The scaling of these functions with respect to tuning parameters and mesh size was analyzed. The derivatives of the smooth maximum approximations revealed strange behavior that is not suitable for optimization. The norm and LSE approximation functions behaved well, and their accuracy was evaluated with temperature distributions obtained using topology optimization. A constraint scaling strategy was then suggested to properly identify constraint activity while maintaining a smooth derivative.

9.2 Key Contributions

1. A framework to classify topology optimization methods was developed. This classification framework drew upon research from several isolated research application communities (electronic circuits, hybrid powertrains, structural design, etc) to identify common themes and unify language. This activity is a first step in bridging the gap between these research communities to identify similarities and the transition potential of numerical methods.

2. Practical heat conduction topology optimization formulations were presented for power electronics systems. This included a side-by-side comparison of optimized structures obtained using several objective functions from the literature, In addition to a study of some objective and constraint combinations to satisfy common power electronic requirements.

3. Two experimental investigations were presented to assess reduced-order convection models for topology optimization. These experiments were performed on test platforms using wide-bandgap power electronic devices. The experiments have shown that the reduced order methods do not accurately predict performance despite CFD validation. The test-platform developed is Ch. 5 is simple and allows for an inexpensive and easy way to reproduce these results.

4. An explicitly-enforced optimization layout constraint was proposed to address checkerboard patterns with the GPM that arise during structural design for heat transfer. This explicit constraint was shown to converge to better solutions when compared to optimizing an implicitly defined layout.

5. An in-depth analysis of maximum approximation functions for use in topology optimization was performed. Reducing or constraining the maximum temperature of a power electronic component is crucial for safe operation. This investigation revealed improper use of certain approximation functions and suggested a scaling technique to improve accuracy while maintaining smooth derivatives.

9.3 Future Work

- **Transitioning relaxation methods to other topology optimization applications** — Relaxation methods have been used successfully for structural topology optimization using analytical sensitivity calculations. Applications areas such as, hybrid powertrain or passive circuit filter design, favor meta-heuristic approaches to both select a topology and size system elements. There is potential for a large impact when applying relaxation methods to other topology optimization design problems.

- **Enhancing reduced-order convection models for topology optimization** — The reduced order convection model as outlined in Ref. [88] presents an intuitive parameterization of the design problem, however, the model simplifications do not capture enough of the physics to produce well performing structures. Enhancements such as those presented in Ref. [89] consider the effect of straight and corner segments, but do not capture additional fluid considerations. Accounting for the effect of the air gap on the convective heat transfer coefficient may provide sufficient improvement to produce structures that perform well in practice without solving an expensive simulation.

- **Alleviating checkerboarding tendency of GPM** — Using an implicit or explicit layout constrained set of geometric elements was shown to reliably produce feasible structures. However, the designs were found to under-perform those produced by other methods. Identifying an alternative adaptive-fix strategy for nearly interfaced elements, similar to the elliptic joint modification proposed in Ref. [96], may alleviate the checkerboard patterns without restricting design flexibility.

- **Extending topology optimization framework to include system-level aspects** — Topology optimization as a design tool is maturing for the design of single structural components. There is a lot of promise for incorporating topology optimization in more holistic optimization loops. Consider the component selection example presented in Sec. 3.4.1, the simultaneous layout and heat spreader design example [8], or this routing and packing optimization example in Ref. [131]. The integration of topology optimization methods with other design tools is still new and a world of opportunity remains.

Bibliography

[1] M. P. Bendsøe and N. Kikuchi, "Generating Optimal Topologies in Structural Design Using a Homogenization Method," *Comput. Methods. Appl. Mech. Eng.*, vol. 71, no. 2, pp. 197–224, 1988, doi: 10.1016/0045-7825(88)90086-2. doi: 10.1016/0045-7825(88)90086-2 (see p. 2)

[2] G. Allaire, F. Jouve, and A.-M. Toader, "Structural optimization using sensitivity analysis and a level-set method," *Journal of Computational Physics*, vol. 194, no. 1, pp. 363–393, Feb. 2004. doi: 10.1016/j.jcp.2003.09.032 (see pp. 2, 3, 115)

[3] B. Bell, J. Norato, and D. Tortorelli, "A geometry projection method for continuum-based topology optimization of structures," in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2012. doi: 10.2514/6.2012-5485 (see pp. 2, 3, 115, 138)

[4] T. Guo and J. T. Allison, "On the Use of Mathematical Programs with Complementarity Constraints in Combined Topological and Parametric Design of Biochemical Enzyme Networks," *Eng. Optim.*, vol. 49, no. 2, pp. 345–364, 2017, doi: 10.1080/0305215X.2016.1188091. doi: 10.1080/0305215X.2016.1188091 (see pp. 2, 3, 6, 25, 33, 116, 138)

[5] N. Aage, E. Andreassen, and B. S. Lazarov, "Topology Optimization Using PETSc: An Easy-to-Use, Fully Parallel, Open Source Topology Optimization Framework," *Struct. Multidiscip. Optim.*, vol. 51, no. 3, pp. 565–572, 2015, doi: 10.1007/s00158-014-1157-0. doi: 10.1007/s00158-014-1157-0 (see pp. 3, 6, 28, 29, 33, 34)

[6] M. P. Bendsøe, N. Kikuchi, Martin Philip Bendsoe, *et al.*, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197–224, 1988, ISSN: 00457825. doi: 10.1016/0045-7825(88)90086-2. url: http://www.sciencedirect.com/science/article/pii/0045782588900862 (see pp. 3, 37, 66, 103, 115)

[7] H.-T. C. Pedro and M. H. Kobayashi, "On a Cellular Division Method for Topology Optimization," *Int. J. Numer. Methods Eng.*, vol. 88, no. 11, pp. 1175–1197, 2011, doi: 10.1002/nme.3218. doi: 10.1002/nme.3218 (see pp. 3, 6, 23, 33, 115)

[8] D. J. Lohan, "Generative design algorithms in topology optimization of passive heat spreaders," Master's thesis, University of Illinois at Urbana-Champaign, 2016 (see pp. 3, 38, 104, 128, 130, 133, 136, 137, 175)

[9] G. I. N. Rozvany, “A Critical Review of Established Methods of Structural Topology Optimization,” *Struct. Multidiscip. Optim.*, vol. 37, no. 3, pp. 217–237, 2009, doi: 10.1007/s00158-007-0217-0. doi: `10.1007/s00158-007-0217-0` (see pp. 6, 23, 30)

[10] J. D. Deaton and R. V. Grandhi, “A Survey of Structural and Multidisciplinary Continuum Topology Optimization: Post 2000,” *Struct. Multidiscip. Optim.*, vol. 49, no. 1, pp. 1–38, 2014, doi: 10.1007/s00158-013-0956-z. doi: `10.1007/s00158-013-0956-z` (see pp. 6, 7, 14, 23, 25, 29, 30)

[11] G. Rozvany, “The SIMP Method in Topology Optimization – Theoretical Background, Advantages and New Applications,” in *Symposium on Multidisciplinary Analysis and Optimization*, doi: 10.2514/6.2000-4738, 2000. doi: `10.2514/6.2000-4738` (see pp. 6, 23, 29, 34)

[12] K. Deb and S. Gulati, “Design of Truss-Structures for Minimum Weight Using Genetic Algorithms,” *Finite Elem. Anal. Des.*, vol. 37, no. 5, pp. 447–465, 2001, doi: 10.1016/s0168-874x(00)00057-3. doi: `10.1016/s0168-874x(00)00057-3` (see pp. 6, 33, 34)

[13] D. Bortis, D. Neumayr, and J. W. Kolar, “$\eta\rho$-Pareto Optimization and Comparative Evaluation of Inverter Concepts Considered for the GOOGLE Little Box Challenge,” in *IEEE Workshop on Control and Modeling for Power Electronics*, doi: 10.1109/COMPEL.2016.7556767, 2016. doi: `10.1109/COMPEL.2016.7556767` (see pp. 6, 33)

[14] T. Guo, D. R. Herber, and J. T. Allison, “Reducing Evaluation Cost for Circuit Synthesis Using Active Learning,” in *ASME International Design Engineering Technical Conferences*, doi: 10.1115/DETC2018-85654, 2018. doi: `10.1115/DETC2018-85654` (see pp. 6, 27, 33)

[15] J. R. Koza, F. H. Bennett, D. Andre, *et al.*, “Synthesis of Topology and Sizing of Analog Electrical Circuits by Means of Genetic Programming,” *Comput. Methods. Appl. Mech. Eng.*, vol. 186, no. 2–4, pp. 459–482, 2000, doi: 10.1016/S0045-7825(99)00397-7. doi: `10.1016/S0045-7825(99)00397-7` (see pp. 6, 33)

[16] A. Das and R. Vemuri, “Topology Synthesis of Analog Circuits Based on Adaptively Generated Building Blocks,” in *ACM/IEEE Design Automation Conference*, 2008, pp. 44–49 (see pp. 6, 33)

[17] C. Ferent and A. Doboli, “Novel Circuit Topology Synthesis Method Using Circuit Feature Mining and Symbolic Comparison,” in *Design, Automation & Test in Europe Conference & Exhibition*, doi: 10.7873/DATE.2014.030, 2014. doi: `10.7873/DATE2014.030` (see p. 6)

[18] J. B. Grimbleby, “Automatic Analogue Circuit Synthesis Using Genetic Algorithms,” *IEE Proceedings - Circuits, Devices and Systems*, vol. 147, no. 6, pp. 319–323, 2000, doi: 10.1049/ip-cds:20000770. doi: `10.1049/ip-cds:20000770` (see pp. 6, 31)

[19] A. Das and R. Vemuri, “An Automated Passive Analog Circuit Synthesis Framework using Genetic Algorithms,” in *IEEE Computer Society Annual Symposium on VLSI*, doi: 10.1109/ISVLSI.2007.22, 2007 (see pp. 6, 34)

[20] T. Guo, D. R. Herber, and J. T. Allison, "Circuit Synthesis Using Generative Adversarial Networks (GANs)," in *AIAA SciTech Forum*, doi: 10.2514/6.2019-2350, 2019. doi: 10.2514/6.2019-2350 (see pp. 6, 28)

[21] E. Silvas, T. Hofman, N. Murgovski, *et al.*, "Review of Optimization Strategies for System-Level Design in Hybrid Electric Vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 57–70, 2017, doi: 10.1109/TVT.2016.2547897. doi: 10.1109/TVT.2016.2547897 (see pp. 6, 7)

[22] X. Zhou, D. Qin, and J. Hu, "Multi-Objective Optimization Design and Performance Evaluation for Plug-In Hybrid Electric Vehicle Powertrains," *Appl. Energy*, vol. 208, no. 174, pp. 1608–1625, 2017, doi: 10.1016/j.apenergy.2017.08.201. doi: 10.1016/j.apenergy.2017.08.201 (see pp. 6, 33)

[23] A. H. Ing and J. McPhee, "Automated Topology Optimisation of Hybrid Electric Vehicle Powertrains," *Int. J. Electr. Hybrid Veh.*, vol. 7, no. 4, pp. 342–361, 2015, doi: 10.1504/IJEHV.2015.074671. doi: 10.1504/IJEHV.2015.074671 (see pp. 6, 33, 34)

[24] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Optimal Dual-Mode Hybrid Electric Vehicle Powertrain Architecture Design for a Variety of Loading Scenarios," in *ASME International Design Engineering Technical Conferences*, doi: 10.1115/DETC2014-34897, 2014. doi: 10.1115/DETC2014-34897 (see pp. 6, 33, 34)

[25] E. Silvas, T. Hofman, A. Serebrenik, *et al.*, "Functional and Cost-Based Automatic Generator for Hybrid Vehicles Topologies," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 4, pp. 1561–1572, 2015, doi: 10.1109/TMECH.2015.2405473. doi: 10.1109/TMECH.2015.2405473 (see pp. 6, 33)

[26] A. E. Bayrak, Y. Ren, and P. Y. Papalambros, "Topology Generation for Hybrid Electric Vehicle Architecture Design," *ASME J. Mech. Des.*, vol. 138, no. 8, p. 081 401, 2016, doi: 10.1115/1.4033656. doi: 10.1115/1.4033656 (see pp. 6, 24)

[27] D. B. Doman, "Fuel Flow Topology and Control for Extending Aircraft Thermal Endurance," *J. Thermophys. Heat Transfer*, vol. 32, no. 1, pp. 35–50, 2017, doi: 10.2514/1.t5142. doi: 10.2514/1.t5142 (see pp. 6, 33)

[28] S. R. T. Peddada, D. R. Herber, H. Pangborn, *et al.*, "Optimal Flow Control and Single Split Architecture Exploration For Fluid-Based Thermal Management," *ASME J. Mech. Des. (to appear)*, (see pp. 6, 33)

[29] M. C. Smith, "Synthesis of Mechanical Networks: The Inerter," *IEEE Trans. Automat. Contr.*, vol. 47, no. 10, pp. 1648–1662, 2002, doi: 10.1109/TAC.2002.803532. doi: 10.1109/TAC.2002.803532 (see p. 6)

[30] D. R. Herber and J. T. Allison, "A Problem Class With Combined Architecture, Plant, and Control Design Applied to Vehicle Suspensions," *ASME J. Mech. Des. (to appear)*, (see pp. 6, 28, 33)

[31] M. Z. Q. Chen, C. Papageorgiou, F. Scheibe, *et al.*, "The Missing Mechanical Circuit Element," *IEEE Circuits Syst. Mag.*, vol. 9, no. 1, pp. 10–26, 2009, doi: 10.1109/MCAS.2008.931738. doi: 10.1109/MCAS.2008.931738 (see pp. 6, 21, 33)

[32] O. Chocron and P. Bidaud, "Genetic Design of 3D Modular Manipulators," in *Proceedings of International Conference on Robotics and Automation*, doi: 10.1109/ROBOT.1997.620042, 1997. doi: `10.1109/ROBOT.1997.620042` (see pp. 6, 33)

[33] J. Hiller and H. Lipson, "Automatic Design and Manufacture of Soft Robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 457–466, 2012, doi: 10.1109/TRO.2011.2172702. doi: `10.1109/TRO.2011.2172702` (see pp. 6, 33)

[34] W. Ma, A. Trusina, H. El-Samad, *et al.*, "Defining Network Topologies that Can Achieve Biochemical Adaptation," *Cell*, vol. 138, no. 4, pp. 760–773, 2009, doi: 10.1016/j.cell.2009.06.013. doi: `10.1016/j.cell.2009.06.013` (see pp. 6, 23, 33)

[35] M. I. Campbell, J. Cagan, and K. Kotovsky, "A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic Environment," *Res. Eng. Des.*, vol. 11, no. 3, pp. 172–192, 1999, doi: 10.1007/s001630050013. doi: `10.1007/s001630050013` (see p. 6)

[36] D. Subramanian and C. S. Wang, "Kinematic Synthesis with Configuration Spaces," *Res. Eng. Des.*, vol. 7, no. 3, pp. 193–213, 1995, doi: 10.1007/BF01638099. doi: `10.1007/BF01638099` (see p. 6)

[37] A. Chakrabarti, "Improving Efficiency of Procedures for Compositional Synthesis by Using Bidirectional Search," *Artif. Intell. Eng. Des. Anal. Manuf.*, vol. 15, no. 1, pp. 67–80, 2001 (see pp. 6, 22)

[38] J. Schaefer and S. Rudolph, "Satellite Design by Design Grammars," *Aerosp. Sci. Technol.*, vol. 9, no. 1, pp. 81–91, 2005, doi: 10.1016/j.ast.2004.08.003. doi: `10.1016/j.ast.2004.08.003` (see p. 6)

[39] C. Münzer, B. Helms, and K. Shea, "Automatically Transforming Object-Oriented Graph-Based Representations Into Boolean Satisfiability Problems for Computational Design Synthesis," *ASME J. Mech. Des.*, vol. 135, no. 10, p. 101 001, 2013, doi: 10.1115/1.4024850. doi: `10.1115/1.4024850` (see pp. 6, 18, 23, 24)

[40] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg, *Systematic Methods of Chemical Process Design*. Prentice Hall, 1997 (see p. 6)

[41] C. Pornet, C. Gologan, P. C. Vratny, *et al.*, "Methodology for Sizing and Performance Assessment of Hybrid Energy Aircraft," *J. Aircr.*, vol. 52, no. 1, pp. 341–352, 2015, doi: 10.2514/1.C032716. doi: `10.2514/1.C032716` (see pp. 6, 33)

[42] H. Qi, Y. Fu, X. Qi, *et al.*, "Architecture Optimization of More Electric Aircraft Actuation System," *Chin. J. Aeronaut.*, vol. 24, no. 4, pp. 506–513, 2011, doi: 10.1016/S1000-9361(11)60058-7. doi: `10.1016/S1000-9361(11)60058-7` (see pp. 6, 33)

[43] A. Chakrabarti, K. Shea, R. B. Stone, *et al.*, "Computer-Based Design Synthesis Research: An Overview," *ASME J. Comput. Inf. Sci. Eng.*, vol. 11, no. 2, p. 021 003, 2011, doi: 10.1115/1.3593409. doi: `10.1115/1.3593409` (see pp. 7, 23, 24)

[44] E. Crawley, B. Cameron, and D. Selva, *System Architecture*. Pearson, 2015, doi: 10.1007/978-1-4020-4399-4. doi: `10.1007/978-1-4020-4399-4` (see pp. 7, 15, 23, 24, 30)

[45] J. Cagan, M. I. Campbell, S. Finger, *et al.*, “A Framework for Computational Design Synthesis: Model and Applications,” *J. Comput. Inf. Sci. Eng.*, vol. 5, no. 3, pp. 171–181, 2005, doi: 10.1115/1.2013289. doi: 10.1115/1.2013289 (see pp. 7, 15, 23, 24, 30)

[46] M. I. Campbell, “A Generic Scheme for Graph Topology Optimization,” in *World Congresses of Structural and Multidisciplinary Optimization*, 2005 (see pp. 7, 14, 23, 25, 29, 30)

[47] M. I. Campbell and R. Rai, “A Generalization of Computational Synthesis Methods in Engineering Design,” AAAI, Tech. Rep. SS-03-02, 2003 (see pp. 7, 15, 18, 30)

[48] B. Korte and J. Vygen, *Combinatorial Optimization.* Springer, 2001, doi: 10.1007/3-540-29297-7. doi: 10.1007/3-540-29297-7 (see pp. 8, 12)

[49] S.-M. Hsieh, C.-C. Hsu, and L.-F. Hsu, “Computational Science and Its Applications - ICCSA 2006,” in, M. L. Gavrilova and et al., Eds. Springer, 2006, ch. Efficient Method to Perform Isomorphism Testing of Labeled Graphs, pp. 422–431, doi: 10.1007/11751649_46. doi: 10.1007/11751649_46 (see pp. 9, 19)

[50] R. T. Marler and J. S. Arora, “Survey of Multi-Objective Optimization Methods for Engineering,” *Struct. Multidiscip. Optim.*, vol. 26, no. 6, pp. 369–395, 2004, doi: 10.1007/s00158-003-0368-6. doi: 10.1007/s00158-003-0368-6 (see p. 15)

[51] A. M. Ross, D. E. Hastings, J. M. Warmkessel, *et al.*, “Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design,” *J. Spacecr. Rockets*, vol. 41, no. 1, pp. 20–28, 2004, doi: 10.2514/1.9204. doi: 10.2514/1.9204 (see p. 15)

[52] M. Farina and P. Amato, “A Fuzzy Definition of “Optimality” for Many-Criteria Optimization Problems,” *IEEE Trans. Syst. Man. Cybern. A Syst. Hum.*, vol. 34, no. 3, pp. 315–326, 2004, doi: 10.1109/TSMCA.2004.824873. doi: 10.1109/TSMCA.2004.824873 (see p. 15)

[53] D. F. Wyatt, D. C. Wynn, J. P. Jarrett, *et al.*, “Supporting Product Architecture Design Using Computational Design Synthesis with Network Structure Constraints,” *Res. Eng. Des.*, vol. 23, no. 1, pp. 17–52, 2011, doi: 10.1007/s00163-011-0112-y. doi: 10.1007/s00163-011-0112-y (see p. 18)

[54] D. R. Herber, T. Guo, and J. T. Allison, “Enumeration of Architectures with Perfect Matchings,” *ASME J. Mech. Des.*, vol. 139, no. 5, p. 051 403, 2017, doi: 10.1115/1.4036132. doi: 10.1115/1.4036132 (see pp. 18, 19, 23)

[55] M. Meissner, O. Mitea, L. Luy, *et al.*, “Fast Isomorphism Testing for a Graph-Based Analog Circuit Synthesis Framework,” in *Design, Automation & Test in Europe Conference & Exhibition*, doi: 10.1109/DATE.2012.6176570, 2012. doi: 10.1109/DATE.2012.6176570 (see p. 19)

[56] D. R. Herber, “Advances in Combined Architecture, Plant, and Control Design,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2017 (see pp. 21, 24, 25, 29, 34)

[57] D. F. Wyatt, D. C. Wynn, and P. J. Clarkson, “A Scheme for Numerical Representation of Graph Structures in Engineering Design,” *ASME J. Mech. Des.*, vol. 136, no. 1, p. 011 010, 2014, doi: 10.1115/1.4025961. doi: 10.1115/1.4025961 (see p. 23)

[58] D. R. Herber and J. T. Allison, "Enhancements to the Perfect Matching-Based Tree Algorithm for Generating Architectures," University of Illinois at Urbana-Champaign, Tech. Rep. UIUC-ESDL-2017-02, 2017. url: https://www.ideals.illinois.edu/handle/2142/98990 (see p. 23)

[59] A. Devert, N. Bredeche, and M. Schoenauer, "Robustness and the Halting Problem for Multicellular Artificial Ontogeny," *IEEE Trans. Evol. Comput.*, vol. 15, no. 3, pp. 387–404, 2011, doi: 10.1109/TEVC.2011.2125969. doi: 10.1109/TEVC.2011.2125969 (see p. 23)

[60] A. Hooshmand, M. I. Campbell, and K. Shea, "Steps in Transforming Shapes Generated With Generative Design Into Simulation Models," in *ASME International Design Engineering Technical Conferences*, doi: 10.1115/DETC2012-71056, 2012. doi: 10.1115/DETC2012-71056 (see p. 24)

[61] Z. Wu, M. I. Campbell, and B. R. Fernández, "Bond Graph Based Automated Modeling for Computer-Aided Design of Dynamic Systems," *ASME J. Mech. Des.*, vol. 130, no. 4, p. 041 102, 2008, doi: 10.1115/1.2885180. doi: 10.1115/1.2885180 (see p. 24)

[62] The MathWorks, *Simulink*, url: https://www.mathworks.com/products/simulink.html. url: https://www.mathworks.com/products/simulink.html (see p. 24)

[63] W. Borutzky, *Bond Graph Methodology.* Springer, 2010, doi: 10.1007/978-1-84882-882-7. doi: 10.1007/978-1-84882-882-7 (see p. 24)

[64] C.-W. Ho, A. Ruehli, and P. Brennan, "The Modified Nodal Approach to Network Analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, 1975, doi: 10.1109/tcs.1975.1084079. doi: 10.1109/tcs.1975.1084079 (see pp. 24, 25)

[65] H. C. Pangborn, J. P. Koeln, M. A. Williams, *et al.*, "Experimental Validation of Graph-Based Hierarchical Control for Thermal Management," *ASME J. Dyn. Sys. Meas. Control.*, vol. 140, no. 10, p. 101 016, 2018, doi: 10.1115/1.4040211. doi: 10.1115/1.4040211 (see pp. 24, 25)

[66] M. G. Kendall, "A New Measure of Rank Correlation," *Biometrika*, vol. 30, no. 1, pp. 81–93, 1938 (see p. 27)

[67] N. V. Queipo, R. T. Haftka, W. Shyy, *et al.*, "Surrogate-Based Analysis and Optimization," *Prog. Aerosp. Sci.*, vol. 41, no. 1, pp. 1–28, 2005, doi: 10.1016/j.paerosci.2005.02.001. doi: 10.1016/j.paerosci.2005.02.001 (see p. 28)

[68] M. G. Fernández-Godino, C. Park, N.-H. Kim, *et al.*, *Review of Multi-Fidelity Models*, arXiv Preprint, arXiv:1609.07196 (see p. 28)

[69] Y. Jin, "A Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, 2005, doi: 10.1007/s00500-003-0328-5. doi: 10.1007/s00500-003-0328-5 (see p. 28)

[70] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design.* Cambridge University Press, 2018, doi: 10.1017/9781316451038. doi: 10.1017/9781316451038 (see p. 28)

[71] W. Hare, J. Nutini, and S. Tesfamariam, “A Survey of Non-Gradient Optimization Methods in Structural Engineering,” *Adv. Eng. Software*, vol. 59, pp. 19–28, 2013, doi: 10.1016/j.advengsoft.2013.03.001. doi: 10.1016/j.advengsoft.2013.03.001 (see p. 29)

[72] C. B. Browne, E. Powley, D. Whitehouse, *et al.*, “A Survey of Monte Carlo Tree Search Methods,” *IEEE Trans. Comput. Intell. AI in Games*, vol. 4, no. 1, pp. 1–43, 2012, doi: 10.1109/TCIAIG.2012.2186810. doi: 10.1109/TCIAIG.2012.2186810 (see p. 29)

[73] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989 (see pp. 29–31)

[74] A. K. Kar, “Bio Inspired Computing – A Review of Algorithms and Scope of Applications,” *Expert Syst. Appl.*, vol. 59, pp. 20–32, 2016, doi: 10.1016/j.eswa.2016.04.018. doi: 10.1016/j.eswa.2016.04.018 (see pp. 29, 30)

[75] A. Banks, J. Vincent, and C. Anyakoha, “A Review of Particle Swarm Optimization. Part I: Background and Development,” *Nat. Comput.*, vol. 6, no. 4, pp. 467–484, 2007, doi: 10.1007/s11047-007-9049-5. doi: 10.1007/s11047-007-9049-5 (see pp. 29, 30)

[76] B. Suman and P. Kumar, “A Survey of Simulated Annealing as a Tool for Single and Multiobjective Optimization,” *J. Oper. Res. Soc.*, vol. 57, no. 10, pp. 1143–1160, 2006, doi: 10.1057/palgrave.jors.2602068. doi: 10.1057/palgrave.jors.2602068 (see pp. 29, 30)

[77] F. Glover and M. Laguna, “Handbook of Combinatorial Optimization,” in, D.-Z. Du and P. M. Pardalos, Eds. Springer, 1998, ch. Tabu Search, pp. 2093–2229, doi: 10.1007/978-1-4613-0303-9_33. doi: 10.1007/978-1-4613-0303-9_33 (see pp. 29, 30)

[78] R. Martí, J. A. Lozano, A. Mendiburu, *et al.*, “Handbook of Metaheuristics,” in, R. Martí, P. Panos, and M. G. C. Resende, Eds. Springer, 2016, ch. Multi-Start Methods, pp. 1–21, doi: 10.1007/978-3-319-07153-4_1-1. doi: 10.1007/978-3-319-07153-4_1-1 (see p. 30)

[79] A. Khetan, D. J. Lohan, and J. T. Allison, “Managing Variable-Dimension Structural Optimization Problems Using Generative Algorithms,” *Struct. Multidiscip. Optim.*, vol. 50, no. 4, pp. 695–715, 2015, doi: 10.1007/s00158-015-1262-8. doi: 10.1007/s00158-015-1262-8 (see p. 30)

[80] J. Zhang, Z. Zhan, Y. Lin, *et al.*, “Evolutionary Computation Meets Machine Learning: A Survey,” *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, 2011, doi: 10.1109/MCI.2011.942584. doi: 10.1109/MCI.2011.942584 (see p. 32)

[81] A Gersborg-Hansen, M. P. Bendsoe, and O Sigmund, “Topology Optimization of Heat Conduction Problems Using the Finite Volume Method,” *Structural and Multidisciplinary Optimization*, vol. 31, pp. 251–259, Mar. 2006. doi: 10.1007/s00158-005-0584-3 (see pp. 37, 88, 116, 119, 135)

[82] F. H. Burger, J. Dirker, and J. P. Meyer, “Three-dimensional conductive heat transfer topology optimization in a cubic domain for the volume-to-surface problem,” *International Journal of Heat and Mass Transfer*, vol. 67, pp. 214–224, Dec. 2013. doi: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2013.08.015 (see p. 37)

[83] S. Yan, F. Wang, and O. Sigmund, “On the non-optimality of tree structures for heat conduction,” *International Journal of Heat and Mass Transfer*, vol. 122, pp. 660–680, 2018, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2018.01.114. arXiv: 1010.1724. url: https://doi.org/10.1016/j.ijheatmasstransfer.2018.01.114 (see pp. 38, 116, 135, 154, 155)

[84] G. Marck, M. Nemer, J.-L. Harion, *et al.*, “Topology optimization using the SIMP method for multiobjective conductive problems,” *Numerical Heat Transfer, Part B: Fundamentals*, vol. 61, no. 6, pp. 439–470, 2012, ISSN: 1040-7790. doi: 10.1080/10407790.2012.687979. url: http://www.tandfonline.com/doi/abs/10.1080/10407790.2012.687979 (see p. 38)

[85] J. Dirker and J. Meyer, “Topology optimization for an internal heat-conduction cooling scheme in a square domain for high heat flux applications,” *Journal of Heat Transfer*, vol. 135, no. 11, p. 111 010, Sep. 2013. doi: 10.1115/1.4024615 (see pp. 38, 44, 116)

[86] C. Zhuang and Z. Xiong, “Temperature-Constrained Topology Optimization of Transient Heat Conduction Problems,” *Numerical Heat Transfer, Part B: Fundamentals*, vol. 68, no. 4, pp. 366–385, 2015, ISSN: 1040-7790. doi: 10.1080/10407790.2015.1033306. url: http://www.tandfonline.com/doi/full/10.1080/10407790.2015.1033306 (see p. 39)

[87] T. E. Bruns, “Topology optimization of convection-dominated, steady-state heat transfer problems,” *International Journal of Heat and Mass Transfer*, vol. 50, no. 15-16, pp. 2859–2873, 2007, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2007.01.039 (see pp. 39, 66, 88)

[88] A. Iga, S. Nishiwaki, K. Izui, *et al.*, “Topology optimization for thermal conductors considering design-dependent effects, including heat conduction and convection,” *International Journal of Heat and Mass Transfer*, vol. 52, no. 11-12, pp. 2721–2732, 2009, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2008.12.013. url: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2008.12.013 (see pp. 39, 66, 73, 76, 86, 88, 101, 174)

[89] *Industrial application of topology optimization for combined conductive and convective heat transfer problems*, vol. 54, 4, Springer Science and Business Media LLC, Apr. 2016, pp. 1045–1060. doi: 10.1007/s00158-016-1433-2 (see pp. 39, 66, 88, 174)

[90] E. M. Dede, S. N. Joshi, and F. Zhou, “Topology Optimization, Additive Layer Manufacturing, and Experimental Testing of an Air-Cooled Heat Sink,” *Journal of Mechanical Design*, vol. 137, no. 11, p. 111 403, 2015, ISSN: 1050-0472. doi: 10.1115/1.4030989. url: http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=2389879http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4030989 (see pp. 39, 66, 89)

[91] J. H. Haertel and G. F. Nellis, “A fully developed flow thermofluid model for topology optimization of 3d-printed air-cooled heat exchangers,” *Applied Thermal Engineering*, vol. 119, pp. 10–24, 2017, ISSN: 13594311. doi: 10.1016/j.applthermaleng.2017.03.030. url: http://dx.doi.org/10.1016/j.applthermaleng.2017.03.030 (see pp. 39, 67, 87)

[92] J. Alexandersen, O. Sigmund, and N. Aage, “Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection,” *International Journal of Heat and Mass Transfer*, vol. 100, pp. 876–891, 2016, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2016.05.013. arXiv: 1508.04596. url: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2016.05.013 (see p. 39)

[93] T. Dbouk, “A review about the engineering design of optimal heat transfer systems using topology optimization,” *Applied Thermal Engineering*, vol. 112, pp. 841–854, Feb. 2017, ISSN: 13594311. doi: 10.1016/j.applthermaleng.2016.10.134. url: http://dx.doi.org/10.1016/j.applthermaleng.2016.10.134https://linkinghub.elsevier.com/retrieve/pii/S135943111632645X (see pp. 39, 63, 68, 88)

[94] E. M. Dede, Y. Liu, S. N. Joshi, *et al.*, “Optimal design of three-dimensional heat flow structures for power electronics applications,” *Journal of Thermal Science and Engineering Applications*, vol. 11, no. 2, p. 021 011, 2018. doi: 10.1115/1.4041440 (see pp. 40, 54, 88)

[95] D. Christen, M. Stojadinovic, and J. Biela, “Energy Efficient Heat Sink Design: Natural Versus Forced Convection Cooling,” *IEEE 17th Workshop on Control and Modeling for Power Electronics (COMPEL), 2016*, vol. 32, no. 11, 2016, ISSN: 08858993. doi: 10.1109/COMPEL.2016.7556668 (see p. 40)

[96] S. H. Yu, K. S. Lee, and S. J. Yook, “Optimum design of a radial heat sink under natural convection,” *International Journal of Heat and Mass Transfer*, vol. 54, no. 11-12, pp. 2499–2505, 2011, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2011.02.012. url: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2011.02.012 (see pp. 40, 175)

[97] S. Ramphueiphad and S. Bureerat, “Synthesis of multiple cross-section pin fin heat sinks using multiobjective evolutionary algorithms,” *International Journal of Heat and Mass Transfer*, vol. 118, pp. 462–470, 2018, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2017.11.016. url: https://doi.org/10.1016/j.ijheatmasstransfer.2017.11.016 (see p. 40)

[98] K. Svanberg and H. Svärd, “Density filters for topology optimization based on the Pythagorean means,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 5, pp. 859–875, 2013, ISSN: 1615147X. doi: 10.1007/s00158-013-0938-1 (see p. 42)

[99] K. Svanberg, “The Method of Moving Asymptotes—a New Method for Structural Optimization,” *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pp. 359–373, Feb. 1987 (see p. 42)

[100] A. E. Patterson, Y. H. Lee, and J. T. Allison, “Overview of the Development and Enforcement of Process-Driven Manufacturability Constraints in Product Design,” in *IDETC 2019*, Los Angeles, USA, Aug. 2019 (see p. 46)

[101] D. J. Lohan and J. T. Allison, "Temperature constraint formulations for heat conduction topology optimization," in *12th World Congress on Structural and Multidisciplinary Optimization*, Braunschweig, Germany, Jun. 2017 (see pp. 52, 131)

[102] C. Le, J. Norato, T. Bruns, *et al.*, "Stress-based topology optimization for continua," *Structural and Multidisciplinary Optimization*, vol. 41, no. 4, pp. 605–620, Oct. 2010. doi: 10.1007/s00158-009-0440-y (see p. 56)

[103] E. M. Dede, P. Schmalenberg, C. M. Wang, *et al.*, "Collection of Low-Grade Waste Heat for Enhanced Energy Harvesting," *AIP Advances*, vol. 6, no. 5, 2016, ISSN: 21583226. doi: 10.1063/1.4950861 (see p. 60)

[104] Y. Lei, C. Barth, S. Qin, *et al.*, "A 2-kW Single-Phase Seven-Level Flying Capacitor Multilevel Inverter with an Active Energy Buffer," *IEEE Transactions on Power Electronics*, vol. 32, no. 11, pp. 8570–8581, 2017, ISSN: 08858993. doi: 10.1109/TPEL.2017.2650140 (see pp. 66, 69)

[105] S. Zeng, B. Kanargi, and P. S. Lee, "Experimental and numerical investigation of a mini channel forced air heat sink designed by topology optimization," *International Journal of Heat and Mass Transfer*, vol. 121, pp. 663–679, 2018, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2018.01.039. url: https://doi.org/10.1016/j.ijheatmasstransfer.2018.01.039 (see pp. 67, 89)

[106] J. H. Haertel, K. Engelbrecht, B. S. Lazarov, *et al.*, "Topology optimization of a pseudo 3d thermofluid heat sink model," *International Journal of Heat and Mass Transfer*, vol. 121, pp. 1073–1088, 2018, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2018.01.078. url: https://doi.org/10.1016/j.ijheatmasstransfer.2018.01.078 (see p. 67)

[107] J. Alexandersen, O. Sigmund, K. E. Meyer, *et al.*, "Design of passive coolers for light-emitting diode lamps using topology optimisation," *International Journal of Heat and Mass Transfer*, vol. 122, pp. 138–149, 2018, ISSN: 00179310. doi: 10.1016/j.ijheatmasstransfer.2018.01.103 (see p. 67)

[108] E. M. Dede, "Optimization and design of a multipass branching microchannel heat sink for electronics cooling," *ASME Journal of Electronic Packaging*, vol. 134, no. 4, p. 041 001, Dec. 2012, ISSN: 1043-7398. doi: 10.1115/1.4007159. url: http://electronicpackaging.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4007159 (see p. 67)

[109] T. Meynard and H. Foch, "Multi-level conversion: High voltage choppers and voltage-source inverters," in *Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE*, Jun. 1992, 397–403 vol.1. doi: 10.1109/PESC.1992.254717. url: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=254717 (see p. 68)

[110] E. C. de Jong, B. J. Ferreira, and P. Bauer, "Toward the next level of PCB usage in power electronic converters," *IEEE Transactions on Power Electronics*, vol. 23, no. 6, pp. 3151–3163, 2008, ISSN: 08858993. doi: 10.1109/TPEL.2008.2004276 (see p. 69)

[111] C. Chen, Z. Huang, L. Chen, *et al.*, "A Flexible PCB based 3D Integrated SiC Half-Bridge Power Module with Three-Sided Cooling Using Ultra-Low Inductive Hybrid Packaging Structure," *IEEE Transactions on Power Electronics*, vol. PP, no. c, p. 1, 2018, ISSN: 08858993. doi: 10.1109/TPEL.2018.2866404 (see p. 69)

[112] D. Chou and R. Pilawa-Podgurski, "A zero-voltage switching, physically flexible multilevel GaN DC-DC converter," in *2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, Oct. 2017, pp. 3433–3439. doi: 10.1109/ECCE.2017.8096615 (see p. 69)

[113] J. K. Guest, "Topology optimization with multiple phase projection," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 1-4, pp. 123–135, Dec. 2009. doi: 10.1016/j.cma.2009.09.023 (see p. 77)

[114] G. H. Yoon, "Topological design of heat dissipating structure with forced convective heat transfer," *Journal of Mechanical Science and Technology*, vol. 24, no. 6, pp. 1225–1233, Jun. 2010. doi: 10.1007/s12206-010-0328-1 (see p. 87)

[115] J. Norato, B. Bell, and D. Tortorelli, "A geometry projection method for continuum-based topology optimization with discrete elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 293, pp. 306–327, 2015. doi: 10.1016/j.cma.2015.05.005 (see pp. 88, 104, 124, 125, 128, 138, 140, 142)

[116] H. T. Pedro, M. H. Kobayashi, C. F. Coimbra, *et al.*, "Effectiveness of complex design through an evolutionary approach," *Journal of Thermophysics and Heat Transfer*, vol. 22, no. 1, pp. 115–118, Jan. 2008. doi: 10.2514/1.29834 (see p. 88)

[117] V. Subramaniam, T. Dbouk, and J. L. Harion, "Topology optimization of conductive heat transfer devices: An experimental investigation," *Applied Thermal Engineering*, vol. 131, pp. 390–411, 2018, ISSN: 13594311. doi: 10.1016/j.applthermaleng.2017.12.026. url: https://doi.org/10.1016/j.applthermaleng.2017.12.026 (see p. 88)

[118] E. M. Dede and Y. Liu, "Experimental and numerical investigation of a multi-pass branching microchannel heat sink," *Applied Thermal Engineering*, vol. 55, no. 1-2, pp. 51–60, Jun. 2013. doi: 10.1016/j.applthermaleng.2013.02.038 (see p. 89)

[119] B. S. Lazarov, O. Sigmund, K. E. Meyer, *et al.*, "Experimental validation of additively manufactured optimized shapes for passive cooling," *Applied Energy*, vol. 226, pp. 330–339, 2018. doi: 10.1016/j.apenergy.2018.05.106 (see p. 89)

[120] T. E. Bruns and D. A. Tortorelli, "Topology optimization of non-linear elastic structures and compliant mechanisms," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 26-27, pp. 3443–3459, 2001, ISSN: 00457825. doi: 10.1016/S0045-7825(00)00278-4 (see pp. 103, 120)

[121] H. E. Ahmed, B. Salman, A. Kherbeet, *et al.*, "Optimization of thermal design of heat sinks: A review," *International Journal of Heat and Mass Transfer*, vol. 118, pp. 129–153, 2018. doi: 10.1016/j.ijheatmasstransfer.2017.10.099 (see p. 111)

[122] X. Guo, W. Zhang, and W. Zhong, "Doing topology optimization explicitly and geometrically—a new moving morphable components based framework," 8, vol. 81, ASME International, 2014, p. 081 009. doi: 10.1115/1.4027609 (see pp. 115, 128, 138)

[123] J. A. Norato, "Topology optimization with supershapes," *Structural and Multidisciplinary Optimization*, vol. 58, no. 2, pp. 415–434, 2018. doi: 10.1007/s00158-018-2034-z (see pp. 116, 125, 138, 141)

[124] V.-N. Hoang and G.-W. Jang, "Topology optimization using moving morphable bars for versatile thickness control," *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 153–173, 2017. doi: 10.1016/j.cma.2016.12.004 (see pp. 116, 128, 138, 139, 146, 172)

[125] R. Boichot, L. Luo, and Y. Fan, "Tree-network structure generation for heat conduction by cellular automaton," *Energy Conversion and Management*, vol. 50, no. 2, pp. 376–386, Feb. 2009. doi: 10.1016/j.enconman.2008.09.003 (see p. 116)

[126] B. Li, J. Hong, S. Yan, *et al.*, "Generating optimal heat conduction paths based on bionic growth simulation," *International Communications in Heat and Mass Transfer*, vol. 83, pp. 55–63, Apr. 2017. doi: 10.1016/j.icheatmasstransfer.2017.02.016 (see p. 116)

[127] D. J. Lohan, E. M. Dede, and J. T. Allison, "Topology optimization for heat conduction using generative design algorithms," *Structural and Multidisciplinary Optimization*, pp. 1–15, 2016, ISSN: 16151488. doi: 10.1007/s00158-016-1563-6. url: http://dx.doi.org/10.1007/s00158-016-1563-6 (see pp. 116, 129, 137)

[128] 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, no. 2, pp. 238–254, Aug. 2004. doi: 10.1002/nme.1064 (see p. 122)

[129] F. Wang, B. S. Lazarov, and O. Sigmund, "On projection methods, convergence and robust formulations in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 43, no. 6, pp. 767–784, Dec. 2010. doi: 10.1007/s00158-010-0602-y (see p. 122)

[130] W. Zhang, Y. Liu, P. Wei, *et al.*, "Explicit control of structural complexity in topology optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 324, pp. 149–169, 2017. doi: 10.1016/j.cma.2017.05.026 (see pp. 125, 137, 156)

[131] A. Jessee, "Simultaneous packaging and routing optimization of thermo-fluid systems," Master's thesis, University of Illinois at Urbana-Champaign, 2019 (see pp. 153, 175)

[132] S. Wu, Y. Zhang, and S. Liu, "Topology optimization for minimizing the maximum temperature of transient heat conduction structure," *Structural and Multidisciplinary Optimization*, vol. 60, no. 1, pp. 69–82, Jan. 2019. doi: 10.1007/s00158-019-02196-9 (see pp. 154, 155, 160, 169)

[133] E. Lee, K. A. James, and J. R. R. A. Martins, "Stress-constrained topology optimization with design-dependent loading," *Structural and Multidisciplinary Optimization*, vol. 46, no. 5, pp. 647–661, 2012. doi: 10.1007/s00158-012-0780-x (see pp. 154, 156)