Summer 2023

Automated Fiber Placement Through Thickness Defect Stacking Optimization

Noah Christopher Swingle

AUTOMATED FIBER PLACEMENT THROUGH THICKNESS DEFECT STACKING OPTIMIZATION

By:

Noah Christopher Swingle

Bachelor of Science
University of South Carolina, 2022

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Mechanical Engineering

College of Engineering and Computing

University of South Carolina

2023

Accepted by:

Ramy Harik, Director of Thesis

Alex Brasington, Reader

Darun Barazanchy, Reader

Ann Vail, Dean of the Graduate School

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude to my mentor Dr. Ramy Harik for inspiring me to pursue research, giving me the opportunity to explore my passions, and always serving as the perfect role model. It was a privilege to learn not only composite manufacturing under you, but more importantly how to lead, how to treat other people, and how to live a full life.

I would also like to thank the entire neXt team for their endless support and encouragement. None of this would have been possible without the considerable efforts made by Joshua Halbritter, Alex Brasington, Roudy Wehbe, Mat Godbold, Ben Francis, and William Montgomery.

This material is based upon work supported by the NASA Established Program to Stimulate Competitive Research (EPSCoR) under NASA Grant / Cooperative Agreement Number: SC-80NSSC21M0327. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NASA EPSCoR.

The material contained in this paper is based upon work supported by NASA under award No. 80NSSC21M0314. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

ABSTRACT

In its 2022 commercial market outlook, Boeing forecasted an 80% increase in the global fleet through 2041 compared to 2019 pre-pandemic levels. This sharp rise in demand will drive pressure onto airframe manufacturers to ramp up production and find more efficient ways to design and manufacture airplanes. Complicating this challenge is the industry's recent transformation from traditional metal-based airframes towards hybrid composite-metal aircraft. While composites have been used in aviation for decades, aircraft manufacturers are still struggling to design and manufacture quality parts at a high rate. Automated Fiber Placement (AFP) is one of the main manufacturing techniques used to produce large-scale composite parts. After a design has been created, a manufacturing strategy has to be developed based on the working material, part geometry, and machine capabilities. This process planning stage is essential to the AFP workflow and currently requires a high level of manual input from an experienced process planner. In an effort to automate and optimize this stage, the Computer Aided Process Planning (CAPP) module was developed. CAPP assists process planners in identifying optimal starting point location and layup strategy for each ply of a laminate. This Ply-Level Optimization (PLO) phase operates on the quantification of ply quality through predictable geometry-based defects such as gaps, overlaps, angle deviation, and steering. As you move from PLO to Laminate-Level Optimization (LLO) the design space grows exponentially, emphasizing the need for automated optimization.

The work presented in this thesis expands CAPP's functionality by comparing the planned fiber paths through the thickness of the laminate to mitigate stacked area defects and achieve an optimal laminate-level manufacturing strategy. Within CAPP, predicted gap and overlap defects are imported from Vericut Composites Programming (VCP) and then discretized to streamline the through-thickness comparison. Two objective functions are used to score different combinations of ply layup strategies based on defect stacking both globally and locally. Four combinatorial optimization algorithms were coupled with these objective functions to investigate the laminate-level manufacturing strategy design space and converge on the optimal plan. These algorithms were evaluated based on accuracy and efficiency through virtual testing on a complex tool surface. A separate LLO approach was developed to achieve near-optimal laminates in significantly less time. The end result is a software package which greatly reduces the required input from process planners, shortening the design-build cycle time and improving part quality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFP ..... Automated Fiber Placement

AHP ..... Analytic Hierarchy Process

ATL ..... Automated Tape Laying

CAD ..... Computer Aided Design

CAPP ..... Computer Aided Process Planning

CFRP ..... Carbon Fiber Reinforced Polymer

DE ..... Differential Evolution

DFM ..... Design for Manufacturing

FEA ..... Finite Element Analysis

GA ..... Genetic Algorithm

GFRP ..... Glass Fiber Reinforced Polymer

GHSD ..... Global Hot Spot Detection

GS ..... Greedy Search

LLO ..... Laminate Level Optimization

NURBS ..... Non-Uniform Rational B-Splines

PLO ..... Ply Level Optimization

PMC ..... Polymer Matrix Composite

PSO ..... Particle Swarm Optimization

VCP ..... VERICUT Composite Programming

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF COMPOSITE MATERIALS

A composite material can be defined simply as a combination of two or more constituent materials that has superior properties compared to the two constituents individually. Composites have been used for centuries and date back to the early Egyptians and Mesopotamians who started combining mud and straw to construct buildings around 1500 BC. The foundation for modern advanced composite materials began with the development of plastics in the early 1900s and glass fibers in the 1930s. Up until the introduction of plastics, natural resins derived from plant and animal byproducts were the only source of adhesives. World War II created demand for stronger, lighter, and more resistant materials to enable aerial combat. Fiberglass composites, or glass fiber reinforced polymers (GFRPs), saw full-scale development and commercialization as a direct result of this demand [1]. Carbon fiber reinforced polymer (CFRP) commercial utilization began later in 1966 following research done at the Royal Aircraft Establishment. Watt et al. defined a process for making high strength carbon fibers that is still serves as the basis for virtually all carbon fiber production today [2], [3].

Modern advanced composites generally consist of two parts: (1) matrix and (2) reinforcement. The matrix (1) serves to bind the reinforcement material together, transfer loads, and maintain a constant shape. Polymer matrix composites (PMCs) are the principal

focus of our efforts. There are several other options for matrix material including ceramics, metals, and carbon. Within the category of PMCs, there are two main families of resin systems: thermosets and thermoplastics. While there are benefits and drawbacks to both, thermosets are more widely used due to lower processing temperatures required during layup [4]–[6]. The reinforcement material (2) is the main load carrying part of a composite. The principal reinforcement materials utilized today are carbon, glass, and aramid fibers [7]. Reinforcement materials are typically domain specific with the aerospace, marine, and automotive industries favoring different systems [8]–[10].

There are countless ways to manufacture composite structures depending on the scale, part complexity, throughput requirements, and specific material selection. Among the more common methods are hand layup, shape-specific processes such as pultrusion and filament winding, generic processes such as resin transfer molding, compression molding, Automated Tape Laying (ATL), and Automated Fiber Placement (AFP) [11]. Within the aerospace industry specifically, AFP has grown rapidly since its inception in the 1970s due to the possibility of higher throughput and increased repeatability. AFP is a derivative of ATL and filament winding where the tapes are further slit into tows. The first known implementation of automated placement using tows instead of tapes comes from a patent published in 1974 [12]. Both ATL and AFP typically use pre-impregnated material, which means the fibers are pre-impregnated with resin. ATL uses wider tapes while AFP uses thinner tows, which enables layup on complex geometries. By the 1980s, AFP machines were commercially available and deployed by aerospace companies like Boeing, Lockheed Martin, and Douglas Aircraft Company [13]. Today, the technology is employed across the aerospace industry to create fuselage skins, wing skins, and rocket barrels.

1.2 AUTOMATED FIBER PLACEMENT

AFP is an additive composite manufacturing technique. An AFP machine lays up strips of unidirectional CFRP material called tows in groups called courses, typically consisting of 4-32 tows. Multiple courses are laid up on a tool surface until the desired surface area is covered – this constitutes a single ply. Traditionally, and in most current commercial applications, tows are oriented in a single direction per ply [14]. However, recent research has exploited the accuracy and flexibility of AFP to produce variable stiffness laminates which feature steered tows within each ply [15], [16]. In either case, multiple plies are laid up to form the full laminate. For thermoset matrix-based composites, this constitutes the green state of the material, meaning it still must undergo a curing process to achieve consolidation and the desired mechanical properties. Thermoset composites will be the main material addressed in this work, however many of the techniques employed could be implemented with thermoplastics.

There are three main types of AFP platforms: vertical gantry, horizontal gantry, and robotic arm. Vertical gantries are suitable for large plate-like structures, horizontal gantries are often used for plates and cylindrical structures, while robotic arms enable more complex geometries. An AFP cell consists of the platform, a fiber placement end effector, and the tool. The fiber placement end effector, or AFP head, contains multiple sub-components including add rollers, tow cutters, a heat source, and a compaction roller. The add rollers feed material from the spools through the head to be placed on the tool surface. There are cutters for each tow supported by the AFP head so that each tow can be cut individually. The distance between the tow cutters and the bottom of the compaction roller defines the minimum length of material that can be placed. The heat source applies heat to

the incoming tows and the substrate material to increase tackiness and promote adhesion. There are several different heating sources such as infrared, hot gas torch, pulsed light, and laser systems. The compaction roller applies pressure to the incoming tows and substrate to reduce voids and promote adhesion. A simplified schematic of an AFP head is shown in Figure 1.1 alongside an image of the AFP head at the University of South Carolina's McNAIR Aerospace Center. For more detailed information on the state-of-the-art for AFP, the reader can refer to [17].

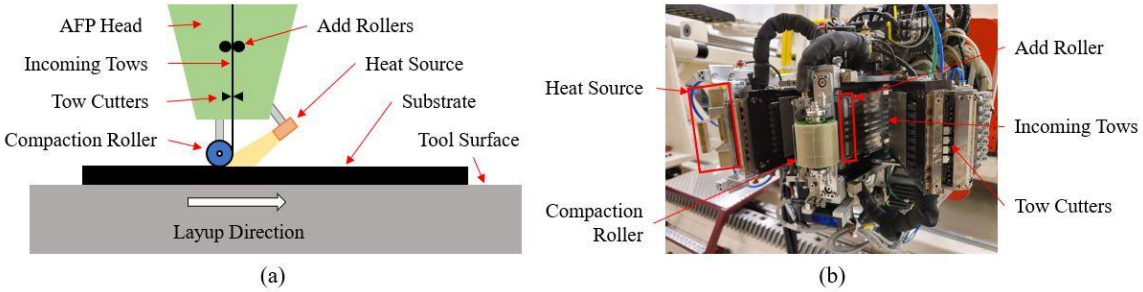


Figure 1.1: (a) Simplified AFP head schematic and (b) AFP head at McNAIR Aerospace Center

1.2.1 AFP PILLARS

The AFP process can be thought of as having four pillars: design, process planning, manufacturing, and inspection. Currently, these pillars are often treated in industry as isolated phases in a sequential workflow. The first phase is design which takes loading conditions, packaging constraints, and other functional requirements as an input, and outputs the part geometry, ply boundaries, and stacking sequence. This data is then passed off to process planners who are tasked with generating a feasible manufacturing plan to achieve the desired design. For AFP this constitutes generation of toolpaths and selection of process parameters like heat, speed, compaction, and tow tension. This plan is then

implemented by the manufacturing team who operates the machine and lays up the part. Finally, the part is inspected for any out-of-tolerance defects by the quality assurance team and reworked until acceptable.

Under this philosophy the flow of data and information is unidirectional, and any insights gleaned from any one pillar rarely influence the others. Cross-pillar connections that do get made are often time-consuming, manual, and rely solely on the experience of those involved. For example, a process planner may receive a design, realize it is not manufacturable, and suggest changes to the designer who then produces a revised design. This loop may happen multiple times per part leading to inefficient process development and no guarantee of an as-manufactured optimal structure. Figure 1.2 presents a novel methodology for an AFP workflow built on “Industry 4.0” concepts [18]. Under this workflow, the data from each pillar is no longer isolated but connected in a closed-loop system; the process becomes cyclic and continuous, where each pillar operates as a trade in a constant global optimization framework.

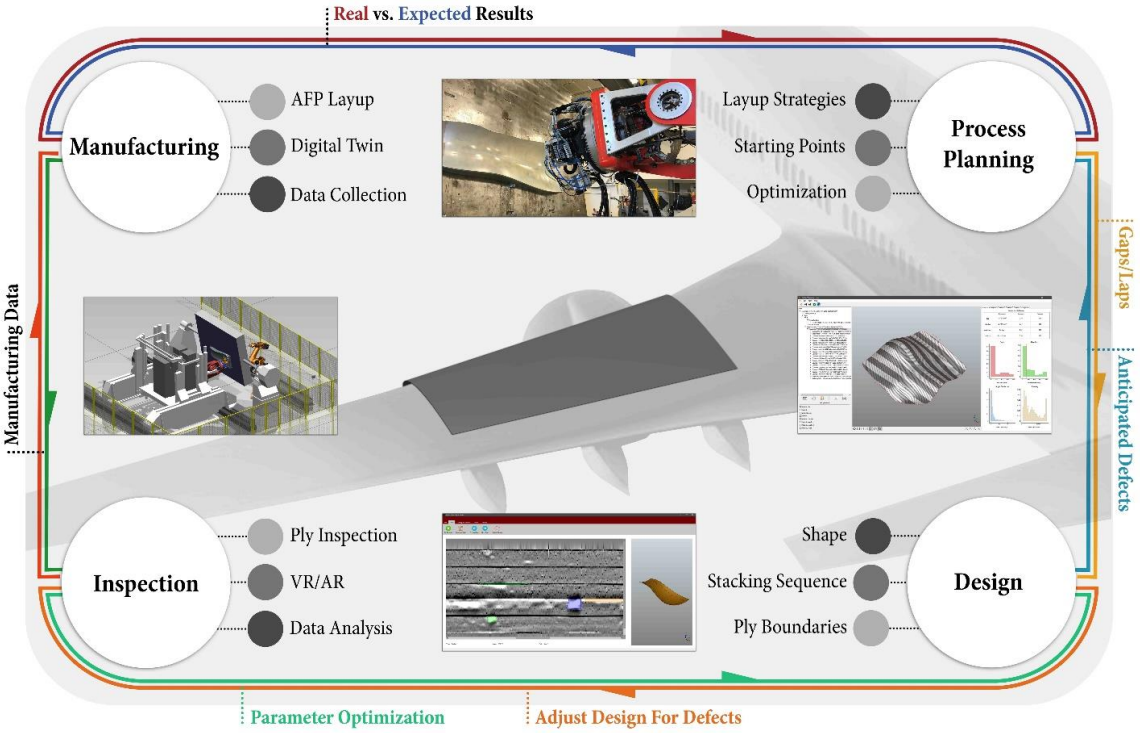


Figure 1.2: Industry 4.0 AFP workflow [18]

The work presented in this manuscript sits squarely within the process planning pillar. Previous efforts sought to create a software tool to help automate and optimize specific AFP process planning decisions [19], [20]. The tool, Computer Aided Process Planning (CAPP), optimized starting point location and layup strategy based on users' preferences of predictable defect distributions on a ply level. Gap, overlap, angle deviation, and steering defects caused by the curvature of the tool surface are predicted by VERICUT Composites Programming (VCP) and then fed into CAPP for analysis. CAPP then suggests a new starting point location and/or layup strategy and this loop is iterated until the desired defect distribution is achieved.

The CAPP workflow operates solely on a per ply basis however and does not consider the location of defects through the thickness of a laminate. Recent research, discussed in

Chapter 2, suggests that the greatest mechanical effects are realized when these defects, specifically gaps and overlaps, co-locate through-thickness. Currently these effects are addressed within the design pillar through the application of design knockdowns. The expected strength of the material is reduced to account for the inevitable creation of these defects through AFP manufacturing. This work employs various combinatorial optimization techniques to find the optimal combination of ply-level manufacturing strategies resulting in the least amount of through-thickness defect stacking.

1.3 THESIS OUTLINE

This thesis supports the vision illustrated by Figure 1.2 through the expansion of process planning software. The research presented will help process planners identify optimal manufacturing strategies resulting in a more efficient workflow yielding higher quality parts. With further validation, the latter consequence could give rise to the reduction or elimination of design knockdowns for toolpath-dependent defects like gaps and overlaps. The remainder of this thesis is organized as follows. Chapter 2 presents a literature review on AFP process planning with particular focus given to toolpath generation. An overview of the CAPP software and relevant combinatorial optimization algorithms is also presented. Chapter 3 describes the main optimization methodology employed. Detailed descriptions of the combinatorial optimization algorithms utilized are provided as well as the novel scoring systems used as objective functions for the optimization. Chapter 4 presents virtual experiments performed to optimize the key performance parameters specific to each algorithm. This work constitutes hyper-parameter optimization and is geared towards improving the application of these algorithms to the specific problem of AFP defect stacking. Chapter 5 showcases the performance of the

algorithms through a virtual case study which optimized the laminate level manufacturing strategy for a wind tunnel blade. Chapter 6 concludes the work and discusses future paths forward.

CHAPTER 2

LITERATURE REVIEW

In this chapter we outline: (1) AFP path planning, defects, and through thickness defect stacking., (2) the CAPP software prototype functionality, and (3) background and applications of combinatorial optimization techniques. The chapter concludes with a summary and gap assessment.

2.1 AUTOMATED FIBER PLACEMENT PROCESS PLANNING

Process planning connects design to manufacturing by generating a manufacturing plan based upon the working material, composite design definition, and AFP cell capabilities. It is one of the more involved steps in the AFP lifecycle, relying heavily on the process planner's knowledge of the connections between the material system, the part geometry, and the effect of different machine inputs. AFP process planning can generally be divided into two separate sub-processes: toolpath generation and process parameter planning. The first step is toolpath planning, which defines the exact motions that the AFP machine will perform to layup the material. Path planning is the focus of this work and is described in Section 2.1.3. Process parameter planning includes the creation of programs for heating, compaction, tow tension, and layup speed. Path planning heavily influences the occurrence of toolpath and geometry dependent defects like gaps, overlaps, angle deviation, and steering. The relationship between process parameters and defect occurrence

is more complex, but improper parameter selection can result in gaps, overlaps, angle deviation, wrinkles, bridging, folds, wandering tows, and other defects.

2.1.1 AFP Defects Overview

AFP is an inherently complex process subject to the inescapable occurrence of defects while manufacturing laminates. There are several different defects possible within AFP, each with their own cause and effect. Harik et al. identifies 14 distinct AFP defects and provides a detailed discussion of the cause, anticipation, existence, significance, and progression [21]. These defects include, gap/overlap, pucker, wrinkling, bridging, boundary coverage, angle deviation, fold, twist, wandering tow, loose tow, missing tow, splice, position error, and foreign object debris. Many of the defects can be influenced by more than one factor, allowing for the prediction and mitigation of some but potentially not all of one defect type through a given technique. Gaps, overlaps, and angle deviation defects can be predicted during the process planning phase through careful analysis of the material system, tool surface geometry, and toolpaths [22], [23]. Wrinkles can also be indirectly predicted through examination of the toolpaths' steering radii [24]. While some defects can occur unpredictably during the process because of process parameters or manufacturing errors [25], recent efforts have shown that the majority of gap and overlap defects are predictable during the process planning phase [26]. Gap and overlap defects resulting from the interaction of toolpaths and tool surface geometry are the principal focus of this work and will be examined in detail in the following sections.

The significance of these defects is that they cause changes in the expected mechanical performance of the laminate and/or require time-consuming manual rework to remove. Depending on the type, location, and severity, defects can cause thickness

variations, voids, resin rich areas, out-of-plane fiber waviness, and delamination [27], resulting in lower strength, higher failure rates, and geometrical tolerance issues [28]. Gaps and overlaps can also cause wrinkles and other defects in subsequent plies [29].

To avoid the unwanted consequences of gaps/overlaps, parts can be inspected and manually reworked. This, however, is a very time-consuming process. For example, the Boeing 787 fuselage barrel produced using AFP spends only 24% of total cycle time being laid up. A full 63% of the cycle time is spent undergoing inspection and repair [30]. Leading AFP machine manufacturer Electroimpact analyzed data from five different builds in 2014 and found that layup time was approximately 30% of the cycle time, with 32% spent on inspection and repair [31]. A later study published by Electroimpact in 2020 estimated layup time at 27% of total cycle time and quality assurance taking up 22% of the time [32]. While time spent reworking defects has certainly decreased over the years as AFP processes improve, the task still demands a significant portion of overall cycle time. Therefore, it is advantageous to dive deeply into the effects of these defects, understand which ones require intervention, and develop methods to reduce their impact.

2.1.2 Gaps and Overlaps

Gaps and overlaps are the most commonly occurring AFP defects. A gap is created when two adjacent tows are not laid up perfectly next to each other, leaving an area without material between them. Similarly, an overlap is created when two tows are laid up on top of each other within the same ply. These defects most frequently appear between courses but can also occur within a course due to tool surface curvature, improper tack, or machine errors. Nguyen et al. provide a simplified schematic of the formation and evolution of these defects when cured in an autoclave without a caul plate [33]. As shown in Figure 2.1, the

absence of material at a gap causes out-of-plane fiber waviness in adjacent plies as well as a matrix pocket, or resin rich area. The addition of material at an overlap also causes out-of-plane fiber waviness in adjacent plies. Both defects induce thickness variations at the laminate level that are typically reduced through the curing process. These geometrical effects are often concluded to be the main contributors to the reduction in mechanical performance and initiation of failure mechanisms [34], [35].

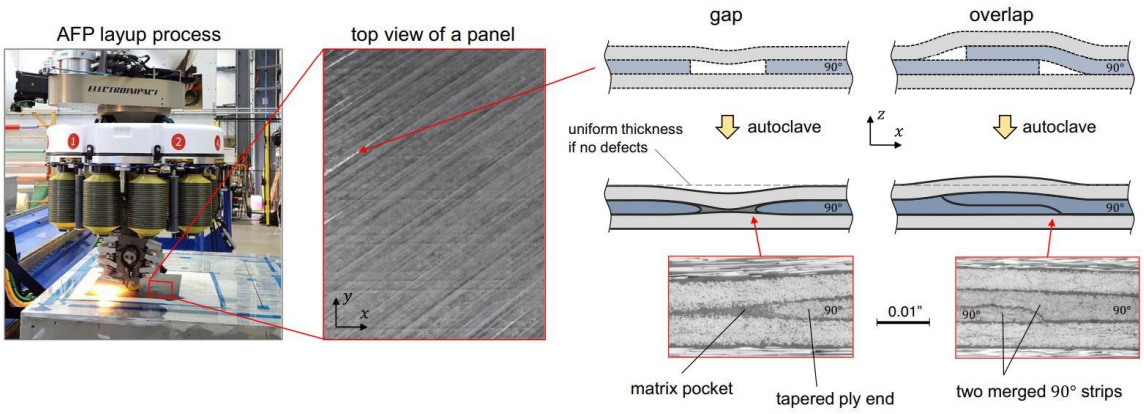


Figure 2.1: AFP process with pre- and post-consolidation schematics of gap and overlap defects [33]

The specific effect gaps and overlaps have on laminate mechanical properties has been studied at length over the past 25 years. Sawicki and Minguet investigated the effects of different configurations of gaps and overlaps on compressive strength [34]. Sawicki and Minguet found that laminates with at least one defect had strength reductions of 5-27%. It was concluded that depending on the size and type of the defect, out-of-plane fiber waviness was induced which led to higher interlaminar shear stress. The defect size to strength reduction relationship was not directly linear, and greater knockdowns were not seen once the defect reached ~0.030” in width. The researchers suggested accounting for these effects by reducing design allowables. Turoski built on this work by expanding the

testing to include both compression and tension [36]. Turoski confirmed Sawicki and Minguet's findings of reduced compressive strength and also found that laminates with three gaps stacked through-thickness had 16% lower unnotched tensile strength. Turoski similarly recommended design allowable knockdowns and called for further experimentation between hard and soft cure tooling.

Croft et al. followed these works up in 2011 with a larger experiment including gap, overlap, half gap/overlap, and twisted tow laminates tested in tension, compression, in-plane shear, open hole tension, and open hole compression [37]. It was concluded that the effects of defects are much more severe on the laminate level (up to 13%) as compared to the lamina level (<5%). The experimental results are summarized in Figure 2.2. Despite the large set of tests performed, the laminates were manufactured with only a single defect, which is uncommon in real-world parts. Croft et al. noted this, called for further experimentation with multiple defects, and predicted that defect distribution would play a large role in the results.

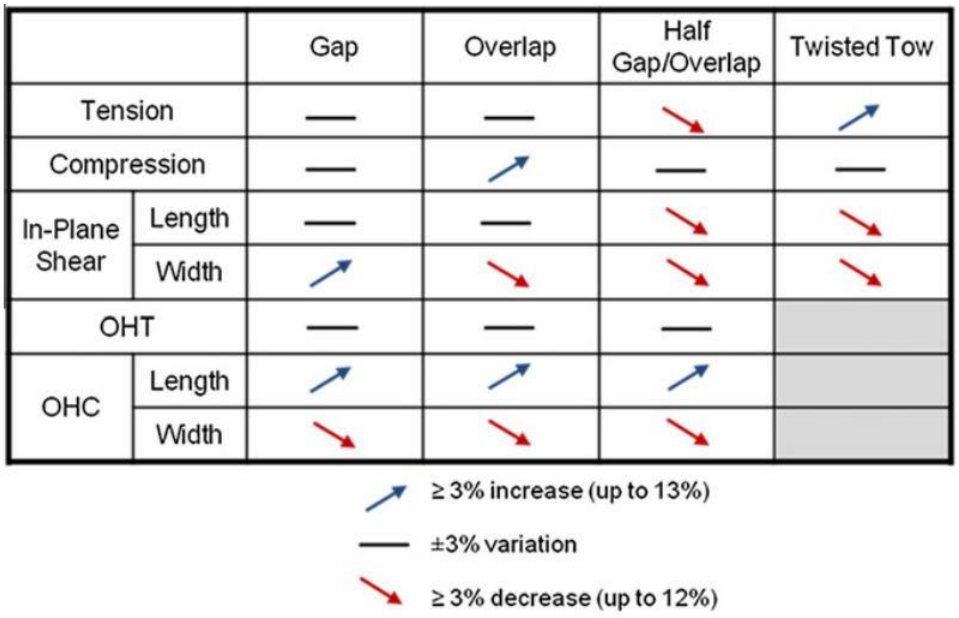


Figure 2.2: Summary of the effects on performance for different defect configurations [37]

Moving from categorizing the effects of singular or few defects within a laminate to multiple defects of different types distributed through the thickness of a laminate greatly expands the number of possible configurations. Blom et al. created a model to explore the influence of stacked and staggered gaps on the strength and stiffness of variable stiffness laminates [38]. Staggering refers to the shifting of courses perpendicular to the fiber direction between plies of a common fiber angle. This practice aims to more evenly distribute defects through-thickness resulting in less laminate thickness variation and out-of-plane fiber waviness. The researchers found that in laminates with stacked gaps the strength was reduced by 15% for 3.125mm tows compared to a pristine laminate. When staggering was applied the strength reduction was only 7%. Blom et al. also suggested applying design allowable knockdowns around these values when designing a part to be manufactured with AFP.

Falco et al. published a study in 2014 examining the impact of gaps and overlaps on variable stiffness laminate tensile strength [39]. The experiment included four main laminate configurations: a pristine baseline laminate, a 100% gap coverage laminate (all overlaps), a 0% gap coverage laminate (all gaps), and a staggered 0% gap coverage laminate. The authors found that for the 100% gap coverage laminate the thickness increased 15% compared to baseline due to the stacked overlaps. The 0% gap coverage laminate resulted in a laminate that was 20% thinner due to the stacked gaps. This reduction dropped to 5% when the gap defects were staggered. Similarly, the 0% gap coverage laminate had a 20.1% reduction in tensile strength while the staggered 0% gap coverage laminate suffered only an 8.6% reduction in tensile strength. This work expanded upon and agreed with many of the findings of Blom et al. in [38]. The benefits of staggering are two-fold: lesser thickness variations and reduced strength knockdowns. However, the exact mechanism by which this benefit is achieved was only theorized in these works.

Li et al. sought to provide some clarity on this topic through the creation of a novel meshing technique to support a more robust gap and overlap finite element analysis (FEA) model [40]. The authors developed a custom mesh generation tool that can be customized to more accurately model the fiber orientations used within a laminate. Based on manufactured samples, the authors created a simplified set of geometrical features defining both a gap and an overlap. These features, combined with the custom mesh and rules imposed to simulate either soft or hard cure tooling, formed the final FEA model. This model was then used to predict the internal geometry and mechanical properties of laminates with different defect configurations. The model showed good agreement with manufactured samples as well as destructive testing. The internal geometrical impacts of

gap and overlap defects captured by the model (Figure 2.3) were pointed to as the largest drivers of mechanical property knockdowns. Specifically, the ramp down or ramp up regions that cause out-of-plane fiber waviness were concluded to be the largest contributors to failure. The impact of this transition zone agrees with the findings of [34], [41] in that the effects of the defects are driven more by the waviness induced and not the size of the defect. The authors noted that the isolated defects tested likely do not represent the worst strength reduction, and that more configurations should be tested.

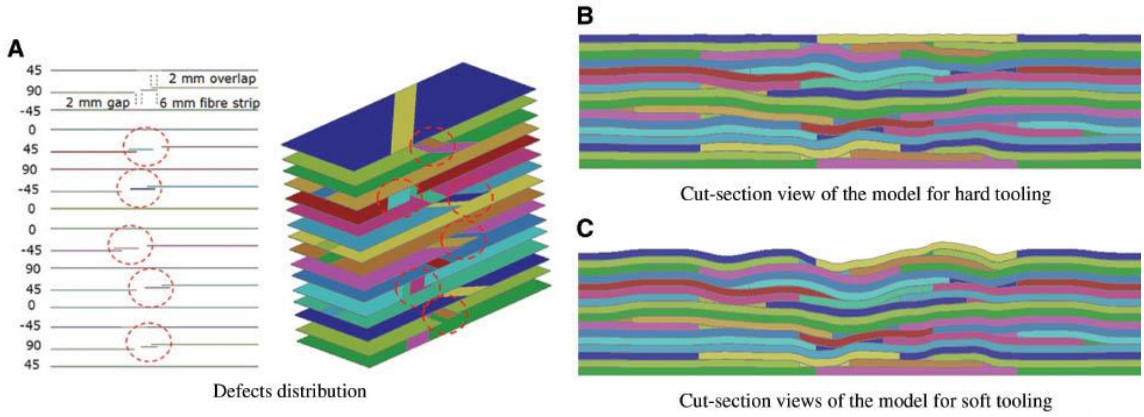


Figure 2.3: Comparison of cross-sectional views of FEA model for laminates with gap and overlap defects cured with (B) hard and (C) soft tooling [40]

This work brings into focus another factor that affects the impact of defects on laminate mechanical performance: cure tooling. Lan et al. published two works in 2015 and 2016 that investigate the influence of caul plates on defected laminates [42], [43]. The earlier work tested two different stacking sequences with and without caul plates. They found that when the defects were stacked between plies of a common fiber orientation the cure cycle allowed for increased resin flow and fiber migration. This freedom helped the laminates fill gaps and even out overlaps, reducing the resulting thickness variation. This effect was greatly enhanced when a caul plate was used. For cross-ply laminates the movement and healing effect was restricted. The later work confirmed the conclusions and

found that compression strength was strongly influenced by the presence of defects. The impact of stacking sequence is notable since many of the earlier works attempted to achieve a worst case scenario defect configuration by stacking defects between plies of a common fiber angle. The work done by Lan et al. suggests that in fact a cross-ply stacking sequence, which is more often used in industry, counteracts some of the defect healing that takes place during curing [42].

Belnoue et al. developed a multi-scale, multi-physics model of composite processing combining a compaction model with a heat transfer model to more accurately capture the effect of curing on internal laminate geometry [44]. Different configurations of green-state gaps and overlaps were generated, with the model predicting final laminate thickness and ply waviness. The model highlighted the process by which compaction can close gaps either by stacked overlaps falling into the gaps or through lateral squeeze flow of the resin. The researchers noted that the ability for these gaps to close was affected by the ply bending stiffness and called for further studies to explore this facet. Another conclusion of the work was that predicted laminate thickness varied by as much as 10% between the all-gaps, soft tooling configuration and the all-overlaps, soft tooling configuration.

Nearly every work discussed thus far notes the complexity of accurately categorizing the effect of gap and overlap defects on composite laminate mechanical properties. The sheer magnitude of variables at play, and combinations between them, prevents a truly exhaustive experimental study. In [45] researchers investigated the effect of gaps on fatigue behavior and concluded the impact of gaps is affected by stacking sequence, gap shape, gap distribution, and stress applied. In [46], Marouene et al. studied

the open-hole compressive strength of defected laminates and found that strength reductions depend greatly on defect type and location. Narrowing the scope to identifying what factors negate the impact of defects provides a useful path forward. Researchers in [40], [42], [43] showed that the use of hard tooling greatly reduces out-of-plane fiber waviness and results in laminates with mechanical properties approaching nominal. The foundation for the benefit of staggering was laid in [38], [39], however the number of configurations tested in those works was limited and both called for further experimentation.

Focusing on the impact of staggering and the influence of gap-overlap interactions, Woigk et al. tested several different laminate configurations in tension and compression [47]. Figure 2.4 shows the selected configurations and cross-sectional images of the manufactured laminates' internal geometry. From the images in Figure 2.4 it is clear to see the impact of defects on out-of-plane fiber waviness as well as the inherent complexity in the different possible combinations. The work came to two notable conclusions. First, it is the severity of fiber misalignment that controls failure, not the defect content. Second, staggering the defects through the thickness of the laminate greatly reduces their impact on mechanical properties and returns the laminate to near-nominal condition. Woigk et al. specifically stated that if the defect is systemic and occurs repeatedly at the same location within each ply it will degrade material strength. However, if the defect distribution within the laminate is random, then it is not likely to cause a major detrimental effect.

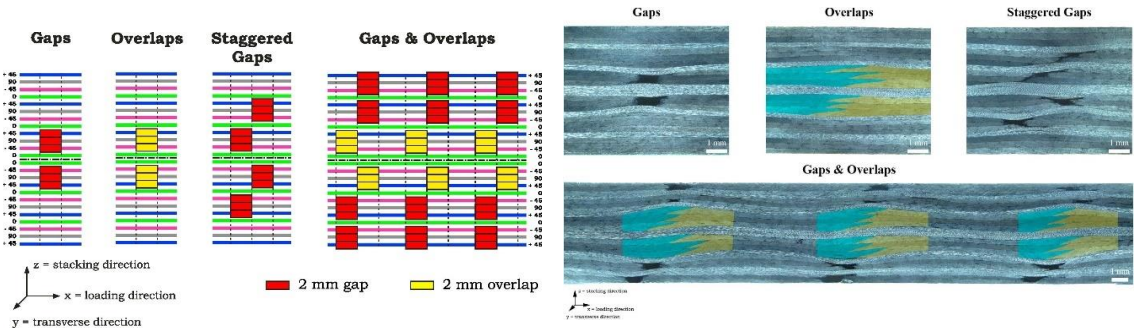


Figure 2.4: (a) Experimental laminate configurations tested and (b) cross-sectional images of manufactured laminates [47]

The ability to stagger a laminate and avoid the negative impacts of gap and overlap defects on mechanical performance is significant. Without intervention these defects can reduce tensile and compressive strength by 5-20%, produce thickness variations of up to 10%, and promote the occurrence of other defects. Reworking the laminates to remove these defects is extremely difficult, time-consuming, and questionably effective [30], [48]. To avoid rework, gaps and overlaps are often accounted for during the design phase where knockdowns are applied to material allowables. This results in thicker, less efficient designs that cost composite manufacturers more time and money.

The central hypothesis of this manuscript is that through intelligent staggering, the problem of gaps and overlaps can be solved during the process planning phase rather than the design phase. The next section provides more background into why these defects occur in the first place and the exact mechanisms through which staggering can be achieved.

2.1.3 Path Planning

Perhaps the core responsibility of an AFP process planner is to select a layup strategy for each ply in the laminate. The layup strategy constitutes a starting point, a reference curve, and a coverage strategy. These decisions are used to generate toolpaths which

determine exactly where and how material will be placed on the tool surface. While a designer might assume that each ply they design will be perfectly manufactured with a uniform thickness and strict adherence to the prescribed fiber angle, this is not always achievable. Complex tool surface curvature pits surface coverage against angle deviation and steering. One strategy may be better for maintaining the desired fiber angle at the cost of increased gap and overlap defects. Another strategy may perfectly cover the surface but will require severe steering to do so. The role of the process planner is to evaluate these competing interests within the context of the original design intent. Each of the components of a layup strategy are discussed in the following sections. For a more detailed review, the reader can refer to [49].

2.1.3.1 Starting Point

The first step in toolpath generation is selecting a starting point location. This point is used to generate the reference curve which in turn is used to create the full set of toolpaths. Since the starting point determines, in part, the location of all the toolpaths, manipulation of the starting point is the primary mechanism by which staggering is achieved. Figure 2.5 shows two starting points on an arbitrary tool surface separated by a stagger shift.

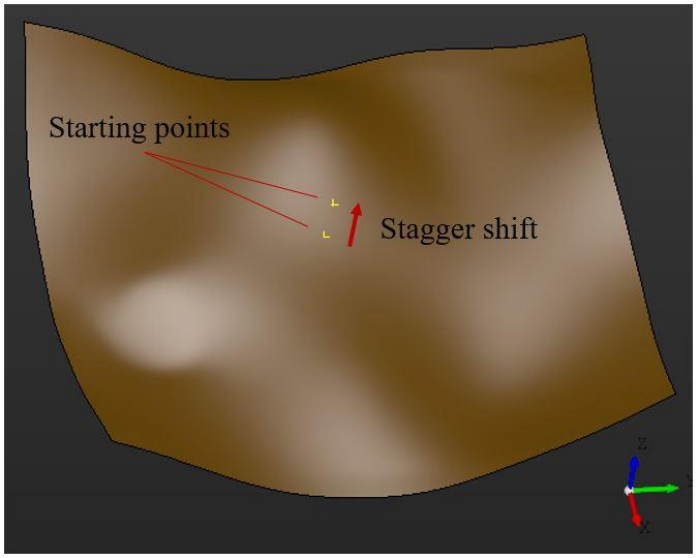


Figure 2.5: Starting point and stagger shift schematic

2.1.3.2 Reference Curves

The second step in toolpath generation is creating a reference or guide curve. There are three main approaches for creating this curve: fixed angle, variable angle, and geodesic. The fixed angle approach is rather simple and involves creating a curve from a given starting point with a constant angle relative to a given axis or direction along the entire tool surface. This can be achieved using a mesh of the surface or through a parametric approach. The mesh approach typically utilizes an .STL file of the geometry and creates the reference curve with a slicing algorithm [50]. Representing the surface parametrically is the more popular approach, as shown in equation 2.1. In [51]–[56], a major axis in the direction of the desired fixed angle is projected onto the parametric surface. The resulting intersection equation is then given by equation 2.2 where *a*, *b*, *c*, and *d* are coefficients of the major axis plane equation.

$$S(u,v) = x(u,v) + y(u,v) + z(u,v) \tag{2.1}$$

$$f(u,v) = ax(u,v) + by(u,v) + cz(u,v) + d = 0 \tag{2.2}$$

The geodesic curve method is often employed when the part is sensitive to steering or wrinkling because the path has no curvature. It is also called the natural path and is defined by being the shortest path between two points on a three-dimensional surface [57]. For a flat plate the geodesic path is a straight line [58]. Blom et al. generated a geodesic path on a conic shell to investigate the differences between constant curvature, constant angle, and geodesic paths [59]. Figure 2.6 shows these paths on the conic shell as well as their impact on fiber angle and curvature. This exemplifies the impact layup strategy selection has on defects, specifically angle deviation and steering.

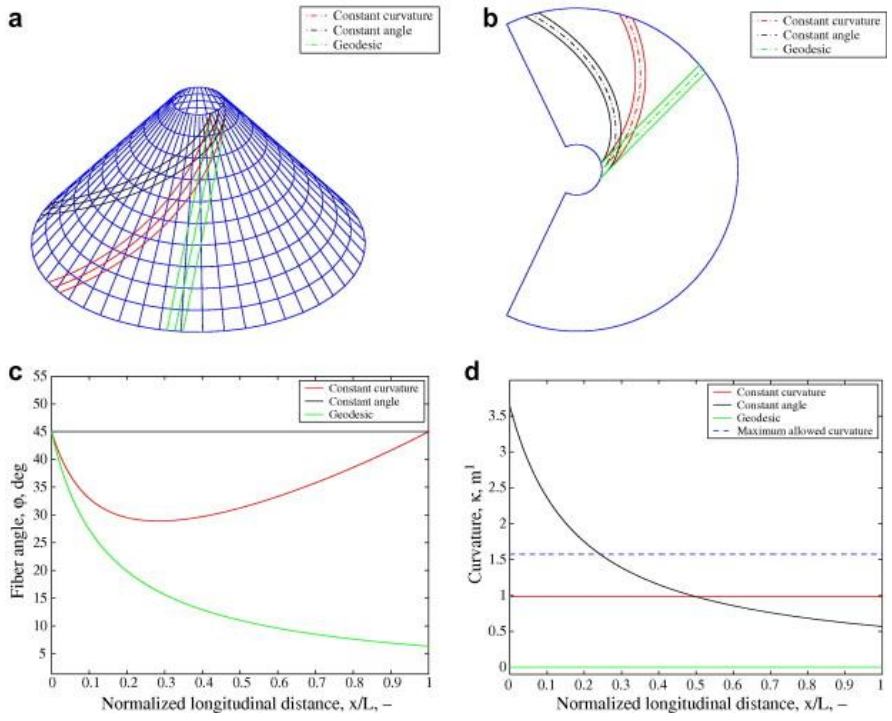


Figure 2.6: Different fiber paths on (a) 3D conic shell and (b) flattened cone with graphs showing (c) fiber angle and (d) curvature [59]

Variable angle reference curves alter the fiber angle throughout the path to enable variable stiffness laminates [15]. These laminates exploit the benefits of composite

materials and automated manufacturing and allow for more efficient designs with specifically tailored mechanical properties [60]–[63]. There are a couple of different strategies for generating variable stiffness reference curves including linear variation [64]–[67] and several versions of nonlinear variation [68]–[71]. Variable stiffness laminates are an exciting field of research but require a high degree of computation and optimization to design and manufacture.

2.1.3.3 Coverage Strategy

Once a reference curve has been created, the rest of the toolpaths can be generated. Each of these toolpaths corresponds to a course of tows that will be laid up on the tool surface. There are three strategies that can be used to create the coverage strategy: independent curves, offset/parallel curves, and shifted curves. The independent curves method essentially just repeats the reference curve approach to independently generate every toolpath in the ply. This method is rarely used but is sometimes suitable for highly complex parts. One implementation was shown by Favaloro et al. where short independent courses were created to limit gap and overlap defects on a conical surface [54].

The parallel or offset coverage strategy is the most frequently used strategy due to its ability to limit gap and overlap defects between neighboring courses [49]. Similar to the fixed angle reference curve method, there are two ways to compute parallel curves: a mesh approach and a parametric approach. The mesh approach is built on the Eikonal equation [72] and utilizes the fast marching method to offset a random reference curve on the meshed surface [73]–[75]. The parametric approach is fundamentally a numerical approach where points are offset along perpendicular surface arcs to form an offset path. If the path

doesn't reach the ply boundary the curve is simply extended through extrapolation until the boundary is met. This process is summarized in Figure 2.7.

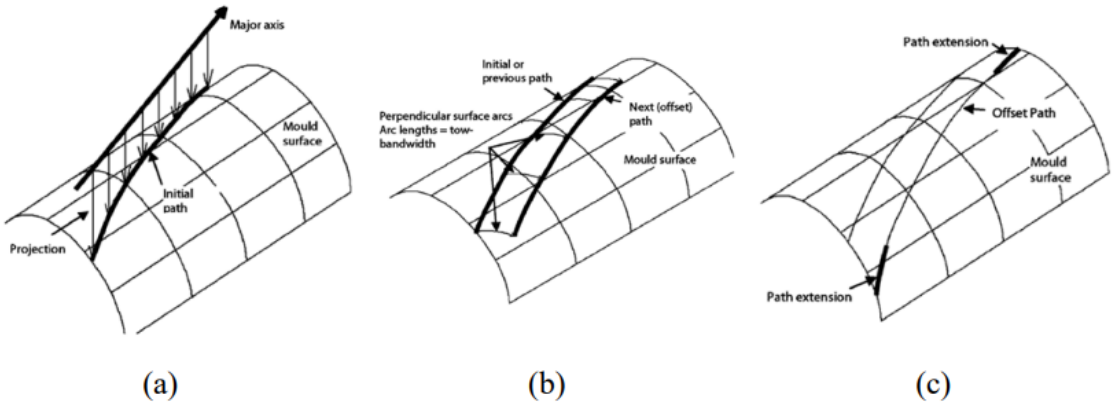


Figure 2.7: (a) Reference curve generation; (b) offset curve generation; (c) offset curve extension [53]

The shifted method is performed simply by translating the reference curve along the surface perpendicular to its fiber direction. This method is implemented in [38], [65], [76], [77] and is often used for its simplicity. Kim et al. demonstrated the difference between parallel and shifted coverage strategies (Figure 2.8) and found that the shifted strategy led to an increase in gap and overlap defects, especially with complex tool surfaces.

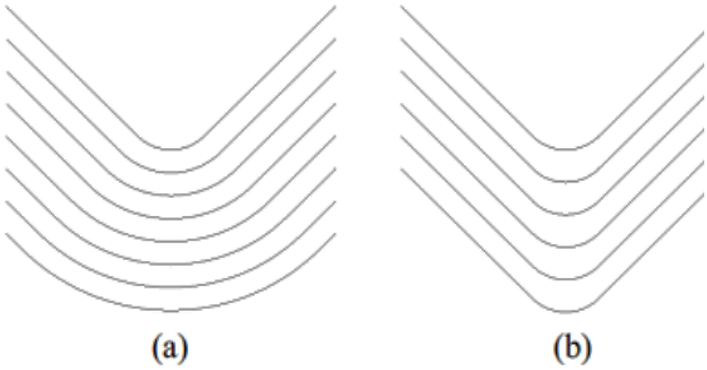


Figure 2.8: Tow path definition generated using (a) parallel method and (b) shifted method [77]

Brasington et al. highlighted some of the key differences between layup strategies and their impact on defect distribution in [20]. A rosette parallel layup strategy will lead to a reduction in total gap and overlap defect area at the cost of increased in angle deviation (Figure 2.9). Similarly, different starting point locations will result in courses being laid on different parts of the tool surface, driving distinct defect distributions. Brasington et al. conclude by highlighting the challenge and complexity of identifying a strategy that achieves both optimal mechanical performance and manufacturing efficiency.

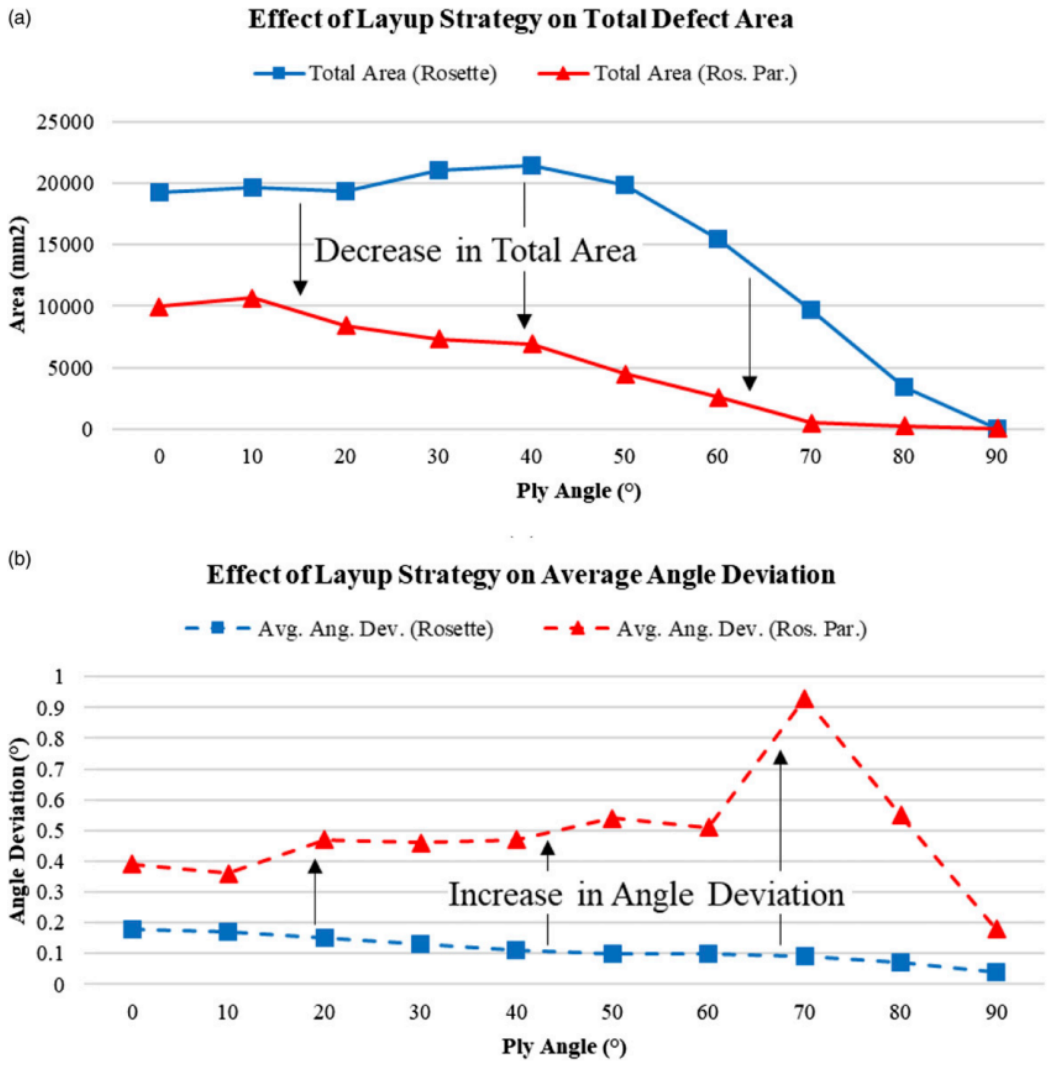


Figure 2.9: Defect distributions for plies at different angles generated through different layup strategies. (a) Gap and overlap defects. (b) Angle deviation [20]

2.1.4 AFP Process Planning Summary

Process planning for AFP is a complicated task involving a constant balancing act between competing interests. At a high level there is the balance between the design definition and available manufacturing resources. Within a smaller scope there exists many balances between defect distributions, machine efficiency, and downstream effects. One of the principal impacts of the decisions made by a process planner is the presence, location, and distribution of manufacturing-induced defects. These defects can have a large effect on the mechanical performance and geometrical tolerances of the laminate. Many of these defects are generated by the decisions made during toolpath generation. While it is impossible to completely avoid these defects, the process planner does exercise a decent amount of control over what type and in what location defects occur [20]. This is not an easy task to achieve however and relies heavily on the experience of the process planner. To alleviate this requirement, as well as automate and optimize certain process planning decisions, the CAPP module was created.

2.2 COMPUTER AIDED PROCESS PLANNING

The CAPP module (see Figure 2.10 for the interface) provides additional analysis and optimization tools for process planners specifically during the toolpath generation phase [19]. A process planner will import an .STP file into CAPP, establish laminate definitions (ply boundaries, fiber angles, rosette), and make a first selection of possible starting point locations and coverage strategies for each ply. This information is then fed into CGTech's VCP which generates toolpaths and defect prediction data, which is fed back into CAPP. The user can then manually analyze this data and repeat the loop with different layup

strategies, or let the software automatically optimize these features based on user preferences of defect distributions.

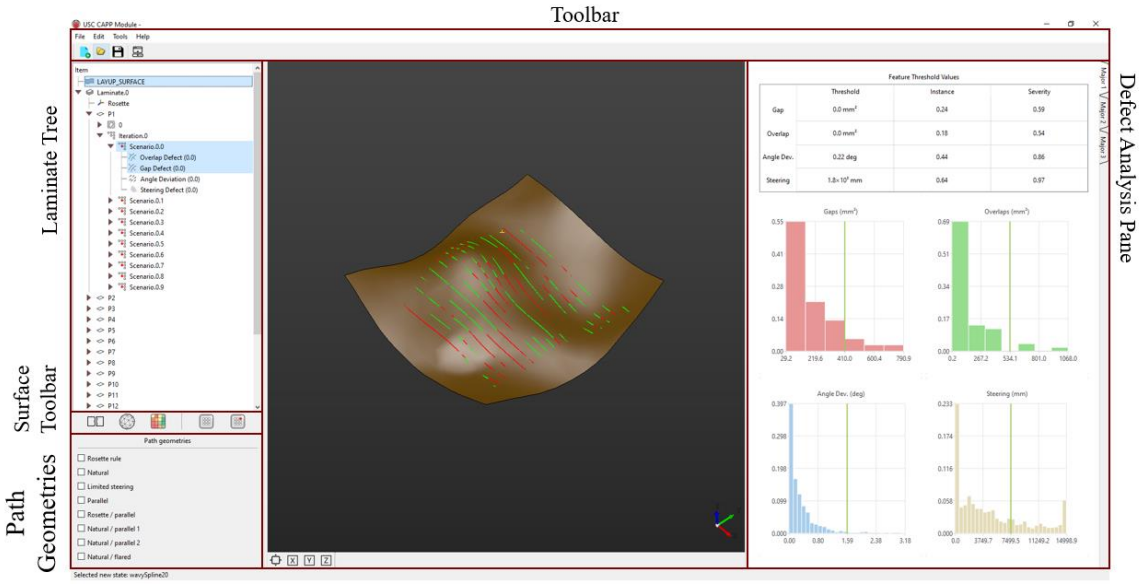


Figure 2.10: CAPP user interface

Recent efforts have sought to incorporate CAPP into a larger design for manufacturing (DFM) workflow by establishing infrastructure for communication between CAPP, VCP, and Collier Aerospace's HyperX software [23], [78], [79]. HyperX is an FEA tool that helps designers optimize the stacking sequence and ply boundaries of composite laminates based on user defined load conditions [80]. The connection of these tools allows not only for predictable manufacturing-induced defects to be fed upstream into the design phase, but also for margin of safety data to be fed downstream and influence the process planning phase. The intended workflow between these tools is shown in Figure 2.11.

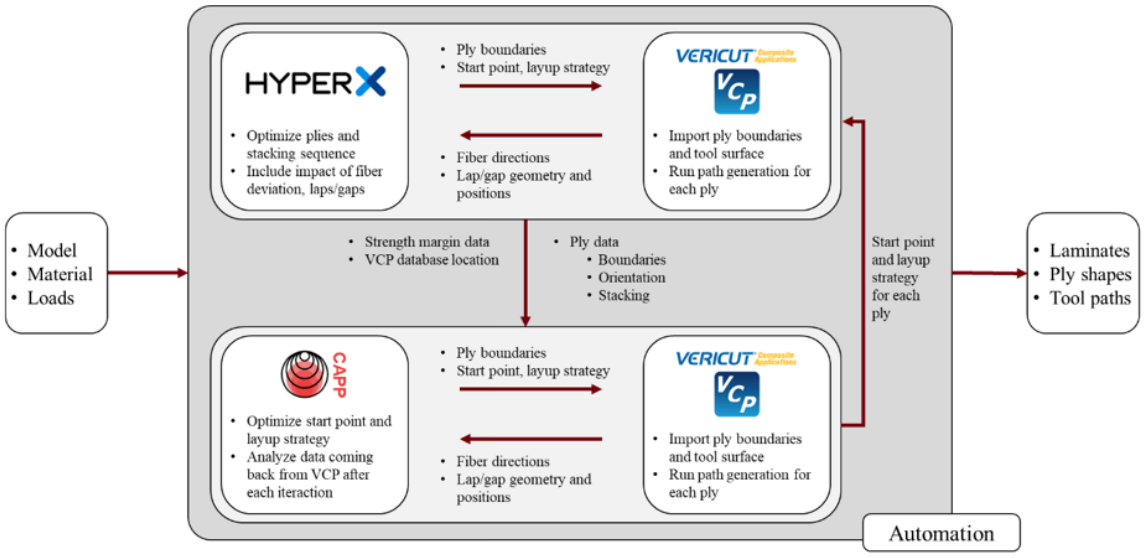


Figure 2.11: Global DFM optimization workflow

2.2.1 CAPP Ply Level Optimization

CAPP's ply level optimization (PLO) functionality focuses on identifying the optimal starting point location and coverage strategy for each ply in a laminate individually. The process starts by importing an .STP model of the tool surface and defining the laminate through creation of plies. Each ply is defined by a boundary and a fiber angle. The user will then step through three steps which prepare for the creation of the first ply scenarios. These steps include splitting the surface to identify the area of the tool surface that will be covered with material, meshing each ply, and performing a heat kernel signature which helps identify center of geometry. A ply scenario is a specific way of manufacturing a given ply, defined by the ply definition in addition to a selection of starting point location and layup strategy. The CAPP software will automatically generate an array of starting points for each coverage strategy the user would like to test for each ply in the laminate. All of these scenarios are then sent to VCP for course generation and defect prediction.

Once VCP has generated the courses and predicted toolpath-dependent defects, this data is automatically sent back to the CAPP module. The defects predicted by VCP include gaps, overlaps, angle deviation, and steering. Within CAPP users can visualize and analyze the defect data while comparing results from different starting point locations and coverage strategies. CAPP's PLO provides two metrics common to all defect types to aid in this analysis: the instance and severity scores. Both scores are customizable based on a user defined threshold that delineates acceptable and unacceptable defects, based on size for area defects (gaps and overlaps) and based on magnitude for point defects (angle deviation and steering). The instance score compares the number of defects above the threshold, deemed unacceptable, to the total number of defects in the ply scenario. The severity score compares the cumulative area or magnitude of the unacceptable defects to the total cumulative defect area or magnitude.

These scores range from zero to one where lower is better. They will vary depending on the defect distribution reported by VCP which is influenced by the layup strategy specific to each ply scenario. The two different metrics applied to the four different defect types yields eight ply scenario sub-scores. These sub-scores are combined into one final ply scenario score through the use of an analytical hierarchical process (AHP) matrix [81]. The matrix allows users to make pairwise comparisons between the eight different sub-scores reflecting their relative importance to each other. These comparisons are then used to compute rankings for each sub-score that function as weights in a weighted average which produces the final ply scenario score. The implementation of thresholds and the AHP matrix gives flexibility to the workflow. One part may be sensitive to wrinkling and require

strict steering control while another may have a critical interface and cannot have a buildup of overlaps causing a tolerance issue.

Creation of the automated VCP defect prediction loop and ply scenario scoring system enable optimization of the toolpath generation process. CAPP employs Bayesian optimization to perform this task [20]. A Gaussian process regression model is constructed through initial sampling and fitting before acquisition function optimization and infill is utilized to optimize the objective function, which in this case is the ply scenario score. Bayesian optimization was chosen due to its ability to more efficiently achieve optimality through the use of a surrogate model. PLO can continue for a set runtime or until a desired score is met for each ply. The output of this process is a set of optimized ply scenarios for each ply. These scenarios have all been optimized based on the inputs of the process planner but may differ slightly in terms of precise defect size and location.

The basis of the laminate level optimization (LLO) process is then choosing from this set of optimized ply scenarios to find the best combination thereof that yields the least amount of through-thickness defect stacking. This requires the development of infrastructure to compare defects through-thickness, creation of a metric(s) to score the magnitude of defect stacking, and implementation of an algorithm to evaluate various laminate combinations. Most laminates produced with AFP consist of anywhere from 20 to 120 plies, or more. A 100-ply laminate with 5 scenarios per ply would then have roughly 8e69 total possible combinations. This prevents a brute force search of the total design space and requires the implementation of a more efficient combinatorial optimization algorithm.

2.3 COMBINATORIAL OPTIMIZATION

The first step in optimization is problem formulation. This involves describing the problem, gathering information, defining design variables, defining the objective, and defining the constraints [82]. With the problem described and ample information gathered in the preceding sections we will advance to the latter three steps. Design variables describe the system, in this case a laminate scenario, and are typically represented by a column vector:

$$x = [x_1, x_2, x_3, \dots, x_n] \tag{2.7}$$

This vector defines a specific design where the number of variables, n, defines the dimensionality of the problem. For LLO, each variable is a ply which can vary within the discrete set of ply scenarios generated by PLO. This introduces another concept of optimization: continuity. Continuous variables are more common among engineering optimization problems and the subject of numerous algorithms [83]–[85].

The next step requires the definition of an objective function. Standard optimization objective function convention takes the form of equation 2.8 and is structured as a minimization problem [86]. For LLO the objective function will be a metric describing the magnitude of defect stacking, specifically defined in the next chapter.

$$\min [f(x)] \tag{2.8}$$

The final step of problem formulation is then to define constraints. For the purposes of this problem no constraints will be enforced. Further development could certainly add them, to check toolpaths for collisions, limit defect stacking within plies of a certain orientation, or reduce stacking within nearer plies. The goal of this optimization is to find the combination of ply scenarios that yields the least amount of through-thickness gap and

overlap stacking, so no constraints are necessary. Constraints are distinct from bounds imposed on the variables, and indeed bounds will be used. The algorithms must choose ply scenarios from the set provided and cannot suggest or create additional ones.

Optimization problems can be classified by their formulation and the definition of the objective function. Martins and Ning provide a useful tree diagram of these classifications shown in Figure 2.12 [82]. There are a multitude of algorithms employed for combinatorial optimization. Each one is best suited, or perhaps even applicable, to a problem with certain characteristics. Wolpert and Macready cemented this “no free lunch” theorem in 1996 by showing that if one algorithm outperforms another for specific cost function dynamics, then the reverse must be true for all other cost function dynamics [87]. In essence, there is no one best algorithm suitable for all problems. Different optimization algorithms applicable to LLO are discussed in the following sections.

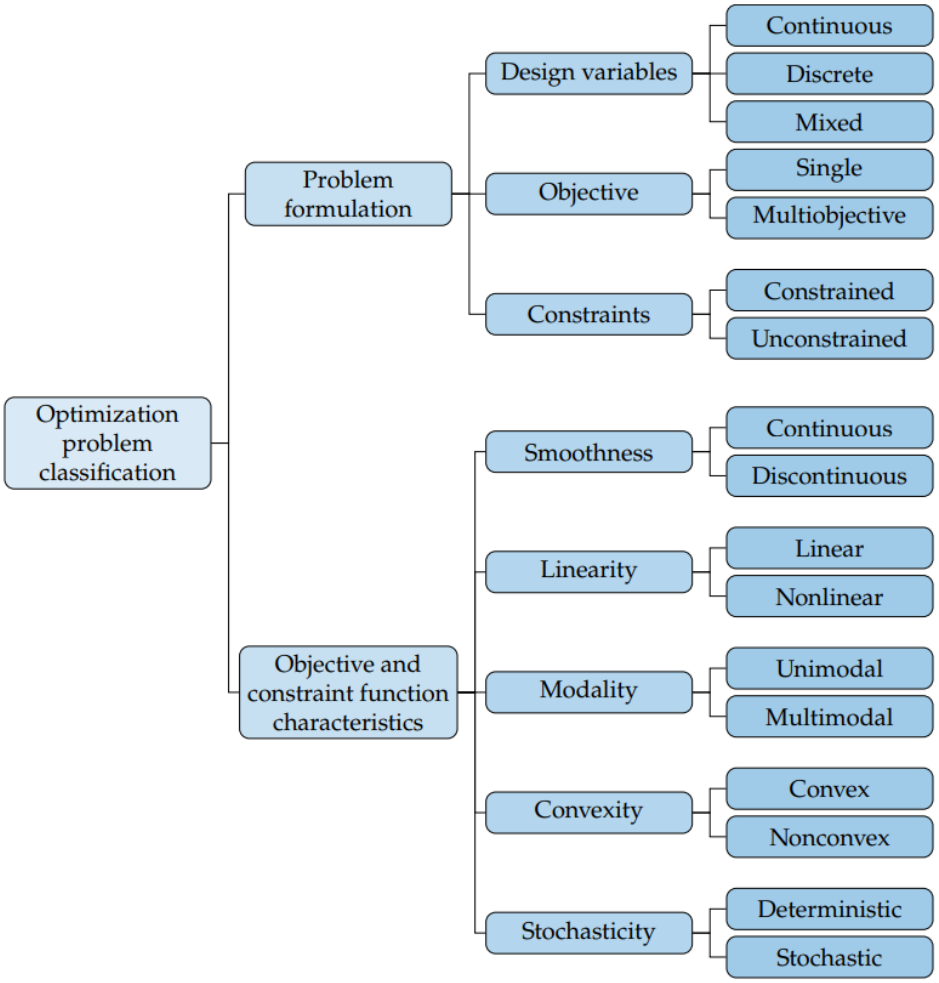


Figure 2.12: Optimization problem classification diagram [82]

2.3.1 Greedy Search

Greedy search (GS) is one of the simplest discrete optimization methods available and is often used as a starting point or control algorithm for comparison. There are many different implementations of GS, depending on the exact problem, but in general GS operates by making several locally optimal, irreversible decisions. GS is employed on problems that can be represented with a weighted directed graph or directed acyclic graph, like the one shown in Figure 2.13. The advantage of a GS algorithm is that it limits computational expense by not revisiting decisions. However, this trait also means that GS

rarely finds the globally optimal solution. GS algorithms are discussed in [88], [89]. Weng et al. developed a greedy search algorithm for optimization of a failure resistant two-phase composite part design, specifically looking at bond toughness [90]. Li et al. built a greedy based generation algorithm to create 2D random distributions of fibers and explore the representative volume element [91]. They used this algorithm to investigate transverse elastic and strength properties through finite element analysis.

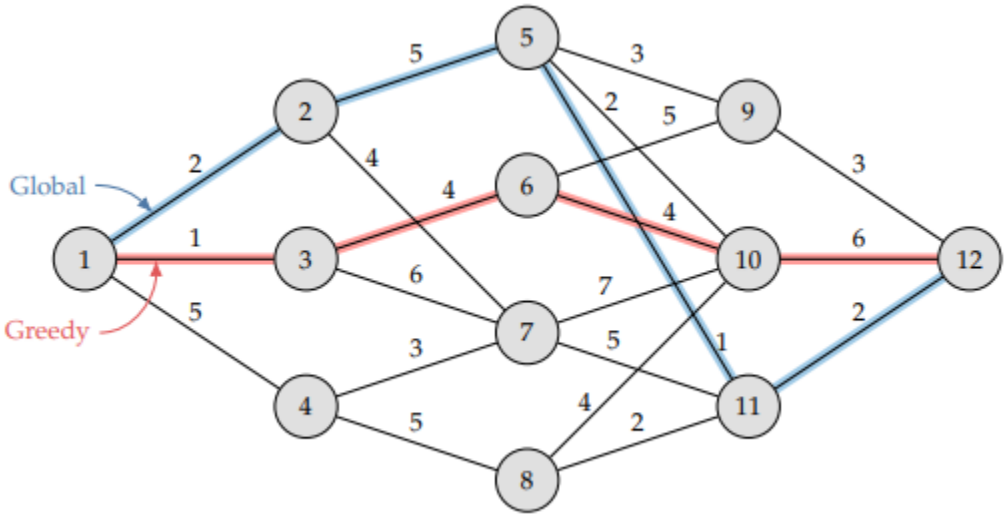


Figure 2.13: Weight directed graph showing greedy path and globally optimal path [82]

2.3.2 Heuristic Algorithms

The GS algorithm is mathematical and deterministic. Mathematical in the sense that it relies on numerical calculations based on provable mathematical principles to make decisions. Deterministic meaning it will proceed in the same way and reach the same solution given the same design space and starting conditions. On the flipside of optimization there are stochastic heuristic algorithms [92]–[94]. Stochastic is opposite of deterministic and refers to the inclusion of randomness within an algorithm. Stochastic algorithms may produce different results given the same starting conditions due to this

randomness. Heuristic algorithms are guided by rules of thumb instead of mathematical reasons. They often do not find the globally optimal solution, but rather solutions that are near-optimal. Heuristic algorithms are particularly effective when the objective function is noisy as they make less assumptions about the smoothness and modality.

In many cases, the algorithms employed are metaheuristic. A useful definition for metaheuristic was provided by Laporte and Osman, "An iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions" [95]. Metaheuristics are typically used for problems that are NP-complete or NP-hard. There are many subclassifications of metaheuristic algorithms including naturally inspired, evolutionary, population based, implicit, explicit, and direct. Of particular interest are the genetic algorithm (GA), differential evolution (DE), and particle swarm optimization (PSO) algorithms.

A GA is a naturally inspired, evolutionary, population-based, metaheuristic algorithm. It simulates the process of biological evolution through selection, crossover, and mutation phases. A given solution is called a chromosome with individual design variables labelled as genes. Selection is the process through which parent chromosomes are chosen for reproduction. Crossover involves choosing which genes will be taken from which parent to be passed down to the child chromosome. Mutation promotes further search of the design space through random changes in genes. GA's have been used extensively to optimize different facets of composite design, process planning, and manufacturing. He and Aref built a GA to optimize the design of a composite sandwich structure using a mix of discrete and continuous variables [96]. Wu and Burgueno used a GA to optimize the structural

shape and stacking sequence of a composite shell [97]. Stacking sequence optimization [98], [99] and variable stiffness local steering optimization [100]–[102] have been of particular focus for many GAs.

DE is a vector-based, evolutionary, population-based, metaheuristic algorithm, designed as a streamlined version of a GA [103]. Within DE, solutions are represented as vectors with each design variable included as a dimension of that vector. Like GAs, DEs undergo phases of crossover and mutation where vector components are either swapped or added together to perturb the solution. DE's also have a selection process which is essentially the same as the GA's. These algorithms have been adapted by many researchers to maximize fundamental frequency [104]–[106], optimize buckling performance [107], [108], and reduce error within a composite structure damage detection system [109].

PSO is a naturally inspired, population-based, implicit metaheuristic algorithm. Invented in 1995 by Kennedy and Eberhart [110], PSO is based on swarm behavior in nature. It is distinct from the GA and DE algorithms in that it does not rely on crossover or mutation, rather global communication between the swarm particles. Each particle in the swarm has a location and velocity. The location defines the design solution, with different dimensions representing the individual design variables. The velocity of each particle determines how it searches through the design space and evaluates different solutions. Each particle is drawn simultaneously toward the best location found globally by all particles and its own best location found thus far. PSO has been used frequently within literature to optimize composite strength [111]–[114], stacking sequence [115], [116], path placement [117], and automotive crashworthiness [118].

2.4 SUMMARY AND GAP ASSESSMENT

Presented in this chapter was an overview of AFP process planning with particular focus given to toolpath planning and resulting gap and overlap defects. The literature shows that different layup strategies will yield very different defect distributions depending on the part geometry and material system used. Indeed, as many as 80% of all gap and overlap defects are toolpath dependent. This opens the door not for complete elimination of these defects, but the manipulation thereof to reduce their effect. While the impact of defects is complex, researchers have found that the greatest strength reductions and thickness variations are observed when they stack through the thickness of the laminate. This conclusion led to the practice of staggering, where plies of a common fiber orientation and boundary are offset to prevent defect stacking. However, this practice does not account for stacking between plies of different angles which have shown to resist healing during cure.

To aid process planners in making toolpath generation decisions, the CAPP module was created. CAPP helps optimize and automate toolpath planning by working with VCP to create and score different layup strategies on a ply-level. Within CAPP, a specific ply-level layup strategy is defined as a ply scenario and the PLO phase will iterate through different ply scenarios until it converges on an optimal set. The scoring systems employed by PLO are customizable to user preferences, allowing for very specific results to be achieved. While PLO is helpful, the greatest effects of defects are realized on the laminate-level, emphasizing the need for an additional LLO phase.

Moving towards LLO, combinatorial optimization will be necessary. There are many steps to optimization beginning with problem formulation, where information is gathered and the design variables, objective function, and constraints are defined.

Depending on the characteristics of the optimization problem, variables, and objective function, certain algorithms will be more suitable than others. Several algorithms were introduced and discussed, with implementations within composite design and process planning highlighted.

These works lay the foundation for the work presented throughout the rest of this manuscript. While there are tools that assist process planners in selecting layup strategies to achieve certain defect distributions, they fall short of addressing the issue in a robust way. Staggering has proved effective in recent studies, but only mitigates stacking between a fraction of the total plies in a laminate. To the author's knowledge CAPP serves as the only automated optimization tool for path planning but is limited in scope to the ply-level. With the stakes being as high as up to 20% reduction in strength and 10% variation in laminate thickness, the need for advanced laminate-level path planning optimization is apparent.

CHAPTER 3

OPTIMIZATION METHODOLOGIES

This chapter covers the methods employed to prepare the defect data, score defect stacking within a laminate, and search through different combinations to identify the optimal laminate level manufacturing strategy.

3.1 DEFECT DISCRETIZATION

Due to the scale and complexity of comparing defects in 3D space, LLO utilizes an alternative method which discretizes defects into a common 2D domain. Comparisons can then be performed for each element in the discretized domain. The basis for the defect discretization relies on the parametric domain of the tool surface models. Since all plies are regions of the tool surface, its parametric representation serves as a common domain for inter-ply comparisons. The tool surfaces are represented through Non-Uniform Rational B-Splines (NURBS), which is a standard mathematical model for representing curves, surfaces, and solids among many computer-aided design (CAD) software packages. For creating 3D surfaces, it utilizes a 2D-to-3D mapping, $S(u, v) \rightarrow P(x, y, z)$. Additionally, the NURBS mapping can be inverted to project 3D entities back to the 2D parametrized domain of the surface.

The projection of 3D entities relative to a NURBS surface, to its parametric domain serves as the foundation for the defect discretization. The process begins with the cartesian contour representation of the defects creating a closed loop. These are generated by VCP

and imported into CAPP for analysis. The defects are then projected onto the tool surface obtaining their 2D representations in the tool surface's parametric domain. The parametric domain is then subdivided into a regular rectangular grid for the discretization step. The area enclosed within the projected defect polygon is mapped to this rectangular grid. The final discretized defects are represented through the on/off cells within the grid. This process is summarized graphically in Figure 3.1. With the newly parametrically discretized defects, the evaluation of stacked defects can be performed efficiently for laminates of varying size and varying ply boundary definitions.

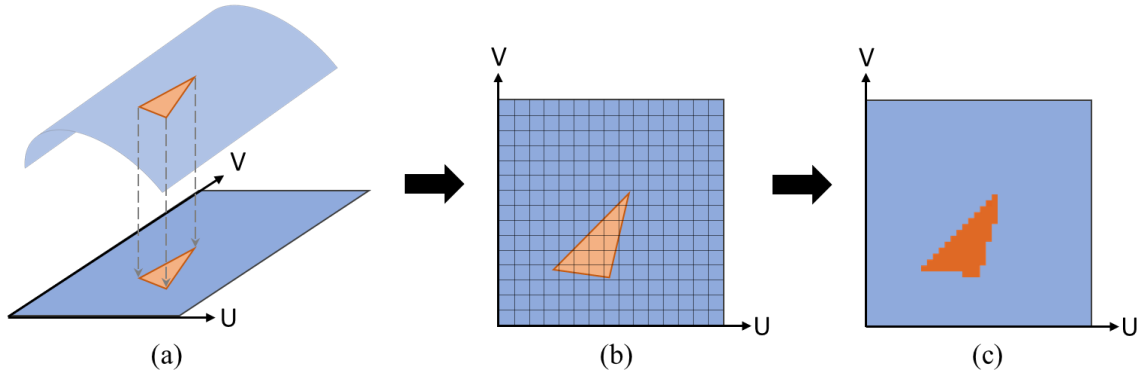


Figure 3.1: (a) Defect projection with NURBS surface; (b)-(c) discretization of projected defect

3.1.1 Defect Stacking Levels

Once the plies have been discretized, they are stored as binary arrays where 1's represent areas on the ply that have a defect and 0's represent defect free area. These ply-level binary arrays can then be mapped back to the surface for visualization or used for efficient through thickness comparison. The arrays are treated differently depending on the scale of comparison desired, but in general they are stacked on top of each other creating a 3D array modelling the entire defect distribution within a laminate. Summing through the third dimension of this array adds all of the defects through-thickness and reveals areas on

the tool surface which are subject to high defect stacking. This process is summarized in Figure 3.2.

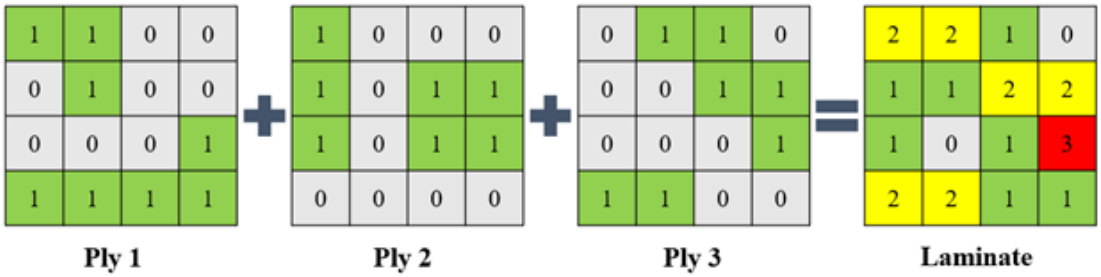


Figure 3.2: Illustration of the process for identifying defect levels

The measurement and scoring of stacked defects rely principally on the levels of defect stacking. A level describes the degree of defect stacking, where level one refers to all areas of the tool surface which have one defect through the thickness of a laminate and no defect stacking. Level two then denotes the areas on the tool surface which have two defects stacked through-thickness. In Figure 3.2 level one is shaded green, level two is shaded yellow, and level three is shaded red. This extends for a maximum possible number of levels equal to the number of plies in the laminate. When the 3D defect array is summed through-thickness to reveal the levels of defect stacking, information regarding distance between stacked defects is lost. With this representation, stacked defects between neighboring plies are treated identically to stacked defects separated by numerous plies.

3.1.2 Discretization Error Analysis

When discretizing a continuous space into discrete cells, some error is guaranteed to be produced. The magnitude of such an error, and the best practices to minimize it are therefore important. When initially discretizing the surface, the number of cells is based on the size of the parametric space. The parametric space is often in the bounds of the lengths of the surface, similar to parameterizing a curve by its length. Each cell of the 2D

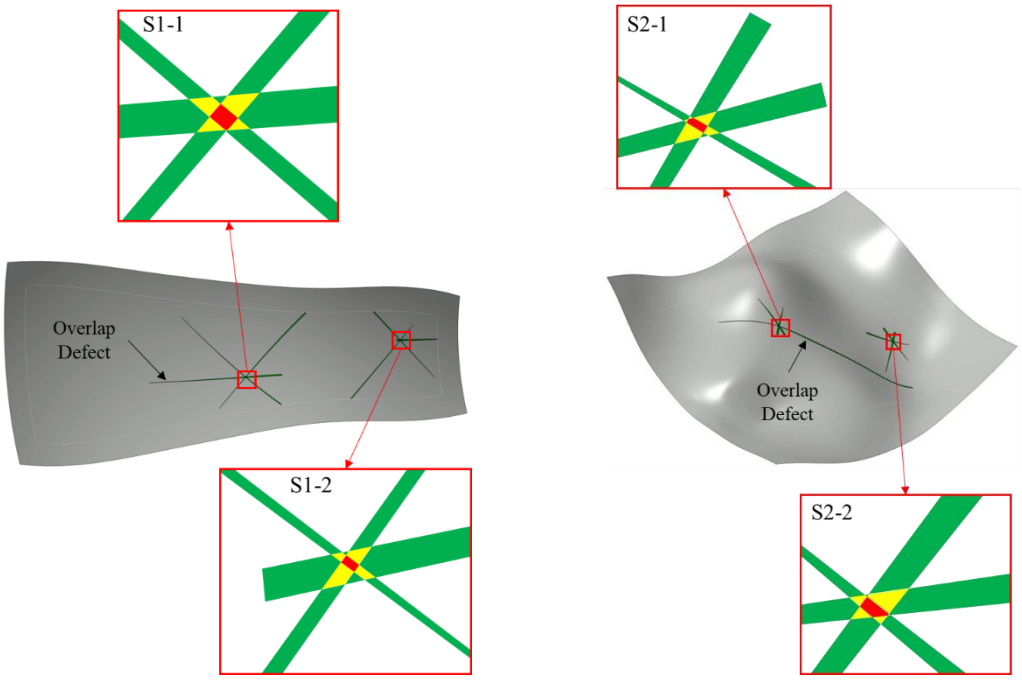
representation has some finite area on the 3D tool surface. This area will be used in the analysis method discussed in a later section. Two methods have been developed to calculate this area. The first performs discrete integrals of the parametric space with the total area of a given set of cells given by:

$$A = \iint_D |\boldsymbol{x}_u \times \boldsymbol{x}_v| dA \tag{3.1}$$

where $\boldsymbol{x}_u = \partial \boldsymbol{x}/\partial u$, $\boldsymbol{x}_v = \partial \boldsymbol{x}/\partial v$, and D is the domain defined by the cells being integrated over. This method requires the integral to be performed for every cell in the domain, resulting in numerous computations. However, an assumption can be made that all cells in the discretized domain have the same area, if the area of each cell is small enough to negate the warpage error. This assumption is based on the fact that as the cells become smaller, the area of each converges to nearly the same value. Rather than computing integrals, this method just sums the number of cells in the domain and multiplies it by the approximate area of a cell. The computations for this method are much quicker than the previous.

Based on these approaches for area calculations, we can check the accuracy and computation time for each to determine validity of the discretization approach. The accuracy will be investigated based on three parameters affecting the size and shape of the discretized domain. The first two, a and b, affect the aspect ratio based on the initial size of the domain. For example, if the initial size is 100 cells by 200 cells, the resulting size would be $a *100$ by $b *200$ where a and b are stretching the domain in the vertical and horizontal directions. The last parameter, c, affects the overall density of the discretized domain. Using the same example from before, the resulting size would be $a * c *100$ by

$b * c *200$. A higher value of c means more cells in the vertical and horizontal directions, and therefore a higher overall density. Two tool surfaces will be used to calculate the discretized area, with each tool surface being investigated with defects at two locations. The tool surfaces and defect locations are shown in Figure 3.3.



(a) (b)

Figure 3.3: (a) Doubly curved and (b) wavy spline tool surface with defects

Figure 3.4 presents the error due to changing the aspect ratio inputs (a and b). Note that the legend of this chart refers to the identifiers shown in Figure 3.3. The figure shows that using the default parametric space aspect ratio ($a/b = 1$) is the best option for minimizing the discretization error.

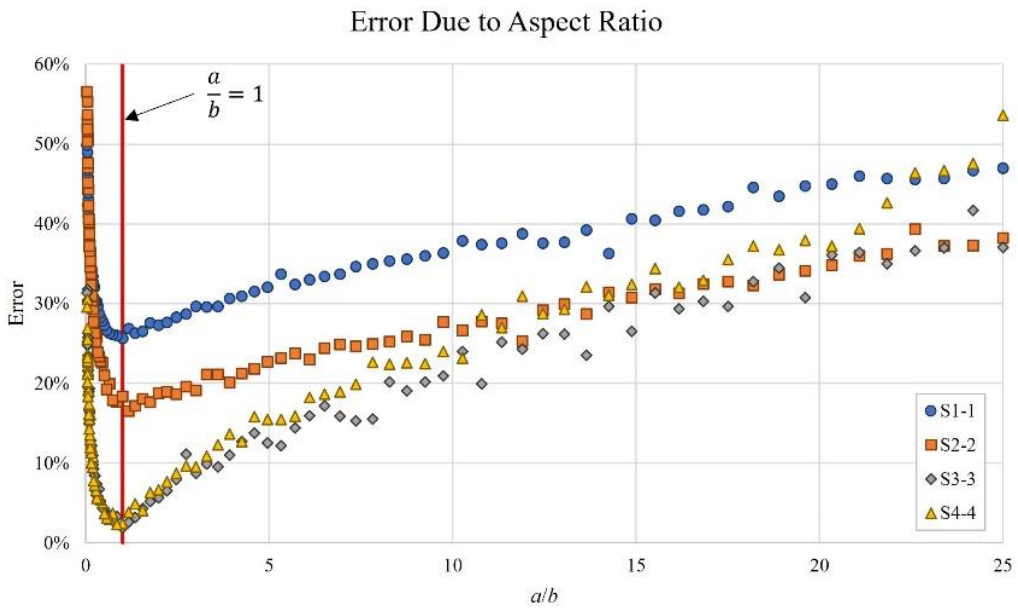


Figure 3.4: Plot of error due to aspect ratio

The error seen in this figure, and the remaining ones, is attributed to a combination of discretization and warpage in the parametric space. The discretization of the cells results in some cells overestimating and others underestimating due to the tessellation of the defect boundary (see Figure 3.5). However, a majority of the error is due to the warping seen in parametric space. This can either be due to the conversion of a curved surface into a flat representation, or the forced warpage presented in Figure 3.4. A higher warpage in parametric space results in less accurate cell representation, and therefore higher error. For the purposes of the later analysis, this error is not detrimental as the analysis is comparing areas on the same tool surface. Therefore, the errors will be relatively similar between the comparisons of defect stacking. This results in a valid minimization of stacked defect area; however, the exact amount will have to be analyzed considering this error.

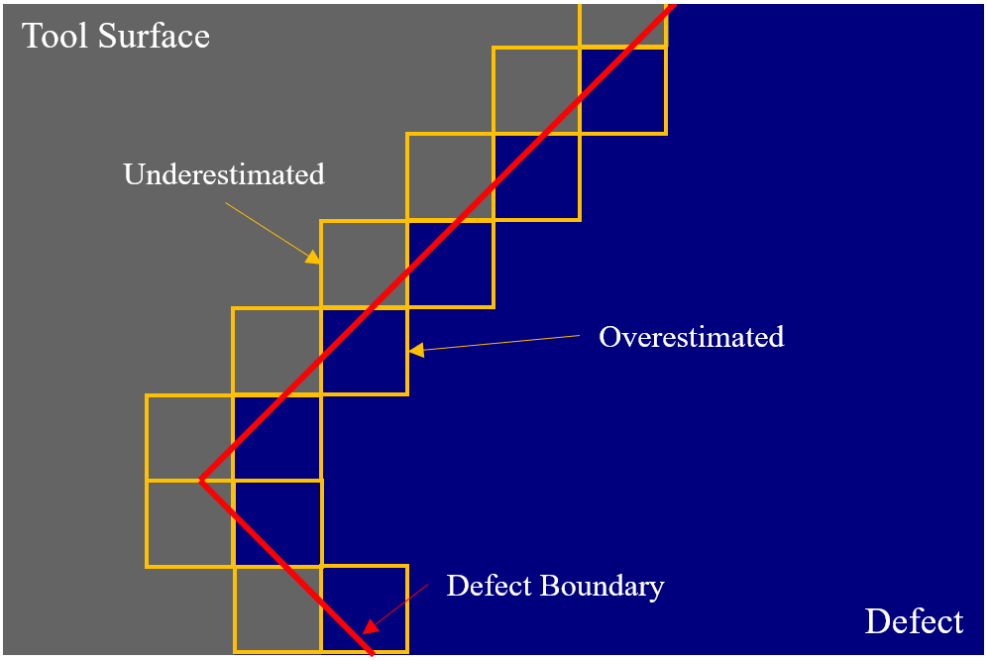
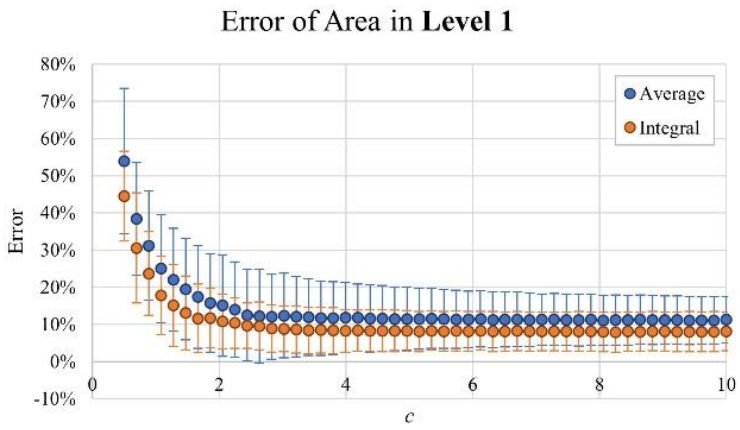
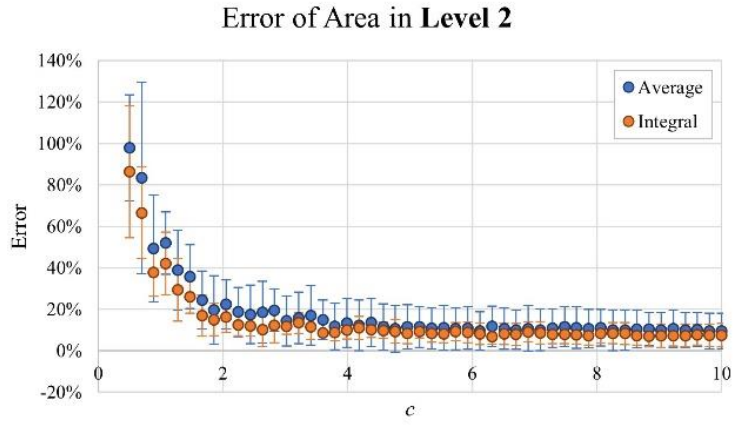


Figure 3.5: Example of source of discretization error

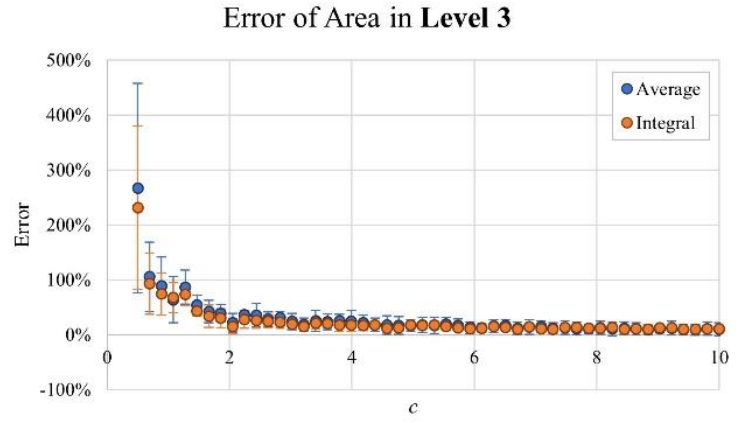
As previously mentioned, two methods have been developed for calculating area where one performs numerical integrals and the other assumes an average area for all cells. Figure 3.6 shows the comparison of error between the two methods for each defect level as a function of the density parameter c. Level 1, 2, and 3 are shown with green, yellow, and red respectively in Figure 3.3. The areas presented in the figure are averaged between the four defects across the two tool surfaces. The results demonstrate, as expected, a higher density of cells leads to a reduction in overall error. As each cell gets smaller, the area they cover also decreases, and therefore the error. This is attributed to a more exact tessellation of the defect boundary and less warpage per cell. A higher standard deviation is also seen for values of $c \leq 1$. This is again attributed to cell size where the larger cells are contributing significantly to the increased error.



(a)



(b)



(c)

Figure 3.6: Error in stacking levels (a) 1, (b) 2, and (c) 3 as a function of c

Lastly, it is key to consider the computational efficiency of each method. Figure 3.7 presents a comparison in computation times as a function of the cell density between the two methods. It is clear that as the cell density increases, the computational time for the integral approach grows significantly quicker than the averaging approach. This is even more severe when considering a higher number of plies since more defects will be present.

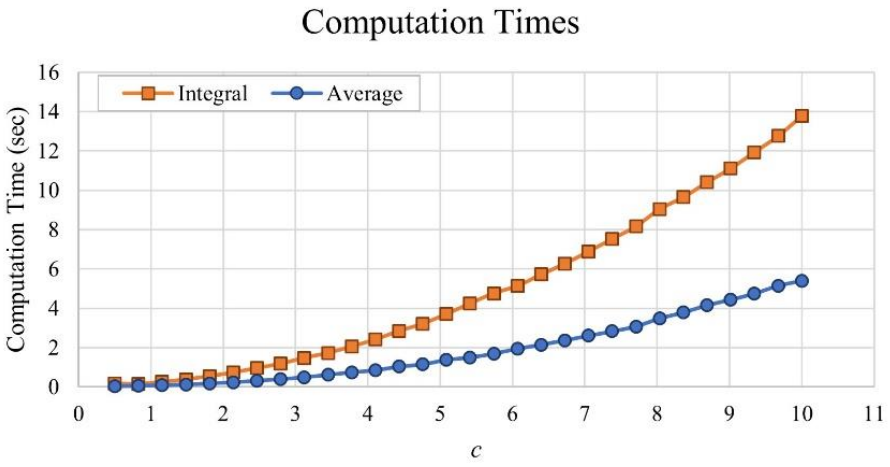


Figure 3.7: Comparison of computation times for the two methods as a function of c

3.2 OBJECTIVE FUNCTIONS

Two different objective functions were created to score the magnitude of defect stacking within a given laminate. The first, called global score, utilizes the defect data in a 2D format. As previously described, this is achieved by stacking each ply's defect array on top of each other and summing through the third dimension. The result is a 2D array with values ranging from zero, representing a region of the laminate where no defects are present through-thickness, to a maximum possible value equal to the total number of plies in the laminate, representing the case where every ply contains a defect in that region. A level represents all of the cells of a particular value in this 2D array, or in other words, the total

surface area of the tool surface with a specific number of defects stacked through the thickness. Each level is mutually exclusive. The second objective function, called local score, utilizes the 3D array created by stacking all of the 2D ply-level arrays directly. By using the 3D array, the score is sensitive to how far apart through the thickness of the laminate the defects are.

3.2.1 Global Scoring Method

The global score is comprised of two sub-scores: frequency and severity. Both sub-scores incorporate a thresholding feature which is defined by the process planner. The threshold defines the level of defect stacking at which the process planner is concerned and would prefer not to reach if possible. Frequency score is used to describe the total stacked defect area above the chosen threshold in comparison to the total defect area. Starting at the threshold level, the defect level area is summed and then divided by the total defect area. This is mathematically defined by:

$$Frequency = \frac{\sum_{threshold}^{n_{levels}} A_{level}}{\sum_{0}^{n_{levels}} A_{level}} \tag{3.2}$$

Here, n_{levels} is the total number of defect levels in the laminate and A_{level} is the area of the level. A frequency score of one denotes the case where all defects are stacked to a level beyond the chosen threshold. If the number of defect levels falls below the threshold, a frequency score of zero is defaulted.

The second metric is the severity score which is similar to the frequency score in that it compares defect levels above the threshold to total defect level area. However, with this metric, laminates with more area further above the threshold have a higher severity than

those with more area closer to the threshold. The severity sub-score is defined mathematically as:

$$Severity = \frac{\sum_{threshold}^{n_{levels}} (Level - Threshold + 1)^2 * A_{level}}{\sum_{0}^{n_{levels}} (Level + 1)^2 * A_{level}} \quad (3.3)$$

The main addition here is the squared term in the numerator which attempts to weight the score by how far it is from the threshold and results in a quadratic behavior. This is balanced similarly in the denominator of the fraction. The purpose of the severity score is to describe the distribution of the stacked defect area beyond the threshold. Two laminate scenarios could have the same frequency score if they have the same amount of stacked area above the threshold regardless of if that defect area exists in the level immediately above the threshold or in the highest possible level. This metric helps process planners identify laminate scenarios that have particularly high levels of stacking, even if the area is lesser.

3.2.2 Local Scoring Method

When optimizing for defect stacking, the goal is to achieve randomness of defects throughout the thickness of the laminate. This can be analyzed through the concept of entropy, which is a measure of uncertainty or randomness. The discretized defects from above represent a 2D image for each ply that can be stacked to form a 3D image. Measuring the entropy of such a data array falls into the field of information entropy within information theory. For an image, local entropy is related to the complexity contained in each neighborhood, typically defined by a structuring element [119]. If an image is interpreted as a sample of an L-gray-level source image, we can model the source's symbol probabilities using the gray-level histogram of the local neighborhood and generate an estimate of the source's entropy (first-order estimate) as:

$$\widetilde{H} = -\sum_{k=1}^{L} p_r(r_k) \log p_r(r_k) \tag{3.4}$$

Here $p_r(\cdot)$ represents the probability of a discrete random variable r_k which represents the gray levels of the image. The shape of the structuring element is arbitrary but needs to be a sufficient size to adequately measure local stacking. In this case, it is a rectangular cuboid of size (l, w, h). The values of each side length depend on how far a defect induces internal geometry changes within the laminate both in-plane (l and w) and through-thickness (h). Woigk et al. manufactured several laminates with different defect combinations, cut samples, and took cross-sectional scans of the laminates revealing the internal geometry [47]. Figure 3.8 shows one of these scans from a triple gap stack-up laminate with annotations measuring how far the induced fiber waviness propagates through the laminate. The number of through-thickness plies affected by the defect can be set directly as h, however l and w must be calculated based on the dimensions of the part and the resolution of the 2D discretization used.

To use this method, we need some way to drive the entropy to more randomness. This is accomplished by setting all cells with a defect to a value of 1, and all others to a uniform random value in the range [0, 255]. Therefore, low entropy zones are those with defects, and high entropy is everywhere else within a single layer. This operation is done for each ply, and the resulting 2D arrays are stacked into a 3D image. Computing the entropy on the 3D image results in areas of stacked defects (less randomness) having lower entropy. Therefore, as the entropy of the 3D image increases the dispersion of defects throughout the laminate improves. The output of the entropy function is an array of the same size as the input 3D array. This array then needs to be reduced into a single value that can be

understood by an optimizer. The single value is achieved by first averaging the output entropy array through the thickness (third dimension). A final score is found by computing the variance of the averaged entropy values. A completely random array has very low entropy variance, therefore a lower value becomes a better result. The variance is low for a random array because the entropy values are essentially the same throughout the array. However, as more defects stack up, lower areas of entropy accumulate and drive the variance up. This value can then be compared amongst others to determine the stacking of defects.

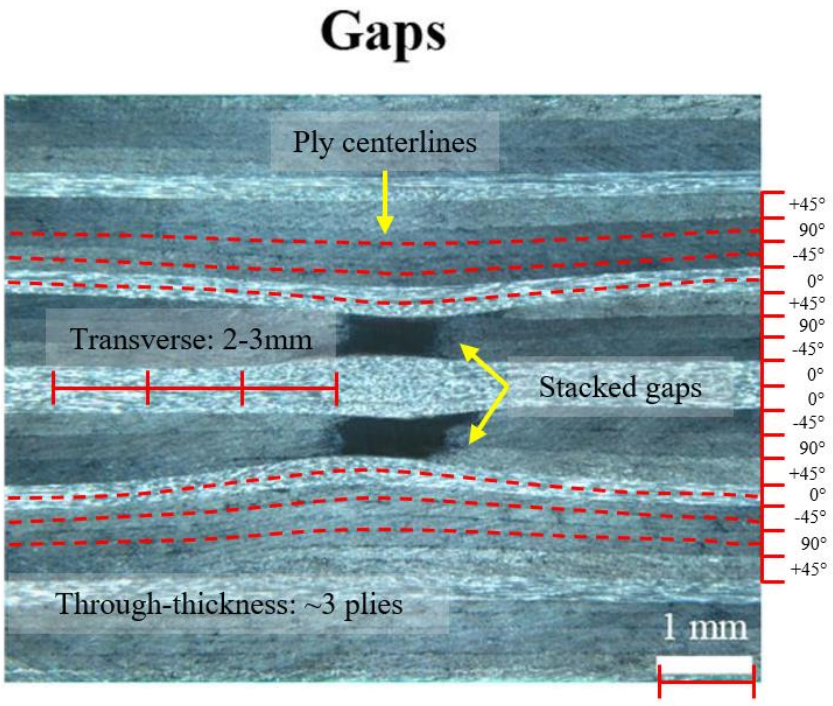


Figure 3.8: Cross-sectional view of manufactured laminate from [47] showing gap-induced out-of-plane fiber waviness propagating through the laminate

3.2.3 Gap and Overlap Score Combination

Both the global and local scoring methods score laminates based on their gap and overlap distributions separately. For the local score, the gap entropy and overlap entropy

scores are first normalized against the score of a completely random array and then combined in a weighted average where the user defines how much the final laminate score considers each defect type. For the global score, there are four sub-scores corresponding to gap frequency, gap severity, overlap frequency, and overlap severity. These scores are combined using an AHP matrix where the user defines pairwise comparisons between individual sub-scores and the matrix transforms these weights into rankings for the final combination [120], [121]. This customization allows process planners to achieve very specific defect distributions within their laminate corresponding to their own preferences and part requirements. For both objective functions a lower score is better and denotes less severe defect stacking.

3.2.4 Generalized Objective Function

The scoring metrics described are used as objective functions when coupled with the combinatorial optimization algorithms presented in the following sections. The algorithms constitute different ways of generating candidate solutions which are then evaluated using the objective function. In general, the optimization problem is then defined by:

$$\min_{x \in X} \left(f(x_1^j, x_2^j, \dots, x_n^j) \right) \tag{3.5}$$

where $f(\cdot)$ is the objective function, x_n^j is a candidate ply level scenario, n is the number of plies, $j \in [1, m]$ where m is the number of ply scenarios to be considered, and X is the collection of feasible inputs. Note that x is required to have a single ply scenario from each ply. The behavior of $f(\cdot)$ is then defined by the selection of evaluation strategy (global or local). In both cases, a lower score is better and corresponds to a laminate scenario with less defect stacking.

3.3 OPTIMIZATION ALGORITHMS

With the framework presented, it is possible to predict gap and overlap defects on a ply-level, discretize them into cells, compare them through the thickness, and score the laminate based on the severity of defect stacking both on a global and local scale. The natural next step is then to evaluate all the different manufacturing strategies for each ply based on how the induced defects stack up and choose the optimal laminate-level manufacturing strategy with minimal defect stacking. In CAPP terminology this is analogous to creating all the different combinations of ply scenarios and scoring them using global or local score to find the best laminate scenario. However, for thick laminates with many ply scenarios the combinatorial design space is prohibitively large. For *n* number of plies and *m* number of ply scenarios per ply, the total number of possible combinations is m^n. A brute force search of this space quickly becomes infeasible. Therefore, efficient combinatorial optimization algorithms are required to search through that design space and converge on an optimal manufacturing strategy.

3.3.1 Greedy Search

A greedy search is developed to provide a rapid method for optimizing the laminate. Such an approach follows the problem-solving heuristic of making locally optimal choices at each stage of the algorithm. The search creates a tree structure of the ply scenarios as shown in Figure 3.9. The levels of the tree are made up of the desired number of scenarios to be considered in each ply. The scenarios are chosen by first eliminating any that are below a given ply level score. Remaining scenarios are then sorted highest to lowest, and the top m scenarios are chosen. The optimal scenario in the root of the tree is then chosen based on the ply level scores. The next level of the tree then investigates the defect stacking

interaction between the chosen root level scenarios and the next branch according to equation 3.5. Each edge, or connection, in the tree is then assigned a weight based on the defect stacking score. The tree is then traversed to the scenario with the best score. Again, the defect stacking is evaluated for each combination. This continues for n plies resulting in the selection of ply scenarios to make up the entire laminate. The greedy search algorithm requires $(n-1)m$ objective function evaluations which is greatly reduced from searching the entire space. Also, initial iterations are much faster as they only considered a limited set of plies within the laminate. The solution guarantees a local optimum, but not necessarily a global one. Since each chosen scenario becomes fixed in the tree, it is unknown whether the selection of another scenario early in the algorithm could produce a more optimal laminate.

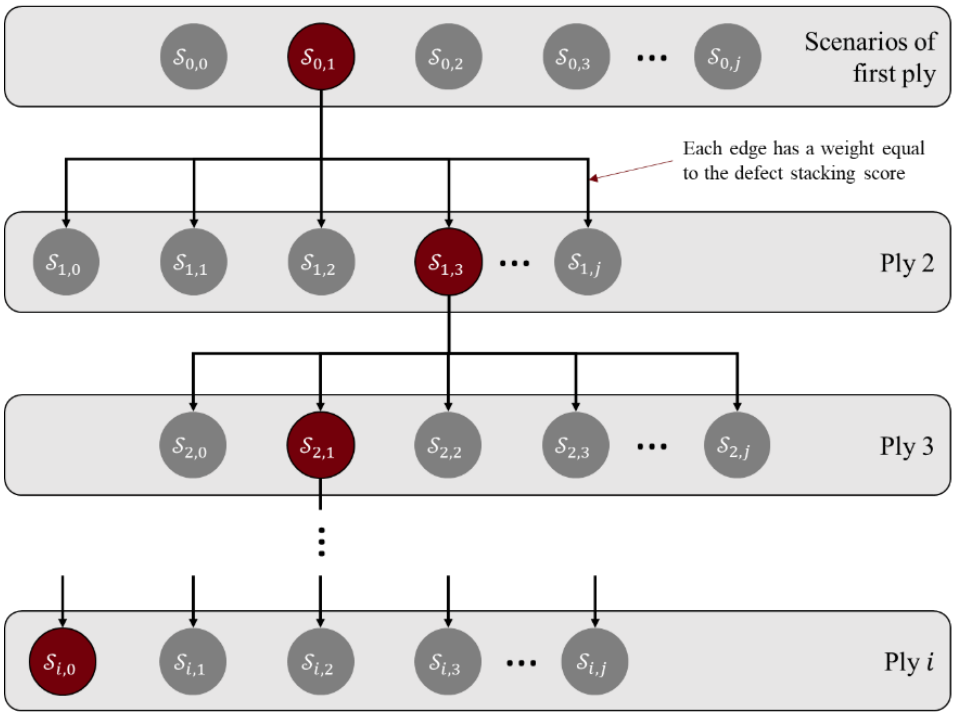


Figure 3.9: Greedy search algorithm for laminate optimization

3.3.2 Genetic Algorithm

Metaheuristic search algorithms are widely accepted as efficient approaches for handling difficult optimization problems and will be used here to try to identify a globally optimal laminate makeup. The specific algorithm to be used is a GA which is a type of evolutionary algorithm belonging to stochastic population-based evolutionary methods and uses a population of candidate solutions. The population of any generation during the GA can be expressed as a matrix P, with a length (m) equal to the population size and a width (n) equal to the number of variables. The initialization of the population matrix is accomplished by setting each value to a uniformly random real value in the range 0 to 1.

$$P = \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \cdots & v_{mn} \end{bmatrix} \tag{3.6}$$

Each row of P is a chromosome with each gene (v_{mn}) being a continuous variable in the range [0, 1] where v_{mn} denotes variable n in chromosome m. P can also be written in a shorthand notation as a list of vectors $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\}$. The continuous variables v_{mn} must be converted to either the associated real or integer value before the fitness of the population can be determined. The real and integer values are calculated with equations 3.7 and 3.8 respectively where min and max are the lower and upper bounds of v_{mn}.

$$x_n = (x_{max} - x_{min})v_{mn} + x_{min} \tag{3.7}$$

$$I_n = \lfloor (I_{max} - I_{min} + 1)v_{mn} \rfloor + I_{min} \tag{3.8}$$

The selection of chromosomes to become parents of the next generation is based on fitness and a variety of methods are available. Here, a tournament selection methodology is utilized. A tournament size of half the population size is selected, and participants are

chosen at random. The most fit of the chosen participants is chosen to be a parent. The same process is repeated to obtain the second parent.

Now mating between the two selected parents is performed using crossover and mutation methods. A uniform crossover is implemented here since it has been found to provide a larger exploration of the design space [122]. The crossover starts with generating a random binary mask with a length equal to the number of variables in each chromosome. This mask then determines the variables that the offspring will receive from the parent chromosomes. This is shown through equation 3.9 where variable v_{mn} in the child chromosome is equal to the variable from the first parent (v_{pn}) if the mask has a value of 0, and vice versa from the second parent (v_{qn}). Doing this for the total number of variables in the chromosome results in the full child chromosome.

$$v_{mn} = \begin{cases} v_{pn} & \text{if } mask_n = 0 \\ v_{qn} & \text{if } mask_n = 1 \end{cases} \tag{3.9}$$

Mutation is performed on a gene-by-gene basis for each chromosome. For each gene, a random number is selected between 0 and 1. If the random value is less than the mutation probability, the gene is mutated. For the case of this algorithm, a dynamic mutation method is used. This mutation method allows the initial generations to undergo uniform mutation (equal probability for mutation), but later generations will favor values near the current gene's value. The value of a mutated gene ($\bar{v}_{mn}$) is calculated with equation 3.10 where (r) is a randomly selected number between 0 and 1.

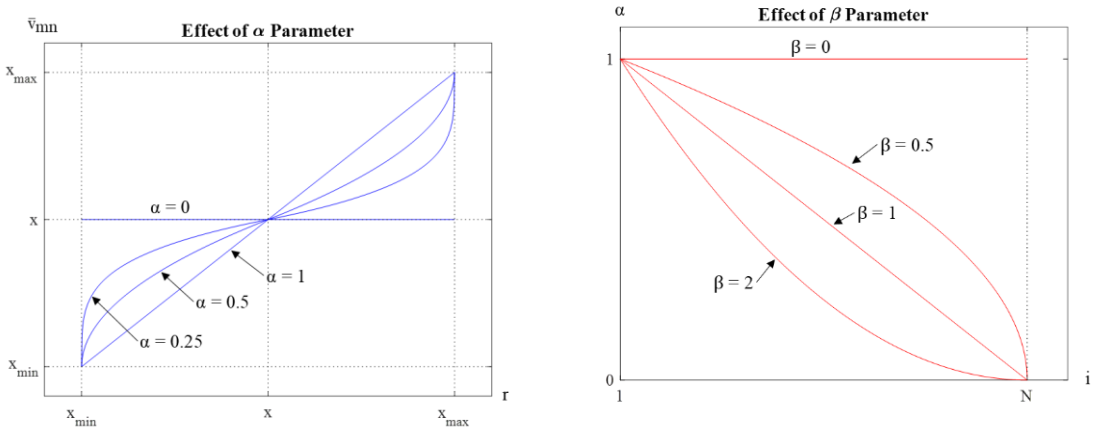
$$\bar{v}_{mn} = \begin{cases} v_{min} + (r - v_{min})^{\alpha}(v - v_{min})^{1-\alpha} & \text{if } r \leq v \\ v_{max} - (v_{max} - r)^{\alpha}(v_{max} - v)^{1-\alpha} & \text{if } r > v \end{cases} \tag{3.10}$$

The α value used in equation 3.10 is defined below in equation 3.11 where i is the current generation number, N is the total number of generations, and β is a user defined parameter [123].

$$\alpha = \left(1 - \frac{i-1}{N}\right)^{\beta} \tag{3.11}$$

Figure 3.10(a) is provided below to demonstrate how the α parameter affects the mutations and shows how the value of a mutated gene ($\bar{v}_{mn}$) changes as a function of r with various α values. An α value of one will result in uniform mutation where the gene has the same probability to mutate anywhere within the bounds. However, as α decreases the mutated gene will favor values that are closer to the current gene. The bias towards the current gene value continues to increase as α decreases, until α reaches a value of 0 where no mutation will occur.

The β term in equation 3.11 will also play a role in the α value, and therefore the expected mutation. Figure 3.10(b) shows how α values change as a function of generation number for various β values. A β value of 1 will result in α decreasing linearly until a value of 0 is reached at the final generation. If $\beta > 0$, α will decrease faster as the generations advance and vice versa for $\beta < 0$.



(a) (b)

Figure 3.10: Diagrams showing the effects of (a) the alpha and (b) beta parameters [124]

The tournament selection, crossover, and mutation processes are performed iteratively until the next population has been produced with the same population size as the current population. Once a new generation has been generated, an elitist approach is implemented. In this approach, the chromosome with the best fitness in the parent population is compared with the chromosome with the worst fitness in the child population. If the parent's fitness is greater, it replaces the child chromosome. This altered population is then used for the next generation. This process ensures that the randomness in the mating process does not result in the loss of the best design. The next generation then continues through the process presented above until a termination criterion is met.

3.3.3 Differential Evolution

DE is a type of evolutionary algorithm belonging to stochastic population-based evolutionary methods and uses a population of candidate solutions [125]. DE operates on a population of size NP and involves two stages. The first is initialization which generates an initial population $P^0 = \{\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \dots, \boldsymbol{x}_{NP}^0\}$. Evolution then progresses the population from

one generation to the next until a stopping criterion is met. During evolution, three operations are performed consisting of mutation, crossover, and selection to steer the population to an optimal solution.

The first step in the DE algorithm is to define the inputs to the optimization. This includes the number of plies N, n scenarios from each ply (parameter bounds), population size NP, differential weight $F \in [0,2]$, crossover probability $C \in [0,1]$, and maximum iterations i_{max}. An initial set of input vectors, or population, is then generated randomly using a uniform distribution in the range [0, 1]. Recall that the continuous values are decoded to their integer counterparts as described previously. These integers then correspond to a specific scenario in the given ply. Once the input vectors are defined, they are evaluated according to equation 3.5 and the best vector $\boldsymbol{x}^*$ is stored. The iteration of the algorithm then begins with mutating each vector in the population. Mutation and crossover of the rand/1/bin strategy consists of selecting three random integers (r_1, r_2, r_3) in the range $[1, NP]$ and another random integer k_{rand} in the range $[1, N]$. Each parameter in the vector is then mutated according to:

$$x_{j,k}^{i+1} = \begin{cases} x_{r_3,k}^{i} + F * (x_{r_1}^{i} - x_{r_2}^{i}) & \text{if } rand(0,1) \leq C \text{ or } k = k_{rand} \\ x_{j,k}^{i} & \text{otherwise} \end{cases} \quad (3.12)$$

where j represents a vector in the current population and k is the parameter within the vector. Such a mutation allows for more exploration of the input space since random vectors are being used to progress the solutions. This operation is illustrated in Figure 3.11. The candidate population is then evaluated and the best NP solutions are selected to create the next population. Also, the current best vector is updated if necessary. This continues until the maximum iterations, or the convergence criteria is met.

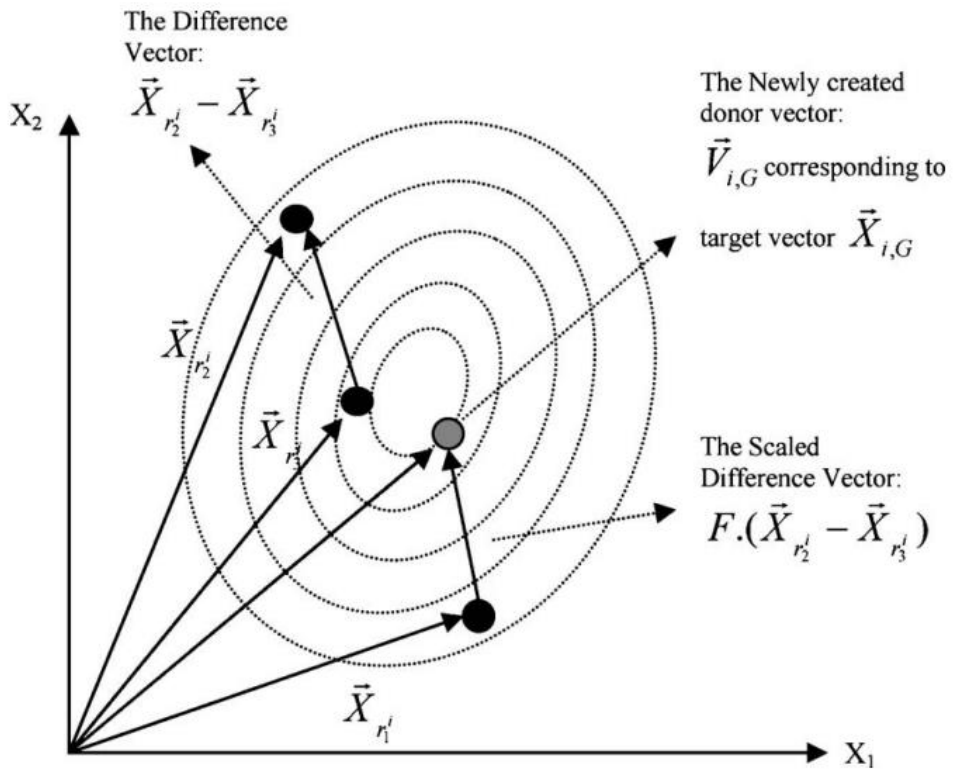


Figure 3.11: Simplified representation of the DE mutation scheme from [126]

3.3.4 Particle Swarm

Particle swarm optimization was initially developed by Kennedy and Eberhart, and it mimics the behavior of bird and fish swarms [127]. The PSO algorithm solves problems by creating a population of particles and moving them around the search space by altering their velocities and positions. Information from the particles best known position and the global best-known position help to guide the particles to the optimal solution. An example of this algorithm is provided in Figure 3.12 and Figure 3.13 where 20 particles were used to optimize Michalewicz's function in ten iterations [103]. The steps of the PSO algorithm are summarized below.

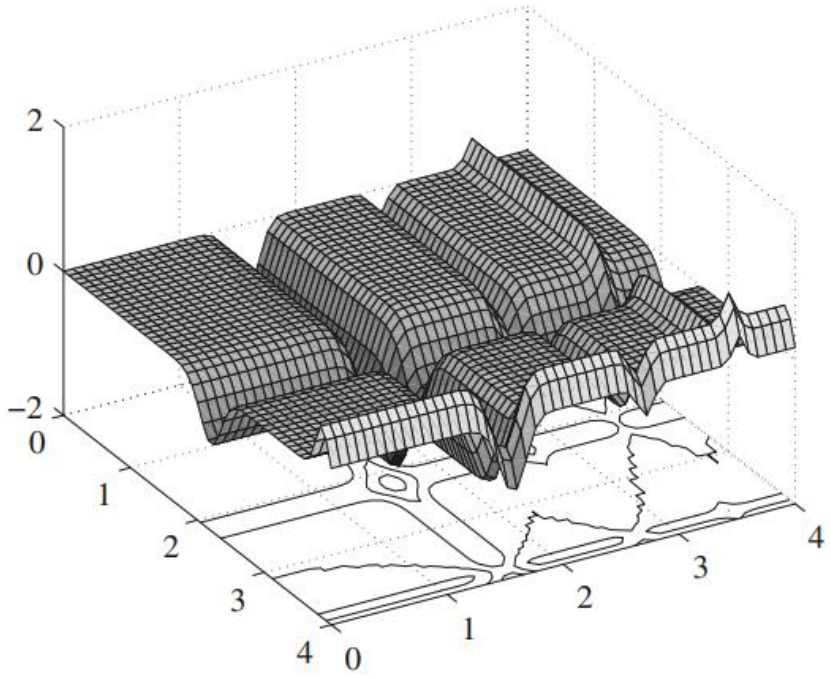
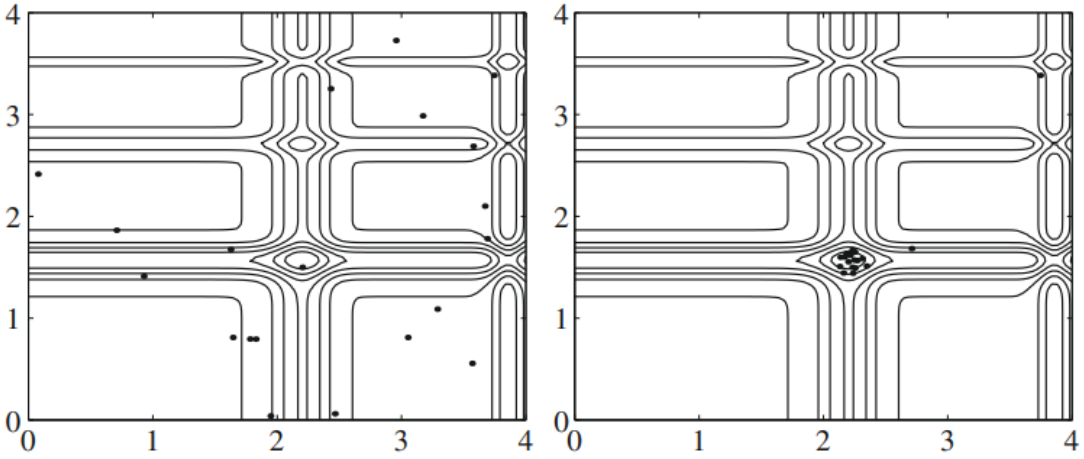


Figure 3.12: Michalewicz's function with global optimum at (2.2, 1.6) [103]



(a) (b)

Figure 3.13: (a) Initial and (b) final locations of 20 particles after ten iterations [103]

Initialization of the algorithm consists of generating a defined number of particles and randomly setting their initial positions (X_i) and velocities (V_i). At this point, a fitness function is also defined to evaluate each of the particles. Successive iterations are then

performed in which the fitness of each particle is evaluated. The particle can then determine if this is its best position along with determining if it is the global best position. After each particle in the swarm has been assessed, the velocities and positions can be updated with equations 3.13 and 3.14 [128], [129].

$$V_i(t+1) = wV_i(t) + c_1 r_1\big(pbest_i(t) - X_i(t)\big) + c_2 r_2(gbest(t) - X_i(t)) \quad (3.13)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (3.141)$$

In these equations, w is an inertial weight parameter, r_1 and r_2 are random numbers in the range (0, 1), c_1 and c_2 are acceleration coefficients, $pbest$ is the particles best position, and $gbest$ is the global best position. This process is repeated until some convergence is met such as a maximum number of iterations, or when the particles have reached a final position.

The initial positions along with input parameters w, c_1, and c_2 will influence the accuracy of the resulting solution. Also, as the dimensions of a problem increase, the optimization will become more complex and the probability of finding the global optimum decreases.

3.3.5 Global Hot Spot Detection (GHSD)

The final optimization method presented is slightly distinct from the framework and approach discussed thus far. Due to the nature of the combinatorial optimization algorithms, many laminate scenarios must be created and evaluated to converge on the optimal strategy. All of this computation takes time, especially as the number of plies and ply scenarios increases. This creates the need for a more efficient methodology for searching through the design space, reducing the number of computations necessary to converge on a solution. GHSD is a method of scoring ply scenarios based on their

likelihood of stacking within a potential laminate. By scoring the ply scenarios individually, the process of picking the best combination of ply scenarios is greatly simplified. This metric no longer needs to be treated as an objective function and coupled with a combinatorial optimization algorithm like the global and local scores. Once all the ply scenarios are scored, the algorithm simply picks the highest scoring ply scenario from each ply to form the optimal laminate scenario.

The scoring uses the same ply-level binary defect arrays described previously and compares each of the scenarios for a given ply to all the other scenarios in the laminate. For example, the scenarios in ply 1 would be compared individually to the combination of all the ply scenarios in plies 2 through *n* where *n* is the total number of plies in the laminate. With this approach, each ply scenario is compared to the combination of all ply scenarios it could realistically interact with (i.e., scenarios within the same ply are not compared). This comparison is achieved by stacking all the defect arrays from the other plies' scenarios together, creating an array that represents where defects are likely to stack within a laminate. The combined array is then bounded to focus on areas that are stacked beyond a certain number of defects. Each ply scenario for the current ply is then compared to the bounded array and scored based on the number of common defect areas. This process is illustrated in Figure 3.14.

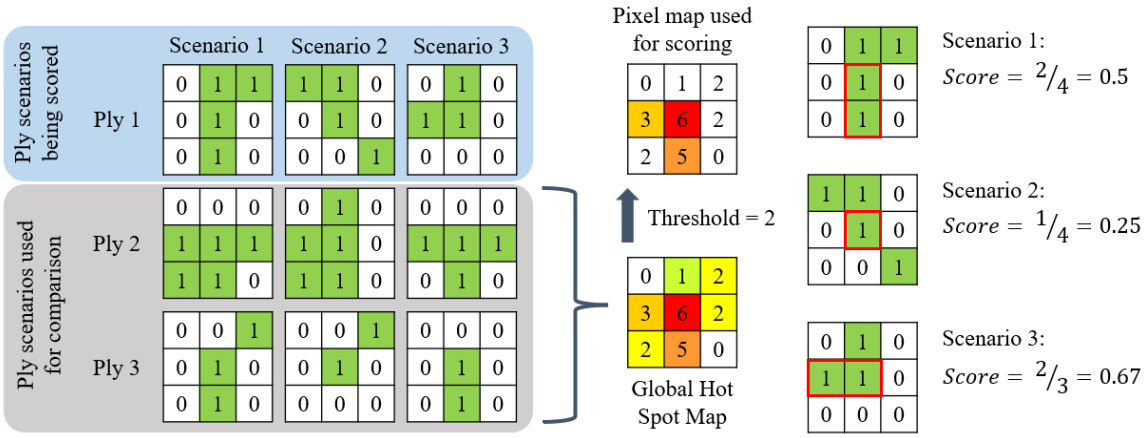


Figure 3.14: Illustration of the GHSD process

The scoring equation in Figure 3.14 is represented mathematically as:

$$Score = \frac{\sum A_{ij} \cap B_{ij}}{\sum A_{ij}} \tag{3.15}$$

Here, A_{ij} is the binary array associated with defects in a given ply scenario. B_{ij} is a global binary array where only the cells exceeding the stacking threshold are turned on. The intersection of these arrays then gives an additional binary array where a 1 represents the locations that the "on" cells overlap. Summing the binary array then gives the total number of cells with a 1 in the binary array.

For a given laminate, GHSD will create a defect hot spot array for each ply that corresponds to the possible defect locations for every other ply in the laminate. This hot spot map is then bounded by a given threshold and compared to each scenario in the current ply. Next, the ply scenarios are scored based on how many cells are common between the two maps using equation 3.15. Once all ply scenarios are scored, the best scenario from each ply is chosen to form the final laminate scenario. For the scenarios in ply 1 shown in Figure 3.14, scenario 2 would be chosen because it has the lowest stacking score. This approach is similar to frequency and severity in that it looks at defect stacking on a more

global scale. However, GHSD is much more efficient than using frequency and severity coupled with an optimization algorithm due to the drastic reduction in number of objective function evaluations. The metric is uses, which can be thought of as essentially an objective function, is also significantly simpler and faster to evaluate.

3.4 CONVERGENCE CRITERIA

Both the GS and GHSD approaches proceed sequentially, irreversibly, and deterministically therefore no convergence criteria is necessary. However, the metaheuristic algorithms (GA, DE, and PSO) proceed generationally and indefinitely, requiring the addition of a convergence criteria to terminate the algorithm. Usually, this criterion triggers when the algorithm fails to produce better solutions for a number of generations. The specific implementation can be sensitive to improvement of the best solution, average of the population, standard deviation of the population, range of the population, or factors specific to the algorithm (like average distance between particles, for example) [130]. Commonly, multiple criteria are combined to form the final convergence check. For this work, criteria applicable to all three evolutionary algorithms were desired to facilitate comparison between the performance of the algorithms.

Nine different convergence criteria have been identified that are applicable to all three evolutionary algorithms. Each criterion is sensitive to a slightly different facet of the population. While all have been employed to measure convergence, it is important to match the criteria to the specific context of the problem. All these criteria utilize one or more tunable parameters that must be adapted to the optimization problem and objective function used. Two of the criteria are combinations of multiple criteria designed to reduce the

likelihood of premature convergence. Each criterion will be evaluated in the following chapter.

CHAPTER 4

PARAMETER OPTIMIZATION

The "No Free Lunch" theorem states that there is no single optimization algorithm best suited to solve every problem, and furthermore there is no single implementation of an algorithm that is best either. The algorithms must be tuned to achieve the highest performance on a specific problem. This chapter presents a set of experiments performed to identify the optimal parameters for each of the three metaheuristic algorithms. The experiments were performed virtually, optimizing two different laminates. The first is a doubly curved tool surface called the "funnel" tool which is shown in Figure 4.1. The funnel laminate had 41 plies spread across four distinct ply boundaries. Every ply had ten ply scenarios corresponding to ten different randomly chosen starting points with each using a rosette layup strategy. The second laminate had 20 plies all with the same ply boundary laid up on a complex tool surface called the "wavy spline" tool (Figure 4.2). Similarly, each ply had ten ply scenarios with the rosette layup strategy and random starting point locations. Rosette was chosen because it produces the most gap and overlap defects. This represents a worst-case scenario and provides the most defect data for the algorithms to optimize.

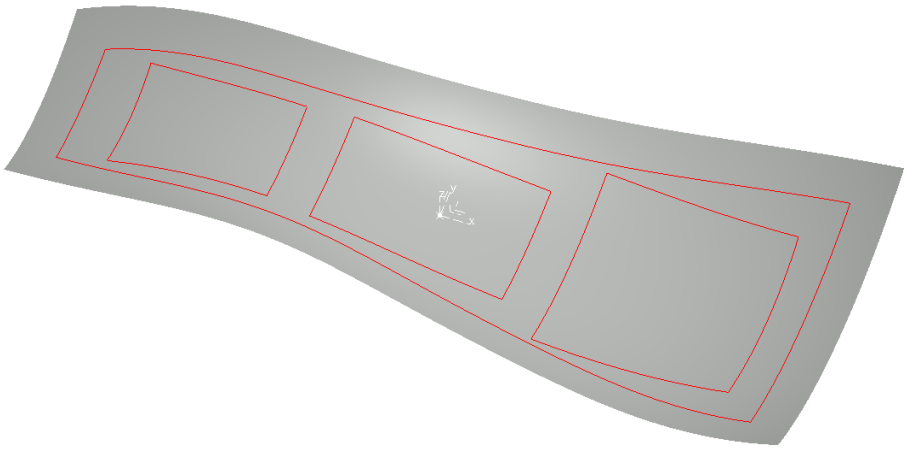


Figure 4.1: Doubly curved "funnel" tool surface with ply boundaries shown

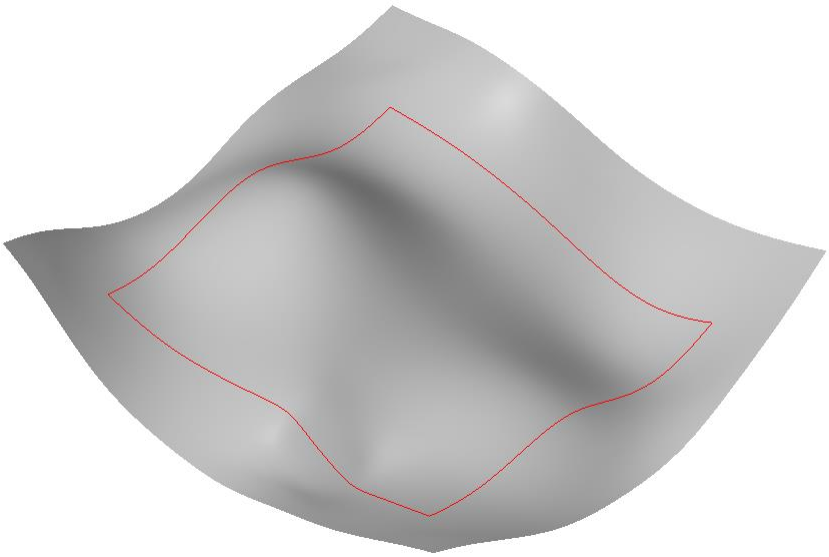


Figure 4.2: Complex curvature "wavy spline" tool surface with ply boundary shown

The experiments started by selecting baseline parameters for each algorithm and feasible ranges to evaluate. These baseline values were typical values used within literature and were theorized to provide a good starting point for the algorithms. Ten iterations of each baseline algorithm were performed on both laminates using both the global and local objective functions. Trials of ten iterations each were then performed for all of the selected

values of each parameter. Parameters were manipulated individually for the trials with all other parameters set to baseline values. A more robust methodology may include various parameter combinations, however this was deemed computationally prohibitive for this work. The baseline and experimental values tested are listed in Table *4.1*. The effect of each parameter on the algorithms' performance was evaluated through comparison of the best laminate scenario score found overall, the average of the best scores for each iteration, as well as the average and standard deviation of the entire population for each trial. If two values were close in their results, or if one extreme was close, additional values were tested to identify the best parameter values.

Table 4.1: Parameter optimization experimental values

Algorithm	Parameter	Variable	Baseline Value	Experimental Values
GA	Mutation Factor	M	0.1	0.01, 0.05, 0.075, 0.1, 0.15
	Beta Value	β	0.25	0, 0.25, 0.75, 1.5, 1.8
DE	Differential Weight	F	0.7	0.5, 1.0, 1.5, 2.0
	Crossover Factor	C	0.5	0.25, 0.5, 0.75, 1.0
PSO	Cognitive Constant	$C1$	1	1, 2, 3
	Social Constant	$C2$	2	1, 2, 3
	Inertia Constant	w	0.5	0.5, 0.7, 0.9

4.1 BASELINE OPTIMIZATION RESULTS

Each algorithm with baseline parameters was run ten times for both laminates and both objective functions. The algorithms were allowed to run for 50 generations with a population of ten candidate solutions. The final section of this chapter explores different

convergence criteria to replace the max generation limit. Figure 4.3 and Figure 4.4 show the progression of each iteration of each algorithm using global score on the funnel tool and wavy spline tool respectively. The vertical axis measures the objective function value of the best laminate scenario found by that run as it progresses. Note a lower value is better. The horizontal axis measures each generation of the algorithms. The dotted line shows the score of the laminate scenario found by GS and the dashed line shows the average score of the best laminate scenarios found by each iteration of the algorithm after 50 generations. Each red line is one iteration of the algorithm.

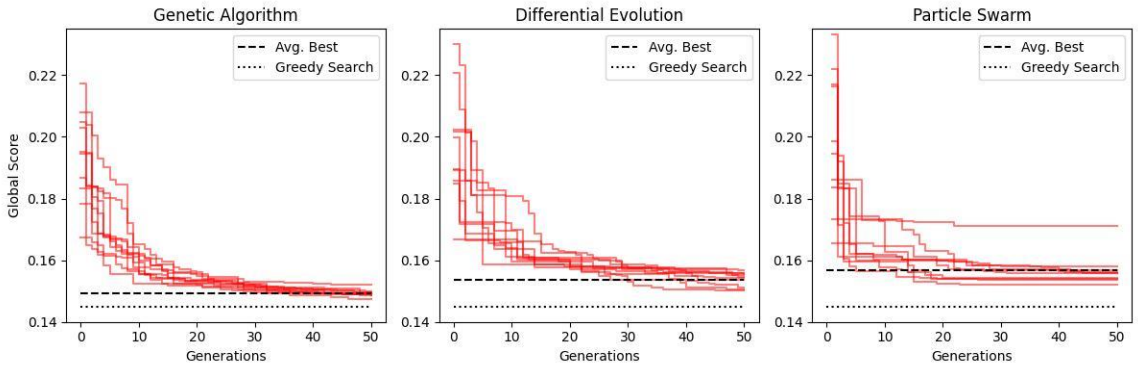


Figure 4.3: Convergence graphs for funnel tool with baseline parameters using global score

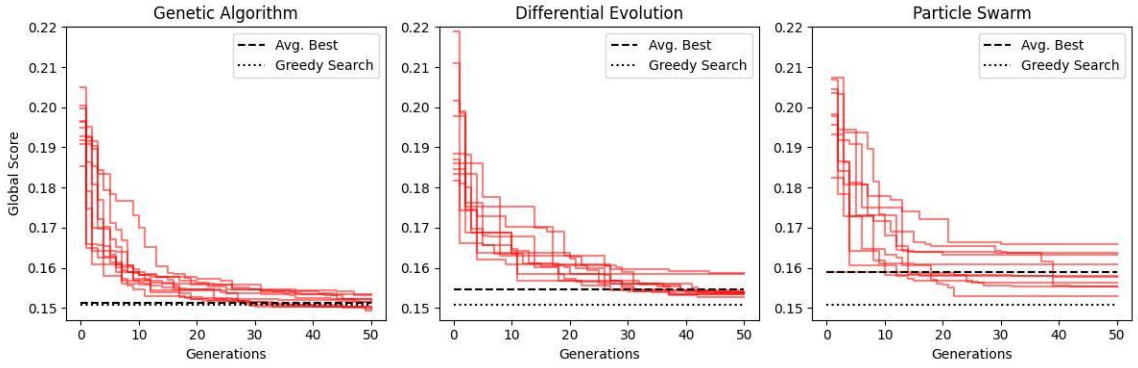


Figure 4.4: Convergence graphs for wavy spline tool with baseline parameters using global score

For the baseline global score optimization trials, it was clear that GS was outperforming all the metaheuristic algorithms. The GA came close, nearly identical on average for the wavy spline laminate, and consistently outperformed the other evolutionary algorithms. The DE and GA progressed similarly, making larger improvements early before making smaller improvements through the final generations. PSO performed the worst, in part due to premature convergence on a local minimum and rarely improving after generation 30. Results were very similar across the two laminates with slightly more variability among results and overall better performance of the algorithms for the wavy spline tool. Baseline runs of the algorithms were also performed using local score, shown in Figure 4.5 and Figure 4.6. A summary of the baseline results is provided in Table *4.2*. The scores presented in the table represent the average of the best scores found by each of the ten iterations of each algorithm.

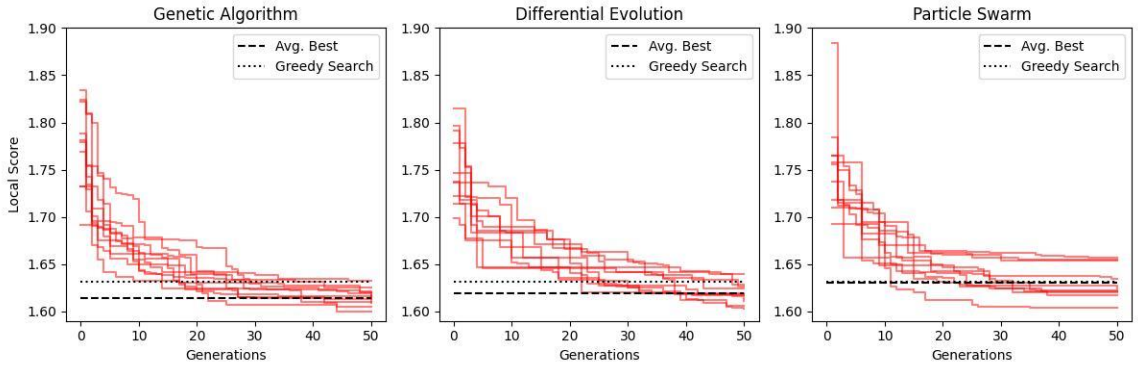


Figure 4.5: Convergence graphs for funnel tool with baseline parameters using local score

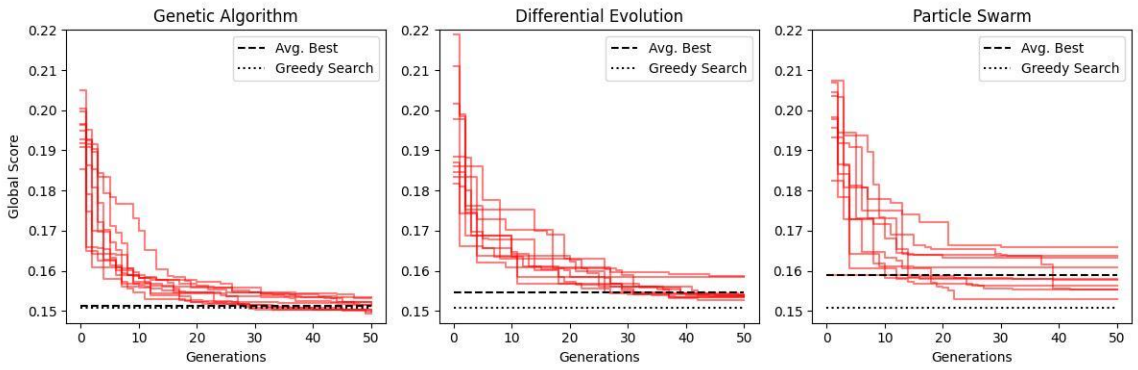


Figure 4.6: Convergence graphs for wavy spline tool baseline parameters using global score

The local score trials showed similar relationships between the metaheuristic algorithms with GA performing the best, followed closely by DE, and then PSO. However, in contrast to the global score, the algorithms performed better on the funnel tool and even outperformed GS on average. This is potentially due to an interaction between the way GS and local score operate with the laminate structure of the funnel laminate. GS proceeds sequentially, ply-by-ply, making locally optimal and irreversible decisions. The local score analyzes defect stacking within a set range of plies through-thickness. For these experiments this range was set to two plies above and below, based upon out-of-plane fiber waviness observed in the literature. The funnel laminate has four transitions between plies that don't share any common surface area and therefore cannot have defects that stack. While the GS using global score is still able to judge those plies based on earlier plies that do share surface area, the local score constricts this range and forces the GS algorithm to make a largely arbitrary decision.

Another notable outcome of the local score trials is the difference in score between the funnel and wavy spline laminates. While global score is normalized from zero to one, local score will vary depending on the size of the laminate and defect distribution. Since the

funnel tool is much larger and has regions where no defects occur, the local score is driven down and is lower on average than the wavy spline laminate. Additionally, due to the increased sensitivity of local score, there appears to be a protraction of convergence compared to the global score runs. As a result, local score may take more generations to converge.

Table 4.2: Summary of baseline parameter algorithm trials' results

	Global Score (Avg. Final)		Local Score (Avg. Final)	
Algorithm	Funnel	Wavy Spline	Funnel	Wavy Spline
GS	0.14497	0.15077	1.63140	2.25631
GA	0.14930	0.15125	1.61452	2.26930
DE	0.15372	0.15459	1.61947	2.31116
PSO	0.15674	0.15893	1.63084	2.35509

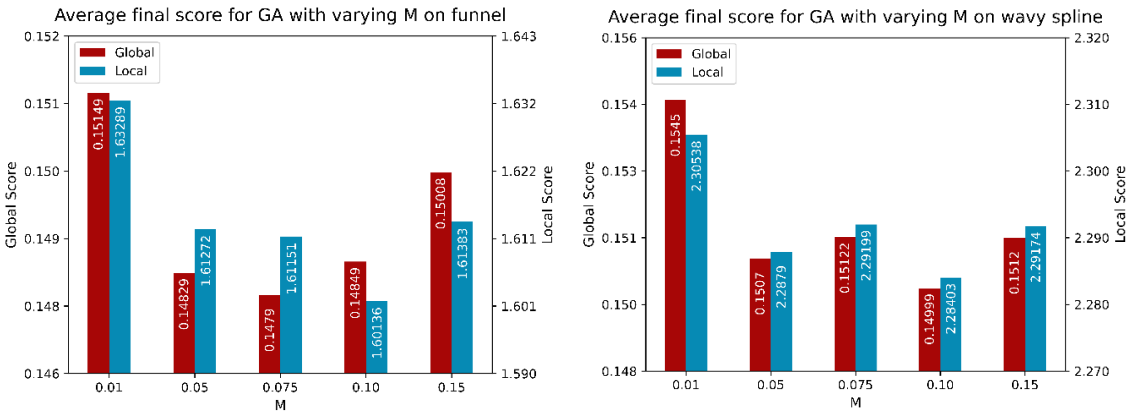
4.2 GENETIC ALGORITHM PARAMETER OPTIMIZATION

There are two parameters of interest within the GA, both relating to mutation. The mutation factor, M, controls the probability that any given gene on each chromosome will mutate and change values. This is analogous to the probability that each ply scenario will switch to another ply scenario during the mutation phase. The typical range for this parameter is 0.01-0.1 with a frequent skew towards smaller values in most applications. The second parameter is beta, β, which influences what the gene will mutate to as a function of generations. For early generations, a mutation can change the ply scenario to any within the set. As the algorithm progresses, the possible mutation values are limited to being nearer and nearer to the original value. This promotes convergence as it reduces the

variability driven by mutation. Beta values range from 0-1.8 roughly, with lower values allowing more freedom of mutation and higher values reducing that freedom.

4.2.1 Mutation Factor

Ten iterations of the GA for each value of M were performed using global and local score for both the funnel and wavy spline laminates. The results (Figure 4.7) show the final score for each trial of the GA averaged across all ten iterations. This value provides a useful metric for what the expected performance of the algorithm would be using a specific parameter. For the funnel laminate, the lowest average global score came at a mutation factor of 0.075 with local score bottoming out at a factor of 0.1. The wavy spline laminate was consistent across both scores with the optimal mutation factor coming out as 0.1. While there was some discrepancy between the funnel local and global results, three out of the four sets of trials agreed on a mutation factor of 0.1 which is the baseline value. In every set of trials, the mutation factor was directly proportional to the standard deviation and average score of the population. As M decreased, the standard deviation decreased and more of the population was identical or nearly identical to the best found so far. This brought the average score of the population down but hindered the ability of the algorithm to search, resulting in less performance overall when looking at the best laminate found. A factor of 0.1 corresponds to about one out of every ten ply scenarios being changed during the mutation phase. The magnitude of change for the ply scenarios will be affected by the beta parameter discussed next.



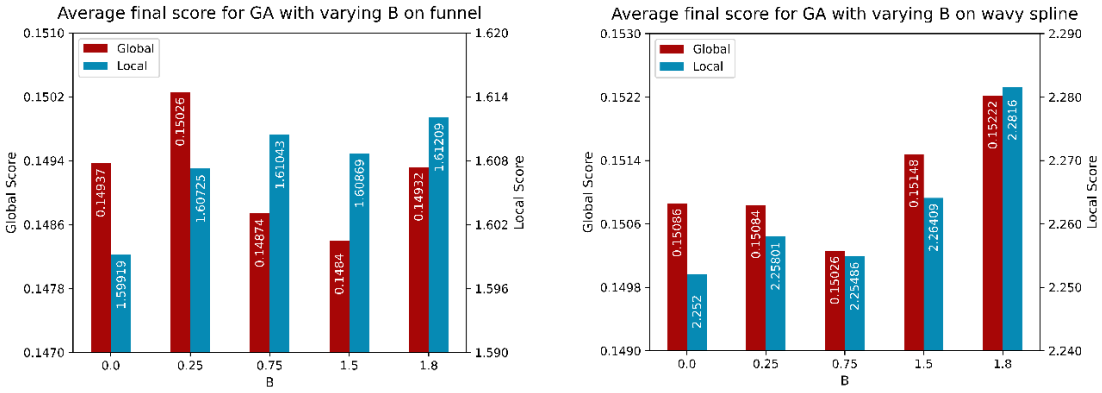
(a) (b)

Figure 4.7: Average final score for GA with varying mutation factor, M, on (a) funnel and (b) wavy spline laminates

4.2.2 Beta Value

The beta value influences the range of ply scenarios that a gene can mutate to, evolving as the algorithm progresses. Differences in beta value will change the rate at which restrictions on mutation range are imposed, with higher values limiting mutation faster. The local score trials were largely consistent across the two laminates, with both showing lowest averages at β=0. The global score trials varied across the two laminates, with the funnel laminate runs performing best at β=1.5 and the wavy spline laminate runs performing best at β=0.75. However, for all trials the difference in performance was relatively low. This makes sense given the context of the problem. The beta operator is designed to promote convergence by progressively limiting mutation. This is based upon the heuristic that nearby values are likely close in objective function performance. However, the ply scenarios and the encoded values that represent them really share no relation. Ply scenarios are encoded into genes as integer values ranging from zero to the total number of ply scenarios within the ply. The two ply scenarios coded as two and three could be vastly different, so limiting mutation as the algorithm progresses may still

promote convergence, but not necessarily performance. For this reason, a beta value of zero will be used. Figure 4.8 shows the results of the beta trials.



(a) (b)

Figure 4.8: Average final score for GA with varying beta value, β, on (a) funnel and (b) wavy spline laminates

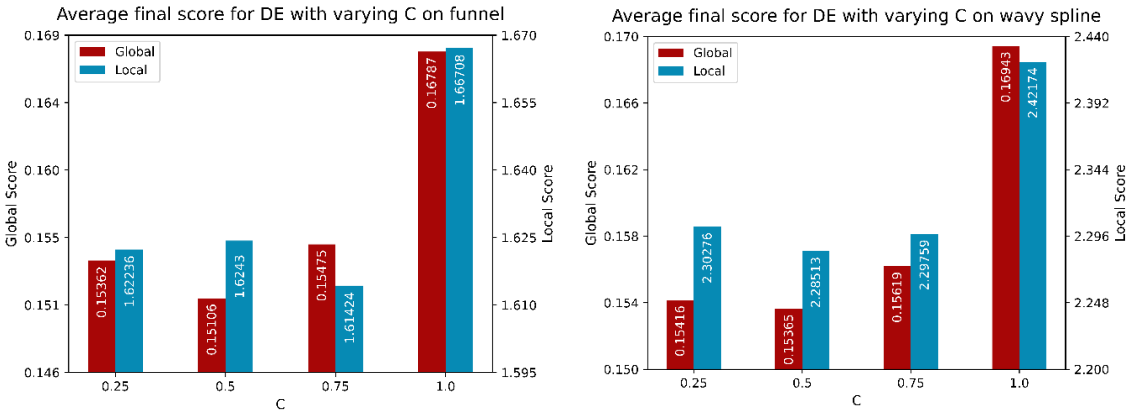
4.3 DIFFERENTIAL EVOLUTION PARAMETER OPTIMIZATION

There are two parameters of interest with the DE algorithm: crossover factor, C, and differential weight, F. The crossover factor determines the probability that crossover will occur between each variable of the candidate solution, represented as a vector, and a randomly selected donor vector. C ranges from zero to one where a value of one means there is a 100% probability of crossover. The differential weight adjusts the magnitude of perturbation during mutation. Mutation occurs in conjunction with crossover and uses the difference between two additional randomly selected vectors, multiplied by F, to promote a wider search of the design space. Values of F can realistically range from zero to two, but zero to one is more popular and typically results in a more stable algorithm.

4.3.1 Crossover Factor

The crossover factor has a large impact on the performance of DE by determining how and if crossover and mutation occur. Ten iterations of the DE algorithm were performed

for each parameter value on both laminates with both objective functions. The average final scores of each set of runs are summarized in Figure 4.9. Notably, a unanimous drop in performance was observed when $C = 1$. This parameter value forces every component of the given vector to copy that of the chosen donor vector and prevents mutation from occurring. After just a few generations the entire population converges prematurely. The global score runs were consistent with a crossover factor of 0.5 yielding optimal performance. The local score runs were split, with the funnel laminate operating better with $C = 0.75$ and the wavy spline laminate finding better results with $C = 0.5$. With three out of four sets of runs in agreement, a crossover factor value of 0.5 will be used. This matches the baseline value and means that half of all vector components will be copied from one donor while the other half will be mutated from another donor.



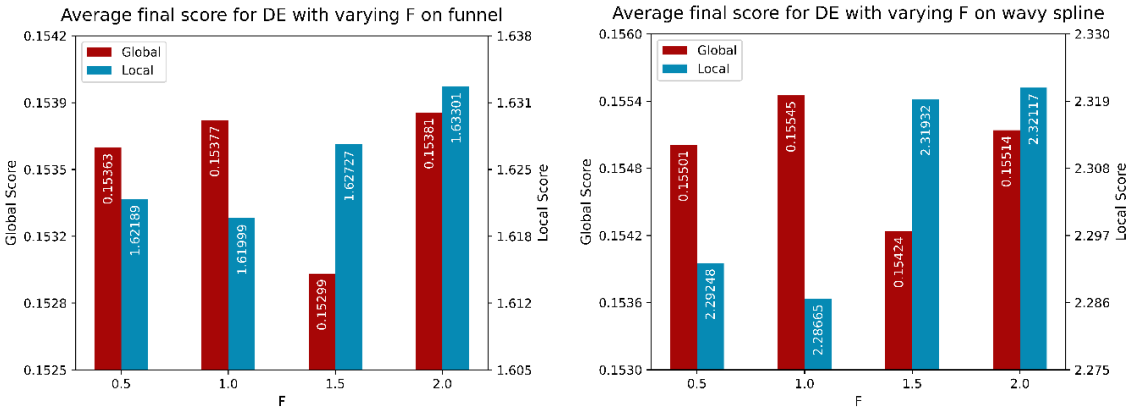
(a) (b)

Figure 4.9: Average final score for DE with varying crossover factor, C, on (a) funnel and (b) wavy spline laminates

4.3.2 Differential Weight

The differential weight is somewhat similar to the beta parameter of the GA in that it affects the magnitude of mutation, though with DE it is not generation dependent. Ten iterations of the DE were performed for each parameter value on each laminate with both

objective functions. The average final scores of each set of runs are summarized in Figure 4.10. Good consistency across the laminates was seen for both objective functions. Global score runs saw increased performance with a differential weight value of 1.5 while local score runs found better solutions on average with a value of 1.0. With the consistency across the laminates for both objective functions the parameter value will be split moving forward. The values will be $F = 1.5$ and $F = 1.0$ for global and local score respectively. The difference in result here suggests that global score may benefit from a bit more random search than local score. Higher values of differential weight mean greater perturbation during mutation which likely helps DE avoid local minimums when optimizing for global score.



(a) (b)

Figure 4.10: Average final score for DE with varying differential weight, F, on (a) funnel and (b) wavy spline laminates

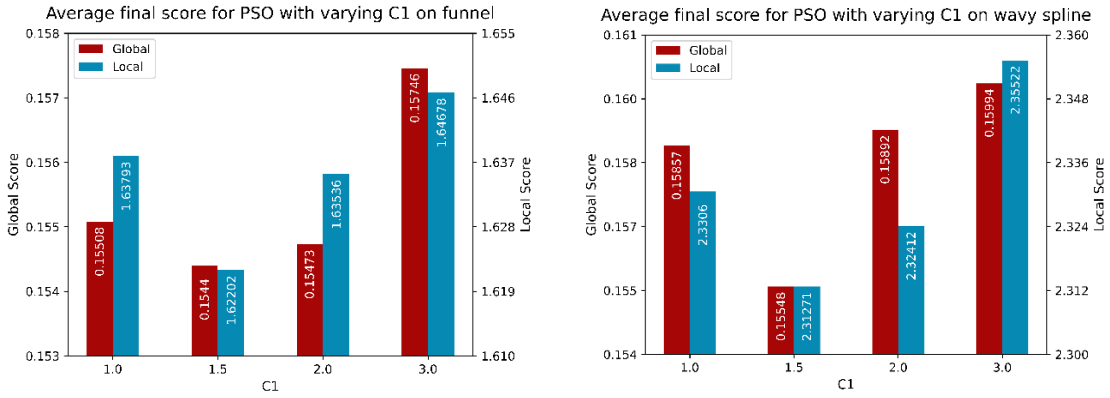
4.4 PARTICAL SWARM OPTIMIZATIO PARAMETER OPTIMIZATION

PSO contains three parameters of interest: the cognitive constant, $C1$, the social constant, $C2$, and the inertia factor, w. The cognitive constant controls the degree to which each particle's velocity is directed towards the location of the best solution it has found so far. The social constant controls the degree to which each particle's velocity is directed

towards the location of the best solution found by the entire population. The inertia factor simulates the particles having mass and controls the degree to which each particle's velocity will continue on its current trajectory. All three of these parameters combine to determine the movement of the particles through the feasible design space. Manipulating the balance between them causes the algorithm to favor local minima, global minima, or more random search.

4.4.1 Cognitive Constant

Ten iterations of the PSO algorithm were performed for each $C1$ value for both laminates and both objective functions. The average final score of each run at each parameter is summarized in Figure 4.11. After performing the initial trial it was observed that $C1$ values of 1 and 2 were both lower than 3, and it was theorized that an intermediate value may be optimal. This was confirmed when a trial at $C1 = 1.5$ was performed. Across both laminates and objective functions, a cognitive constant of 1.5 provided the best performance. Similar trends were observed for both laminates and objective functions across the full parameter experimental set. Trials where $C1$ was set to 3 were the worst among all trials. This is likely due to the particles being heavily directed to their individual best locations leading to the algorithm becoming stuck on a local minimum. A cognitive constant of 1.5 avoids trapping the algorithm while still promoting convergence by keeping particles near their best location. This value is also slightly higher than the baseline value of 1.

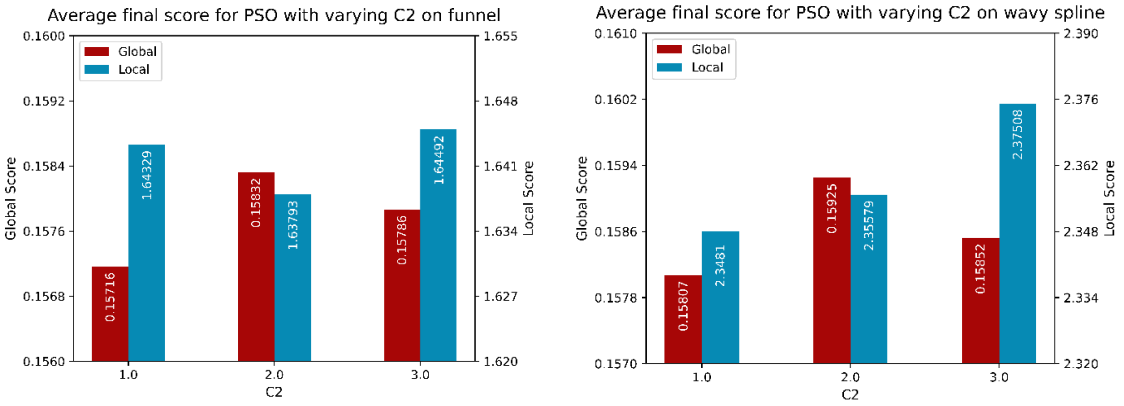


(a) (b)

Figure 4.11: Average final score for PSO with varying cognitive constant, C1, on (a) funnel and (b) wavy spline laminates

4.4.2 Social Constant

Ten iterations of the PSO algorithm were performed for each parameter value on both laminates with both objective functions. The average final score of each set of runs is summarized in Figure 4.12. Results were largely consistent across the four sets of parameter trials. For the global score, the baseline value of 2 performed the worst across both laminates, with a constant value of 1 leading to the best performance. Local score results varied slightly between the two laminates. The wavy spline local trials agreed with the global trials on $C2 = 1$ being the optimal value, but the funnel trials saw increased performance with $C2 = 2$. With three of the four sets in agreement on $C2 = 1$, this will be the value used moving forward. This is lower than the baseline value of 2. Using a lower social constant places less pressure on the particles to converge to the best location found by the population and allows for more search of the design space. This likely led to the higher performance on average seen in the results of this experiment.

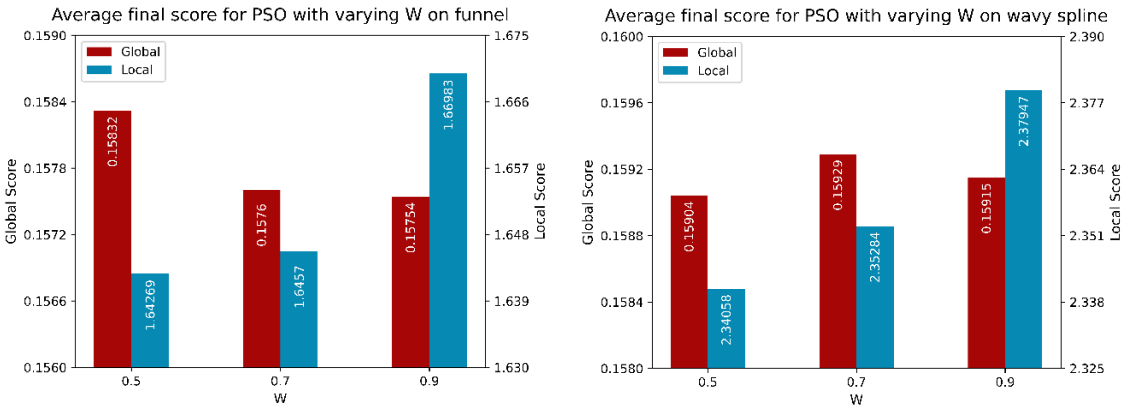


(a) (b)

Figure 4.12: Average final score for PSO with varying social constant, C2, on (a) funnel and (b) wavy spline laminates

4.4.3 Inertia Constant

Ten iterations of the PSO algorithm for each inertial constant value were performed on both laminates with both objective functions. The results are summarized in Figure 4.13. The effect of w on PSO performance was consistent for local score across the two laminates. Higher values corresponded to worse performance with the optimal value being the baseline of 0.5. The global score results differed across the two laminates with the algorithm performing better on the funnel tool with higher values and better on the wavy spline with lower values. With three out of the four sets of trials reaching optimal performance at an inertia constant value of 0.5 this will be used moving forward. A lower value simulates less weight for the particles and allows them to be pulled more strongly towards their individual best location and the population's best location. This likely helped convergence by limiting random search.



(a) (b)

Figure 4.13: Average final score for PSO with varying inertia constant, W, on (a) funnel and (b) wavy spline laminates

4.5 COMPARISONS

The final step is then to investigate whether individual parameter changes resulted in better performance overall for the algorithms. To do this, an additional set of ten runs of each algorithm with the updated parameters was performed on each laminate using each objective function. The following section explores the results of these runs compared to the original baseline runs discussed in Section 4.1. While each parameter has a very specific effect on the progression of the algorithm, evaluation will focus on overall performance defined by final scores.

The first comparison is between baseline and optimized parameters for runs on the funnel laminate using global score as the objective function. The results are summarized in Figure 4.14. For each algorithm, the optimized parameters improved the performance of the algorithm on average. Specifically, the final average score of GA, DE, and PSO improved by 0.4%, 1.1%, and 1.5% respectively. These are not drastic improvements but illustrate the slight success of the parameter optimization work. Reducing the amount of defect stacking even by as little as 1% can result in more efficient designs depending on

the laminate and baseline margins. The GA having the lowest improvement makes sense given one parameter remained at the baseline value and the other was shown to not have a large effect.

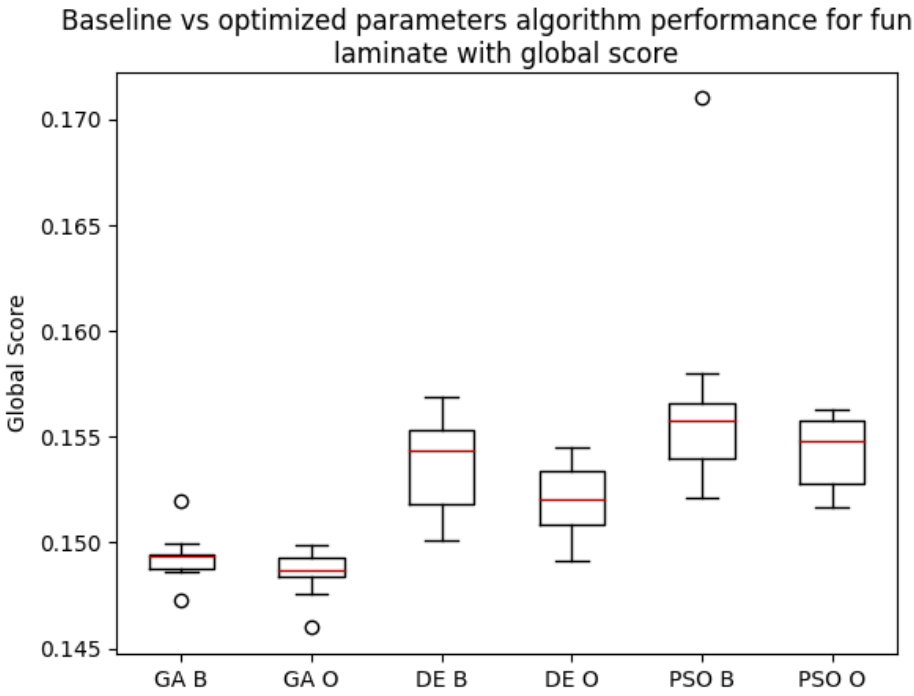


Figure 4.14: Results for algorithm runs with baseline vs optimized parameters on funnel laminate with global score. "B" denotes baseline results and "O" denotes an optimized result

The second comparison is also on the funnel laminate but with local score runs instead of global score. The results are summarized in Figure 4.15. For this set, only the GA and DE saw improvement while the PSO algorithm regressed with optimized parameters. The GA and DE improved by 0.4% and 0.3% respectively while the PS performed 0.3% worse. This could possibly be due to the social constant selection. While three out of the four sets of trials performed better with a social constant of one, the funnel laminate local score set performed better with a value of two. This highlights the complexity of optimization algorithms and the level of specificity that's required to reach optimal performance.

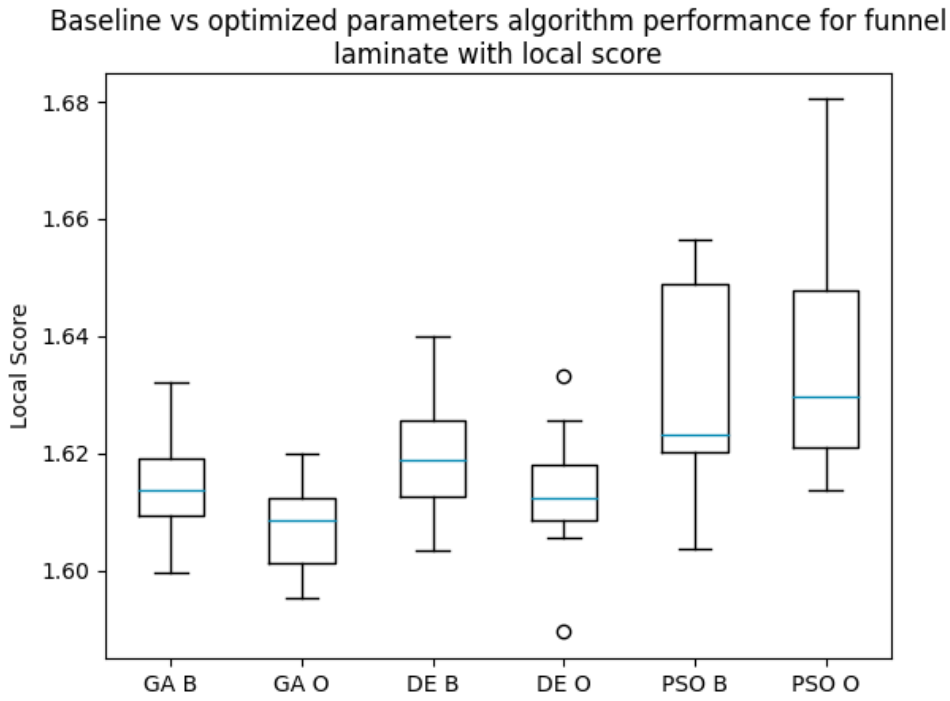


Figure 4.15: Results for algorithm runs with baseline vs optimized parameters on funnel laminate with local score. "B" denotes baseline results and "O" denotes an optimized result

A similar story can be seen with the baseline vs optimized comparison of the wavy spline laminate runs using global score. As shown in Figure 4.16, both GA and DE saw increased performance with optimized parameters while the PSO algorithm slightly regressed. While this result can't be explained by a specific parameter selection, it could be due to the sample size and stochasticity of the algorithms. Each algorithm incorporates a certain amount of randomness, meaning no two runs are exactly alike. This was accounted for by doing ten runs of each algorithm for each problem setup, however more may be needed to filter out the noise. Another example of this is the difference in baseline vs optimized GA runs. The optimized GA performed 1.3% better despite being nearly identical to the baseline GA.

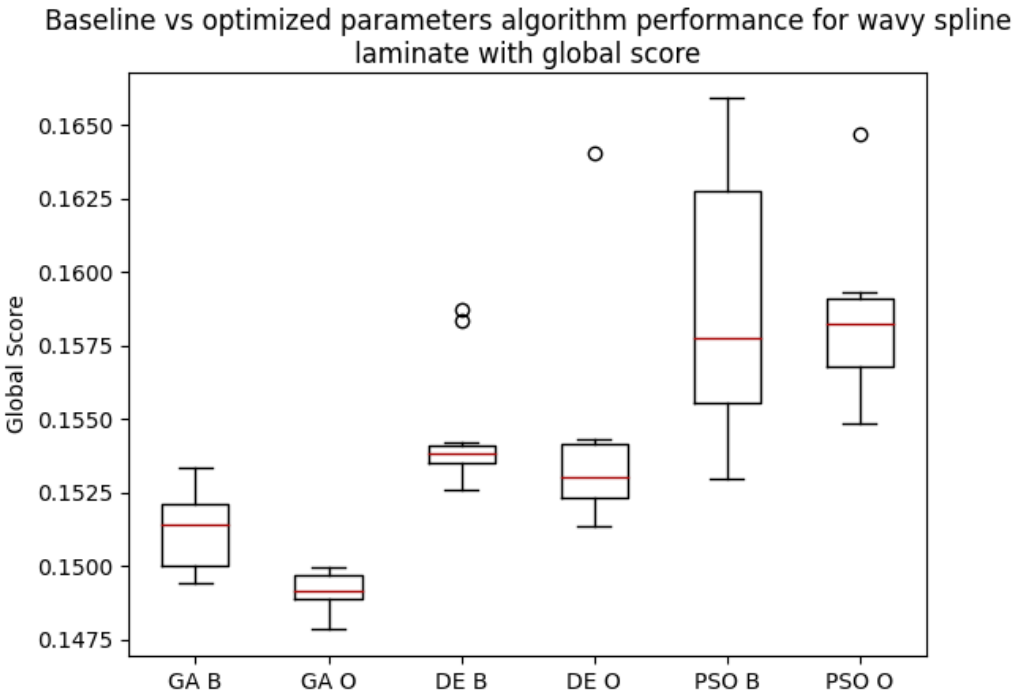


Figure 4.16: Results for algorithm runs with baseline vs optimized parameters on wavy spline laminate with global score. "B" denotes baseline results and "O" denotes an optimized result

The final comparison is between baseline and optimized algorithm runs on the wavy spline laminate using local score. The results (Figure 4.17) show improvement across the board for the optimized parameters over baseline values. GA, DE, and PSO finished on scores that were 0.6%, 0.7%, and 1% better than baseline runs. Again, the improvements are small but point to some level of benefit being achieved by the parameter optimization work.

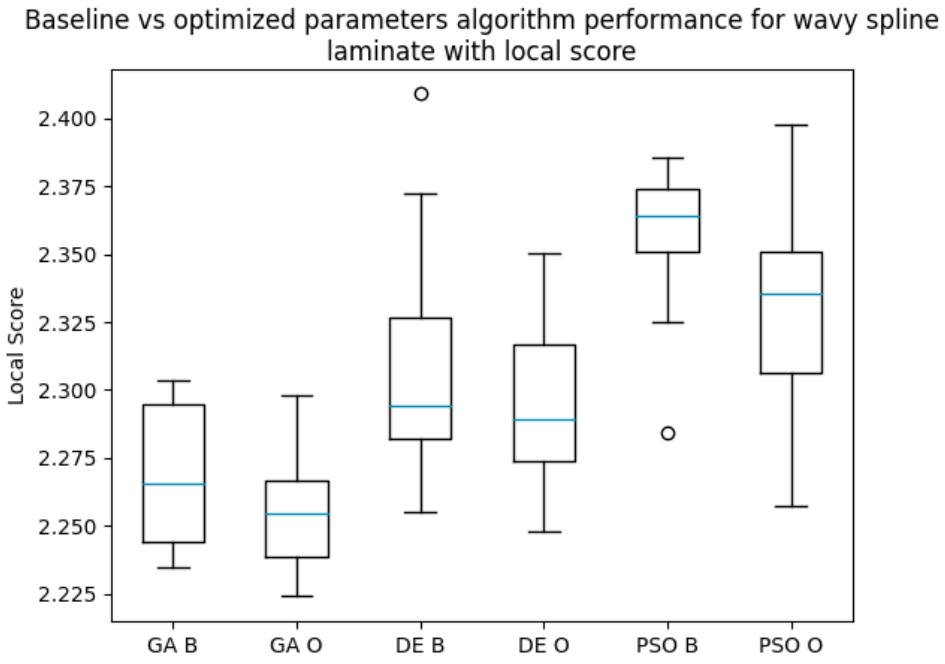


Figure 4.17: Results for algorithm runs with baseline vs optimized parameters on wavy spline laminate with local score. "B" denotes baseline results and "O" denotes an optimized result

There are a couple main takeaways from the parameter experimentation. First, the effect each parameter has on an algorithm is complex and very specific to the context of the optimization problem. Some values may work better for one objective function over another, or on one data set (laminate) over another. While many parameter experiments showed a unanimous optimal value, there were a handful of trials that stood as exceptions to the trend. This speaks to the specificity of these algorithms and their parameters. Additionally, there was not a huge difference in performance between the baseline runs and the updated parameter runs. Only one of the twelve comparisons yielded a t-test p-value below 0.05 meaning there was a statistically significant difference in results. This is in part due to the fact that out of seven parameters investigated, three remained at baseline values after experimentation and one was determined to not have much effect. The lack of

difference in parameter values suggests that the selection of baseline values was strong and also that finding better values applicable to two different laminates and two different objective functions is difficult. Others seeking to employ an optimization algorithm to solve a problem should take heed of the typical values used in literature, as this work suggests little benefit can be achieved by tweaking them. However, if absolute optimal algorithm performance is desired, some gains may be made, but only when optimizing the parameters to solve a specific problem. While parameters were tweaked and evaluated individually, their effects are connected within the algorithms and additional gains may be achieved through more extensive experimentation.

4.6 CONVERGENCE CRITERIA

The final facet of the optimization process that needs to be decided is convergence criteria. There are a multitude of methods to terminate an optimization algorithm and determine convergence. Common methods include setting a max number of generations, setting an objective function value threshold, or a specific run time. Using max generations or max run time as a convergence criterion works fine, but neither is sensitive to the actual performance of the algorithm. Additionally, setting a score threshold requires knowledge of a suitably optimal value prior to initiating the algorithm, which is not feasible with this application. Other common criteria include those that are specifically tailored to the algorithm that it's used on. For example, PSO could be terminated when the average distance between particles falls below a certain threshold. However, this type of criteria prevents a direct comparison across the three metaheuristic algorithms chosen for this work. Therefore, a single convergence criterion that responds to the performance of the algorithm but is universally applicable was desired.

Seven different convergence criteria were identified. The first is triggered when the standard deviation of the population falls below the average of the population multiplied by a tolerance value. The second is triggered when the improvement of the best objective function value within the population falls below a threshold for a certain number of generations. The third is triggered when the improvement of the average objective function value of the population falls below a threshold for a certain number of generations. The fourth is triggered when the standard deviation of the objective function values of the population falls below a threshold. The fifth is triggered when the difference between the objective function values of the best and worst solution of the population falls below a threshold. The sixth is similar to the fifth except that the range of the population is compared to a tolerance value multiplied by the average score of the population. The seventh criterion focuses not on the objective function scores but rather the population itself and is triggered when a certain percentage of the population shares a certain percentage of individual design variables.

Two combination convergence criteria were also evaluated. The first is triggered when the change in best objective function value within the population and the change in average objective function value fall below specific thresholds for a certain number of generations. This combination was selected because while the best solution within a population may not improve for several generations, the population as a whole is still evolving. The addition of the change in average score check ensures that the entire population is also relatively stagnant before converging. The second combination criterion is triggered when the change in best objective function value within the population falls below a threshold for a certain number of generations and the standard deviation of the population is below a certain value.

This combination also attempts to balance the change in best criterion with a measure of the variance of the population. All nine convergence criteria are defined in Table 4.3.

Table 4.3: Convergence criteria expressions

Convergence Criteria	Expression
CC1	$\sigma_i \leq \bar{x}_i * t_1$
CC2	$\Delta \min(x_i) \leq t_2$
CC3	$\Delta \bar{x}_i \leq t_3$
CC4	$\sigma_i \leq t_4$
CC5	$\max(x_i) - \min(x_i) \leq t_5$
CC6	$\max(x_i) - \min(x_i) \leq \bar{x}_i * t_6$
CC7	$\frac{\left[\frac{P_i \in \{P_{j \neq i}\}}{n}\right]_{i=1}^{n}}{m}$
CC8	$\Delta \min(x_i) \leq t_{8a}$ and $\Delta \bar{x}_i \leq t_{8b}$
CC9	$\Delta \min(x_i) \leq t_{9a}$ and $\sigma_i \leq t_{9b}$

These criteria were evaluated on longer 80 generation runs across both laminates, objective functions, and all three algorithms. Evaluation took place after the run had completed 80 generations. In doing so, the criteria were not enforced proactively but retroactively. This enabled comparisons to be made between where each criterion would have stopped the algorithm and where the algorithm ended up after 80 generations. The threshold, tolerance, and other criteria parameters were tuned for each set of ten runs. They were adjusted so the average final score was minimized while ensuring that all ten runs still converged within the 80-generation limit. Criteria were evaluated based on average final score, score standard deviation, average generation of convergence, and average difference between the converged score and the final score after 80 generations. An example set of runs for the GA on the funnel laminate using global score is shown in Figure 4.18. The figure shows a zoomed portion of the algorithm's progression with color coded circles on each iteration corresponding to where each convergence criteria would have

triggered. In some instances, these circles overlap, meaning multiple criteria were triggered. The legend shows the colors of the nine different convergence criteria markers denoted “CC#”.

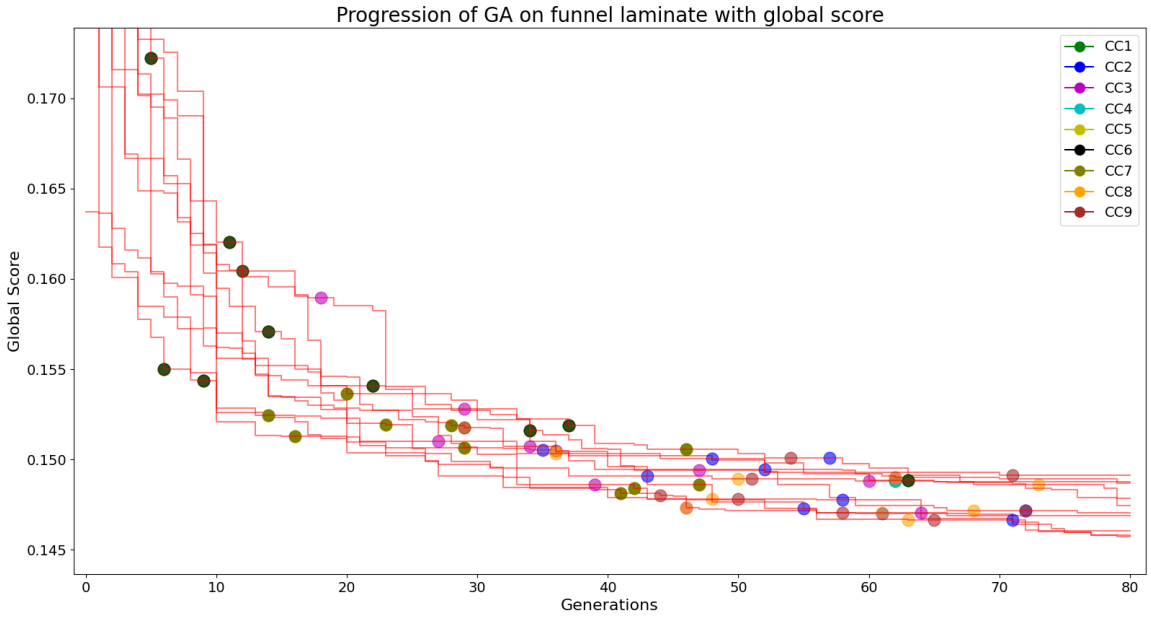


Figure 4.18: Ten 80 generation GA runs on the funnel laminate using global score with convergence marked for each criterion

The first notable finding was that each of the selected criteria includes at least one tunable parameter that determines when the criterion will trigger. While this was known prior to experimentation, it was discovered that these values are rather sensitive and require manipulation to achieve reliable termination at a reasonable point in the algorithm’s progression. Therefore, none of the criteria could be imposed prior to running the algorithms unless the user has knowledge about how the specific metrics that the selected criteria rely on will progress. This essentially removes the benefit these criteria have over ones which require the score to reach a certain threshold, since knowledge of the results is needed.

Another surprising outcome of the investigation was how much variability there was with where each criterion would trigger between different runs of the algorithm. For example, the first convergence criteria, denoted CC1, converged on the fifth generation for one run and the 63rd generation for another run. While these algorithms are stochastic and every run is different, that amount of variability prevents the algorithm from reaching its full potential and is unacceptable. Even the most reliable criterion ranged from converging on generation 35 to generation 71. That's nearly half the range of the total number of generations possible. It's theorized that this variability might be reduced with stricter convergence criteria parameters. However, this would extend the runs and push the algorithms into a region of diminishing returns.

Sensitivity and reliability aside, some criterion performed better than others in regard to average generation converged, standard deviation of scores converged on, and difference between converged score and final score after 80 generations. Particularly, the first combination criterion, denoted CC8 in Figure 4.18, which combined change in best score with change in average score for a certain number of generations performed well. While it had one of the higher average generations of convergence, this enabled it to achieve a very low score standard deviation and difference between the converged score and the score at 80 generations. Specifically for the set of runs shown in Figure 4.18 the average generation of convergence was 54 with an average difference of only 0.0011 in final global score between CC8 and the max generation limit.

4.7 CONCLUSIONS

Driven by the "No free lunch" theorem, an experiment was devised to investigate the effect of different parameters on the performance of the GA, DE, and PSO algorithms.

Initial trials were performed with baseline parameter values for each algorithm using both objective functions on two separate laminates. Each parameter was then manipulated individually within a set range and additional trials were performed. Values for each parameter were selected based on the average final score across all four optimization setups. A final set of trials were then performed using the updated values which were compared to the baseline set. Finally, several convergence criteria were explored and tested.

Overall, little statistically significant difference in performance was observed between the performance of the algorithms with baseline values and optimized values. There are several explanations for this. The baseline values were carefully selected based on what had been used successfully within similar studies. Out of the seven parameters, three remained unchanged from baseline and one was deemed to not have an effect on the algorithm for this specific problem context. This means there wasn't a significant difference between the baseline and optimized algorithms to begin with. Additionally, parameter values were selected based on what worked best across all four problem setups. While one optimized parameter differs between global and local score, all of the others are common. There were several instances where one of the four problem setups differed from the trend exhibited by the rest, and choosing the most commonly optimal value may have reduced the effectiveness of the parameter optimization work.

Nine distinct convergence criteria were created and evaluated on each algorithm, laminate, and objective function combination. These criteria were selected in part due to the ability to employ them without having prior knowledge of how the optimization will progress. However, the sensitivity of the various thresholds and parameters specific to each

criterion prevents this application and a trial-and-error approach is necessary to tune them. Additionally, a striking variability was observed between the generation that each criterion would trigger at across different iterations of the algorithm. This variability could be reduced with more stringent criteria parameters; however this would likely incur significant efficiency reductions. The best criterion among those tested was a combination sensitive to the change in best score and the change in average score over a certain number of generations.

With the metaheuristic algorithms optimized for LLO and convergence criteria identified, the next chapter will evaluate the performance of each algorithm in conjunction with GS and GHSD through a virtual case study.

CHAPTER 5

VIRTUAL CASE STUDY

The presented optimization techniques were tested and compared on a more realistic aerospace part through a virtual case study. This chapter outlines the experimental setup, results, and conclusions.

5.1 EXPERIMENTAL SETUP

In order to evaluate the efficiency and performance of the different optimization approaches, a virtual case study was performed. The study focused on optimizing the laminate-level layup strategy for a wind tunnel blade laminate called the "PREDICT" tool. 56 plies were generated using a quasi-isotropic, balanced, symmetric stacking sequence and a constant ply boundary. The tool surface and ply boundary are shown in Figure 5.1. The tool measures approximately 4.13 ft long by 1.71 ft wide with a surface area of 6.89 ft^2. Ten ply scenarios were generated for each ply with a random starting point and the rosette coverage strategy. Rosette was chosen due to its propensity to promote gap and overlap formation, giving more data for the algorithms to optimize for. This simulates a scenario where a process planner is most concerned with angle deviation or steering, hence choosing the rosette strategy, but wants to reduce the stack-up of gaps and overlaps as much as possible.

Toolpaths were generated within VCP assuming courses with up to eight ¼" tows. Tow-dropping was set to 50% which splits the bias evenly between gaps and overlaps, and

a maximum tow-to-tow overlap of 0.125” was enforced. Due to the tool surface used and layup strategies employed, the ply scenarios generated included vastly more overlaps than gaps. For this reason, the results presented will focus on overlap stacking minimization. Toolpath generation and defect prediction was performed by VCP version 8.3.1. The optimization algorithms were written in Python 3.10 and run on a Dell OptiPlex 5000 PC with a 12th Gen Intel© Core™ i7-12700 CPU. All reported computation times were based on this setup and would certainly change if replicated in another language or on different hardware.

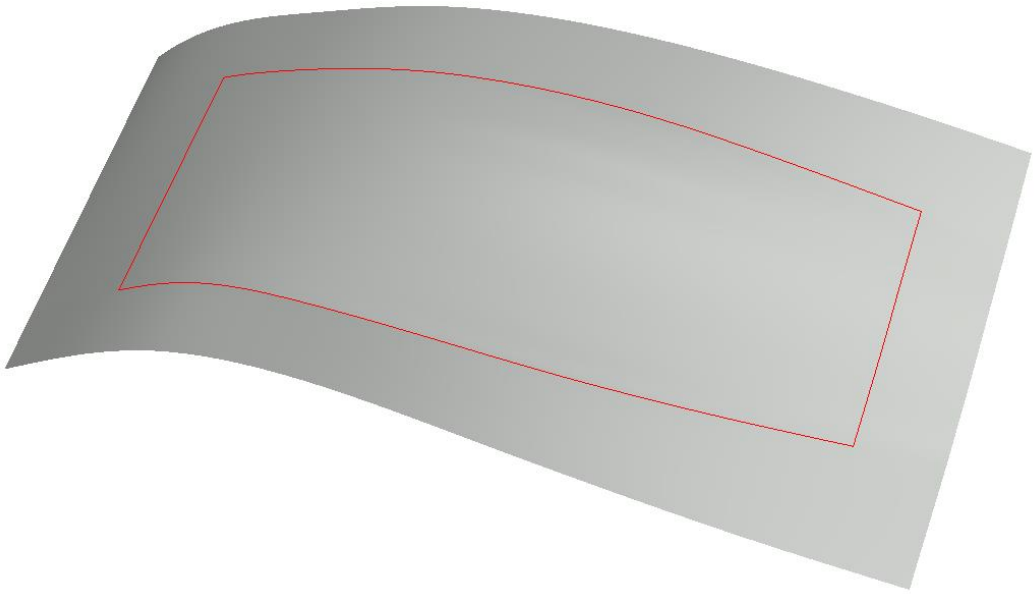


Figure 5.1: PREDICT wind tunnel blade tool surface with ply boundary

Optimization trials were performed using GS, GA, DE, and PSO paired with both global and local score. GS was run once per objective function since it is a deterministic algorithm. GA, DE, and PSO were run ten times each to establish an average progression since they are stochastic algorithms. A max generation convergence criterion of 100 generations was used to generate a consistent data set for each algorithm. GHSD was only

run once since it is deterministic and is not paired with an objective function. The tool surface consists of three faces, each running the full length of the part. Each face was discretized into a 509x187 array to facilitate optimization. This size was chosen to minimize the error induced by parametric discretization while still running efficiently. For global score runs, an even weighting was set between gap and overlap distributions as well as frequency and severity sub-scores. The stacking threshold was set at level two, meaning any stacking of defects through-the-thickness of the laminate was undesirable. For local score, an analysis window of 5 x 5 x 5 was used, meaning defect stacking would be evaluated within two plies above and below the ply in question and within two elements in both in-plane dimensions. This size was based on the magnitude of fiber waviness observed in manufactured laminates in the literature, the size of the tool, and discretization used.

5.2 GLOBAL SCORE TRIALS

Results from the global score trials are shown in Figure 5.2. The vertical axis of each graph measures global score, where lower is better. The horizontal axis measures the generations, showing how each algorithm progresses over time. Each red line represents one run of the algorithm. The dashed black line shows the average final score of each set of ten iterations of an algorithm. The dotted black line shows the score of the laminate optimized by GS for reference.

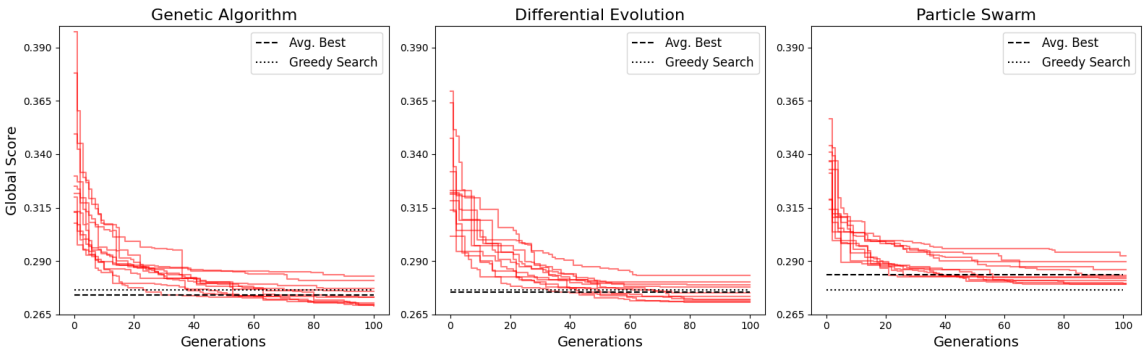


Figure 5.2: Progression of optimization algorithms on PREDICT laminate using global score

Two of the three metaheuristic algorithms outperformed GS, with PSO lagging slightly behind. The GA and DE algorithms improved quickly through the first 20 generations before making slower progress through the max generation limit. The DE tended to converge around generation 80, failing to find better laminate scenarios in the final ~30 generations. The GA had several runs continue to improve in the final 20 generations, pushing the average lower than the DE. This is likely due to a difference in how the mutation operation is performed between the two algorithms. With the GA, mutation occurs on a ply-by-ply basis where each ply scenario has a chance to mutate to any other ply scenario within the ply. With the DE, mutation occurs through perturbing the given candidate vector by adding the scaled difference between two other candidate vectors in the population. This means that as the algorithm progresses and the population becomes more similar, mutation is reduced. While this approach promotes convergence, it limits search of the design space, particularly in later generations.

The PSO algorithm began similar to the GA and DE but leveled off more quickly. This is likely due to most of the population converging on one location within the design space prematurely. Another factor causing PSO's comparatively lesser performance could

be the implicit assumption of continuity within the design space. The PSO algorithm represents candidate solutions as particles moving through n-dimensional space. PSO searches through the design space by manipulating the velocities of the particles within the population. Specifically, the social, cognitive, and inertial factors influence how a particle will be drawn to specific locations identified by the population. The issue with this heuristic is that it relies on the assumption that two nearby locations are likely close in objective function score. In reality, two locations that differ by just one integer value in every dimension are two entirely different laminate scenarios with no relation. This spatial representation might be contributing to a higher likelihood of premature convergence on a local minimum.

Table 5.1 shows the numerical results of the global score trials. The greedy search algorithm terminated on a laminate scenario with a global score of 0.276. The GA, DE, and PSO algorithms finished on laminate scenarios with average global scores of 0.274, 0.275, and 0.283 respectively. Looking at optimization runtimes, all three of the metaheuristic algorithms required almost the same amount of time, coming in at just over two minutes. This suggests that the vast majority of computation time is spent evaluating the objective function and not performing the various internal operations specific to each algorithm. The GS algorithm progressed much faster, finishing in less than half of the time of the metaheuristic algorithms. While the DE and GA did outperform the GS in terms of objective score minimization, the difference in score compared to the difference in time only justifies their use if time is not a limitation or if absolute optimization is desired.

Table 5.1: Global score trials results

	Global Score		
	Final Average	Best Overall	Time [s]
GS	N/A	0.276	61.664
GA	0.274	0.269	137.764
DE	0.275	0.270	137.233
PSO	0.283	0.279	136.501

Figure 5.3 - Figure 5.7 show the 2D stack-up of overlap defects within the laminate scenarios optimized by GS, GA, DE, PSO and a randomly generated laminate scenario respectively. Overlap defects are represented as pixels which are color scaled depending on how many overlaps are stacked through the thickness of the laminate at that area. Dark blue represents one defect through-thickness with light blues, yellow, orange, and red denoting increased levels of defect stacking. Each of the four optimized laminate scenarios are dominated by dark and light blue regions, with just a few small areas of yellow and orange. The randomly generated laminate scenario by comparison has many areas colored dark orange and red, particularly in the 90° (vertical) direction. The progression from more stacked defects seen in the random laminate scenario to more evenly distributed defects common among the optimized laminate scenarios is clear. Patterns can also be observed across all images. This grid-like pattern is driven by the constant course width creating defects in a periodic fashion. Moving the starting point does move the defects around, but largely only within a single course-width, before the pattern repeats.

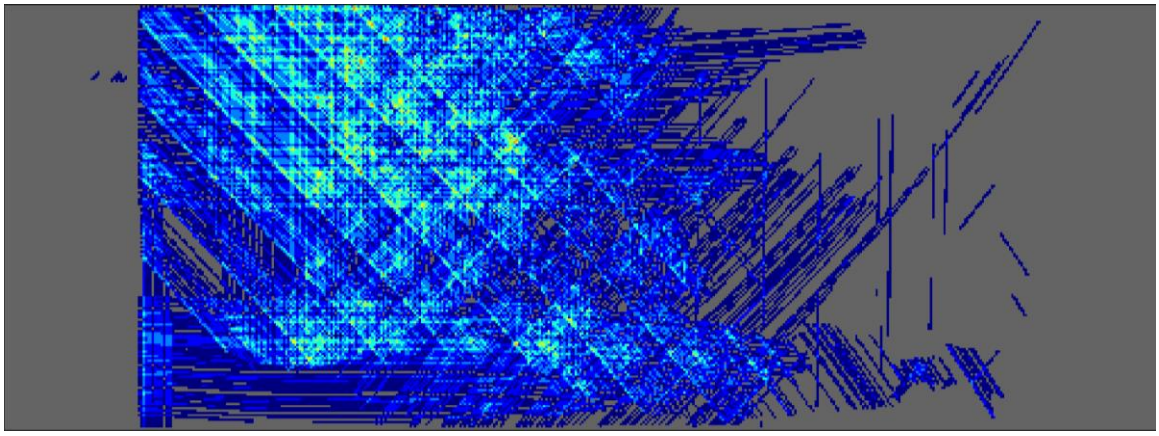


Figure 5.3: 2D overlap stack-up for PREDICT laminate optimized by GS with global score

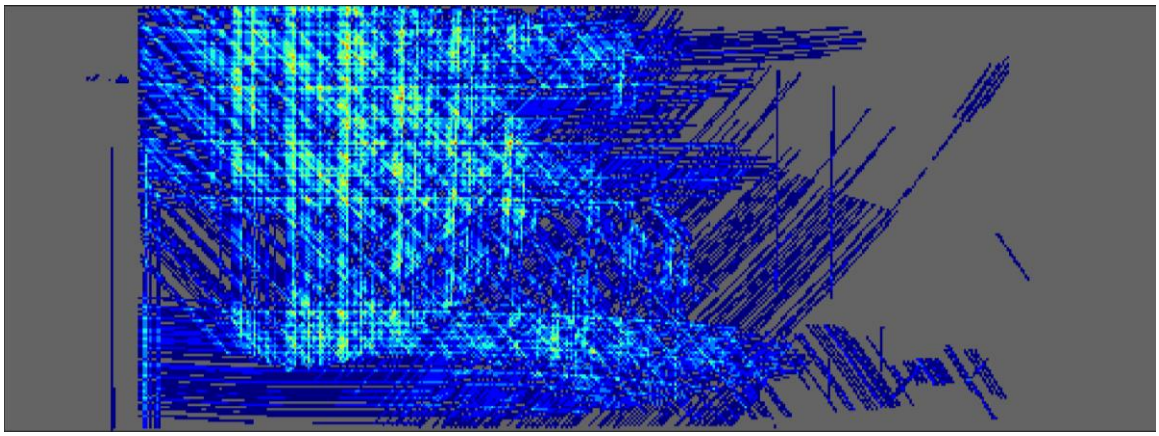


Figure 5.4: 2D overlap stack-up for PREDICT laminate optimized by GA with global score

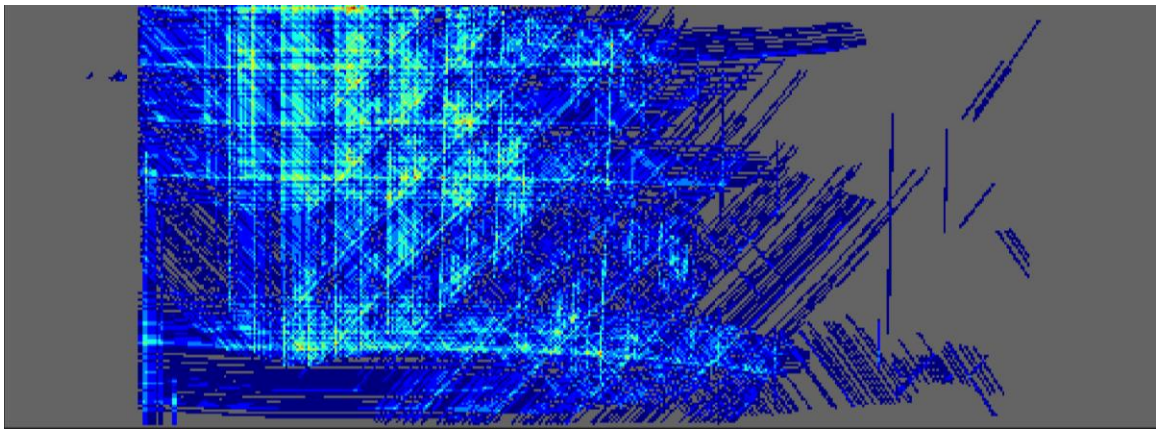


Figure 5.5: 2D overlap stack-up for PREDICT laminate optimized by DE with global score

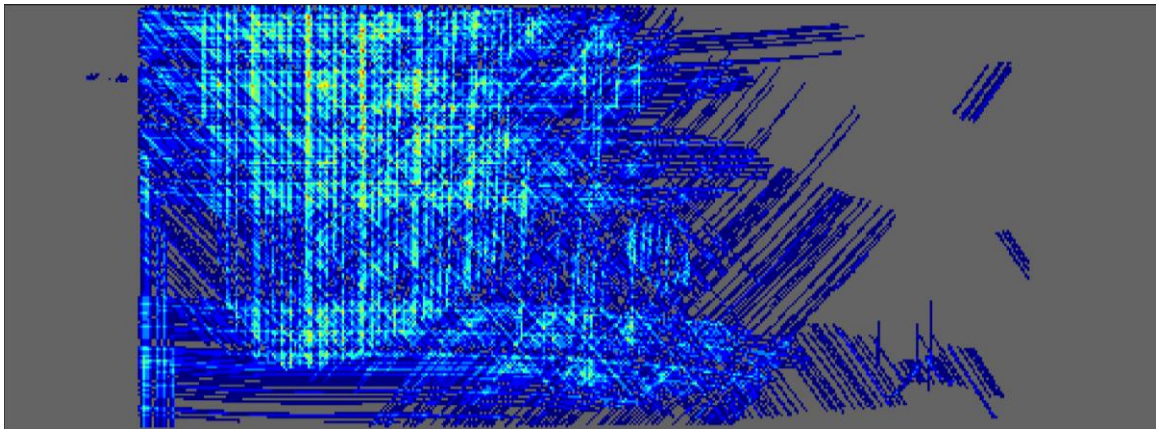


Figure 5.6: 2D overlap stack-up for PREDICT laminate optimized by PSO with global score

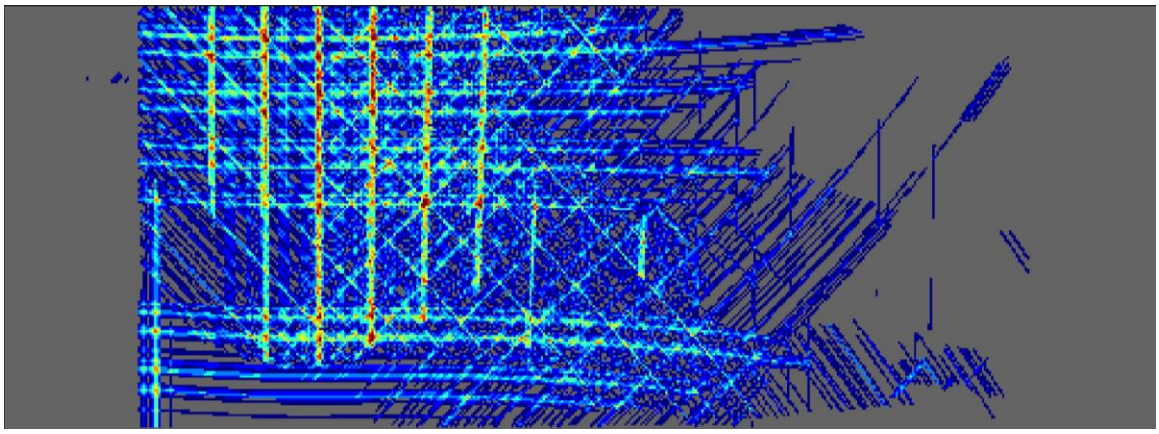


Figure 5.7: 2D overlap stack-up for PREDICT laminate randomly generated

While the visualizations are helpful on a high-level for seeing the impact of the optimization, examining the data can provide some more quantitative conclusions. Figure 5.8 shows the distribution of overlap stack-up data for each of the optimized laminate scenarios and the randomly generated laminate scenario. The horizontal axis shows each level of defect stacking, with level one referring to the areas where only one defect is located through-the-thickness of the laminate. Level two then denotes the regions where two defects are stacked, and so on until the max number of levels is reached. The theoretical max level of defect stacking is equal to the total number of plies in the laminate, however most laminate scenarios only achieve a max level that is about 25% of the total ply count. The vertical axis of Figure 5.8 measures the number of cells at each level and is log-scaled. While each cell has a unique area due to the parametric discretization step, the difference in area is small, so this metric can be thought of as similar to surface area.

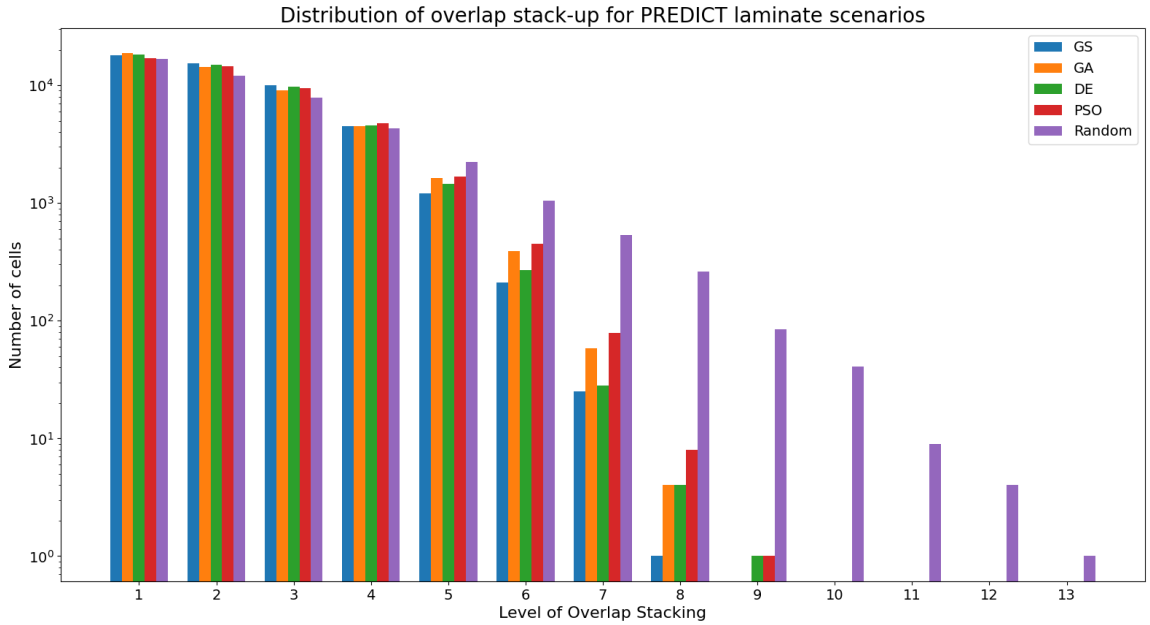


Figure 5.8: Distribution of the overlap stack-up data for each PREDICT laminate scenario

The data clearly show a sharp reduction in defect stacking, particularly at the higher levels. The GS and GA laminate scenarios only had eight levels of overlap stacking, with the DE and PSO laminate scenarios including one cell each at level nine. In comparison, the randomly generated laminate scenario had a region where overlaps were stacked between 13 of the 56 total plies. The ability of the optimization algorithms to identify laminate scenarios that avoid these regions of severe stacking serves as one of the main benefits. The random laminate scenario actually has less defect area than the optimized laminates at levels 1-4, with more area in all subsequent levels. This is a result of the optimization algorithms pushing the defect distributions towards ones with more area stacked less severely. Comparing the optimized laminate scenarios, it appears that the GS laminate has the same or more cells for levels 1-4 before outperforming the other optimized laminate scenarios through levels 5-8. While the difference in the earlier levels may appear

small, the log-scale minimizes its appearance, and this difference is likely driving the difference in global score.

5.3 LOCAL SCORE TRIALS

Results from the local score trials are shown in Figure 5.9. Each metaheuristic algorithm was run ten times to establish an average performance, with each run represented by a red line in the figure. The vertical axis measures the local score of the best laminate scenario found by the algorithm as it progresses. The horizontal axis measures the generation of the algorithm. A dashed line has been added to each graph showing the score of the best laminate scenario averaged across all ten runs. The dotted line shows the score of the laminate scenario optimized by GS.

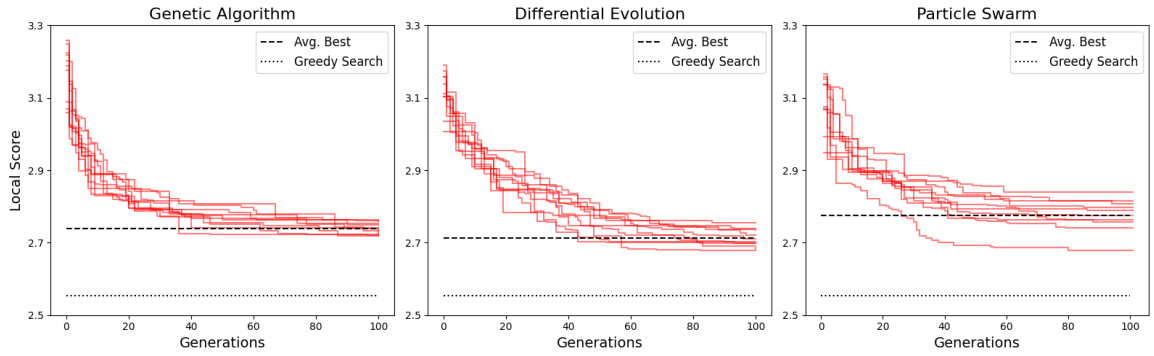


Figure 5.9: Progression of optimization algorithms on PREDICT laminate using local score

The most notable result of the local score trials was the superior performance of the GS algorithm. While the GA, DE, and PSO algorithms were averaging final laminate local scores of 2.738, 2.712, and 2.775 respectively, the GS algorithm produced a laminate scenario with a local score of 2.554. This is likely due to an interaction between the GS algorithm and local score as an objective function. GS proceeds ply-by-ply making sequentially, locally optimal decisions without revisiting any plies. Local score evaluates

defect stacking within a set range of plies, two above and below for this study. Global score evaluates defect stacking completely through-the-thickness of the laminate. When GS is paired with global score, it lacks most of the data necessary to make an informed decision until it has decided half of the plies in the laminate. However, when GS is paired with local score, it nearly always has exactly half of the data necessary to make an informed decision. The restriction in analysis scope imposed by local score lends itself to GS's sequential progression. The metaheuristic algorithms only ever evaluate laminate scenarios in their entirety, so they perform about as well with global score as local score.

Between the metaheuristic algorithms DE performed the best on average. It improved quickly through the first ten generations followed by a largely linear progression in the middle before converging around generation 80. The GA performed similarly, with quicker improvement early on before a premature convergence slightly higher than the DE. PSO again was the worst performing metaheuristic algorithm on average. The runs improved early on but converged about halfway through and failed to make significant improvements through the latter 50 generations. The results are summarized in Table 5.2. As with global score, all the metaheuristic algorithms had near identical runtimes. The main difference is that with local score they took much longer to complete, about 32 times longer on average. This confirms the theory that most of the computation time is driven by the objective function and not the internal operations specific to each algorithm. The switch to local score also drastically affected GS, which went from a runtime of just over a minute to over nine hours.

The difference in computation time is driven by the difference in complexity of the two objective functions. Global score measures the distribution of integer values in a 2D

array, which is a very quick operation. Local score measures the distribution of integer values within a set range of nearby cells at every cell in a 3D array. This takes much longer to compute and increases as the size of the analysis window increases. The benefit of using local score is that in addition to minimizing through-thickness defect stacking it also minimizes in-plane and diagonal defect interactions as well. It is built explicitly by measuring randomness, and therefore serves as a good choice when striving to achieve near-random defect distributions.

Table 5.2: Local score trials results

	Local Score		
Algorithm	Final Average	Best Overall	Time [s]
GS	N/A	2.554	33421.632
GA	2.738	2.719	4362.099
DE	2.712	2.677	4406.074
PSO	2.775	2.678	4440.181

Figure 5.10 - Figure 5.13 show the 2D overlap stack-up images of PREDICT laminate scenarios optimized using local score with GS, GA, DE, and PSO respectively. Again, these images are color-scaled representations of the parametric stack-up of overlap defects. While the global score objective function analyzes the underlying 2D data set directly, the local score objective function analyzes the defect data in a 3D array. These images then are slightly better suited for visualizing global score results than local score, since the number of plies between stacked defects is obscured. Regardless, they show a clear improvement from the randomly generated laminate scenario (Figure 5.7) to all of the locally optimized laminate scenarios. There is not much visual difference between the optimized laminate scenarios despite the large difference in local score. It does appear that the GS laminate scenario has less area colored light blue or teal than the other optimized laminates.

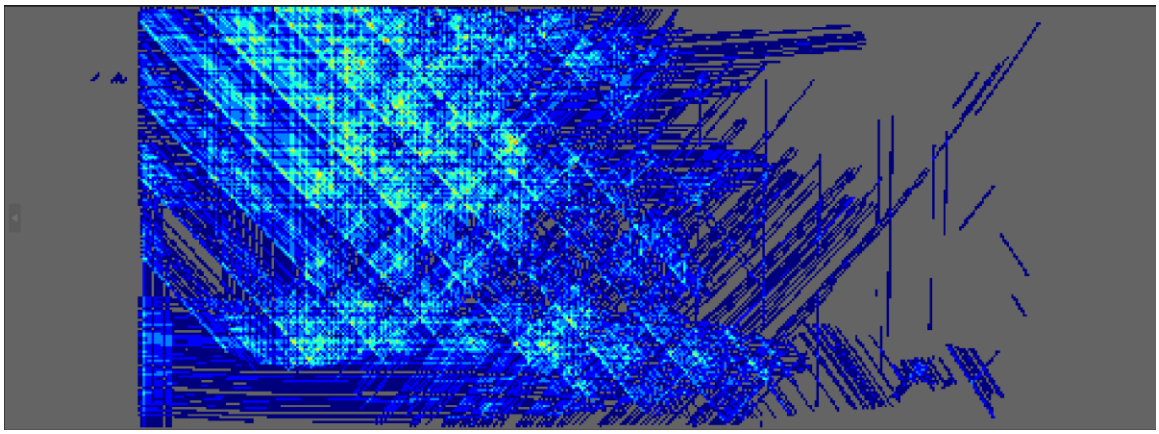


Figure 5.10: 2D overlap stack-up for PREDICT laminate optimized by GS with local score

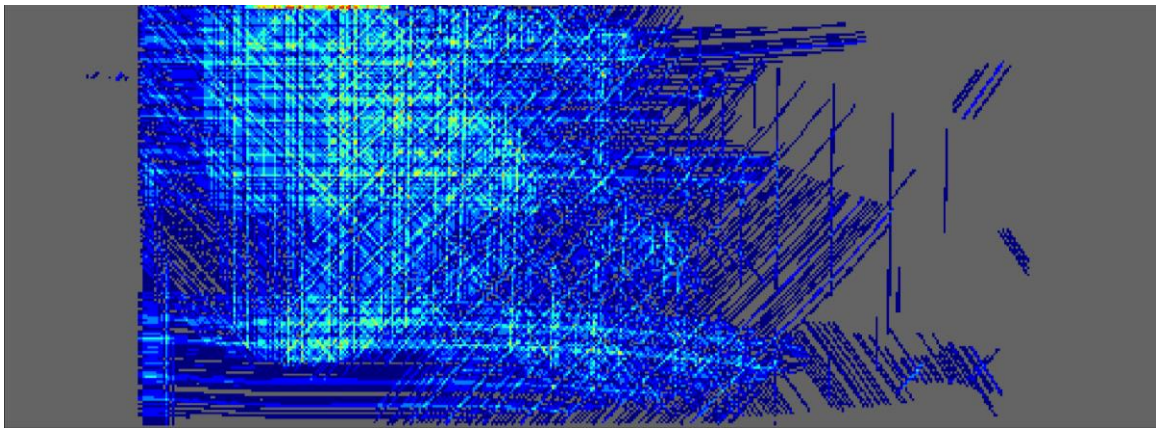


Figure 5.11: 2D overlap stack-up for PREDICT laminate optimized by GA with local score

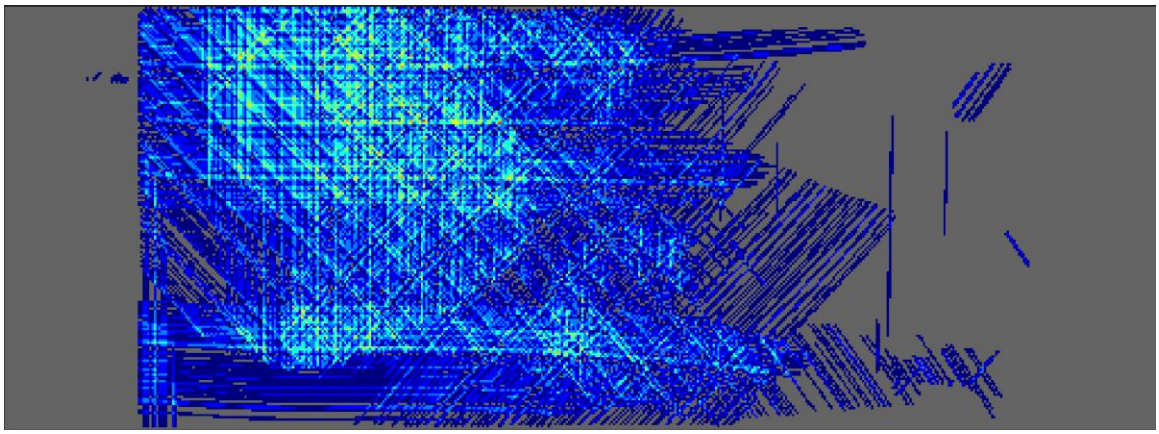


Figure 5.12: 2D overlap stack-up for PREDICT laminate optimized by DE with local score

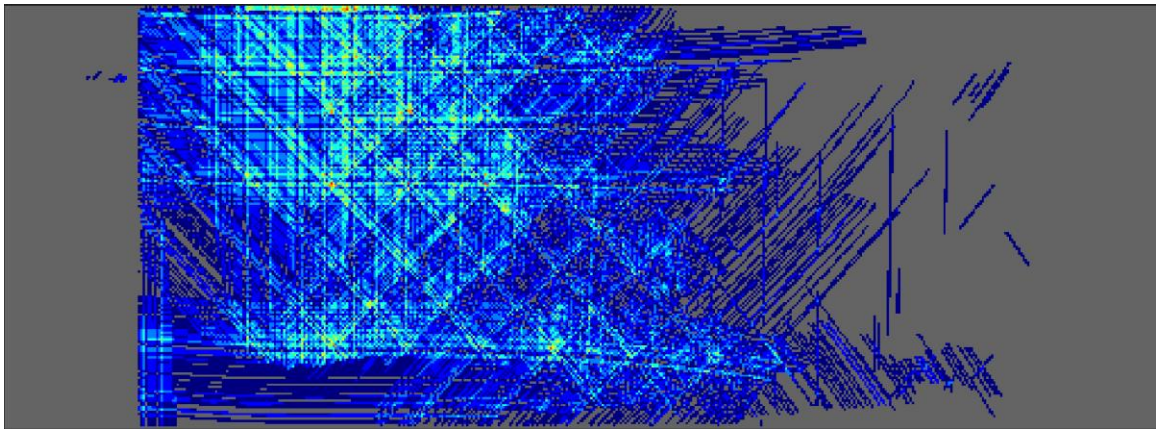


Figure 5.13: 2D overlap stack-up for PREDICT laminate optimized by PSO with local score

Examining the underlying data behind the images reveals a similar story to the global score trials. Figure 5.14 shows the distribution of surface area with overlaps stacked at different levels of severity for each of the optimized laminates and a randomly generated laminate scenario. Specifically, the graph shows the number of cells along the vertical axis, however since the discretization error is low and each cell represents roughly the same amount of area, this can be thought of as pseudo-surface area. The randomly generated

laminate scenario had at least one region on the laminate with 13 overlap defects stacked through-thickness. This was reduced to eight or nine overlaps stacked through-thickness for the optimized laminates. Interestingly, the best scoring laminate scenario (GS) had the most area among the optimized laminates for levels five through nine. This is surprising given the goal of the optimization is to reduce areas of severe defect stacking. However, this data set, like the 2D images shown previously, is different from the data local score evaluates. It is likely that the overlaps that are stacked between eight or nine plies within the GS laminate are separated by intermediate plies. This separation is accounted for within local score and the defect interaction is negated. However, global score draws on this data set directly and ignores any intermediate plies when evaluating defect interaction. The complexity of the local score objective function makes accurate visualization difficult to achieve.

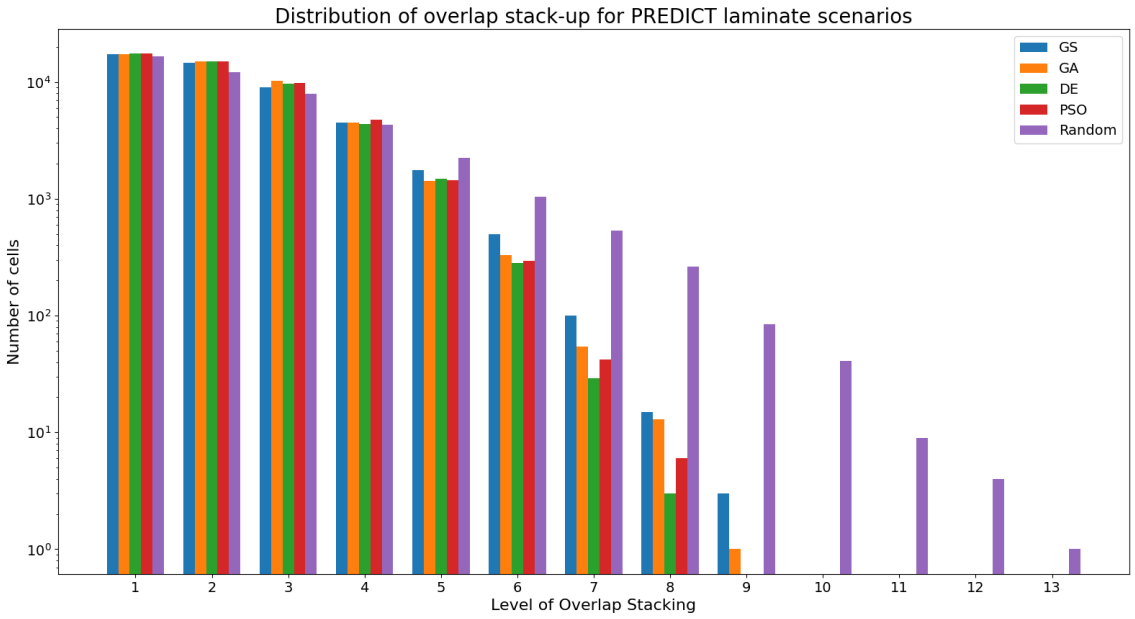


Figure 5.14: Distribution of overlap stack-up data for each PREDICT laminate scenario

5.4 GLOBAL HOT SPOT DETECTION RESULTS

GHSD is an optimization approach separate from the traditional framework of an objective function paired with an algorithm. It scores the input variables themselves based on the likelihood that each ply scenario's defects will stack within the combined design space of possible laminate scenarios. This is distinct from the global and local score objective functions which evaluate entire laminates. This approach is also unique in that it places a hard limit on the amount of computation required to reach a solution. Once every ply scenario is scored, the algorithm is essentially complete. The best scoring ply scenario from each ply is simply chosen and the final laminate scenario is formed. Shifting from searching through the entire design space to looking directly at the inputs makes it an incredibly efficient algorithm. GHSD completed a run of the PREDICT laminate in 9.623s, which is 6.5 times faster than GS with global score and 3542 times faster than GS with local score.

This efficiency comes at a cost: performance. Figure 5.15 shows the 2D overlap stack-up for the PREDICT laminate optimized by GHSD. There are a few regions of severe defect stacking. These regions appear to be driven by defects stacking between plies of different ply angles. This stands in contrast to the randomly generated laminate scenario which saw most severe stacking between plies of a common fiber angle, specifically 90° plies. While the defect distribution is slightly more condensed than the other optimized laminates, it is still clearly more dispersed than the randomly generated laminate scenario.

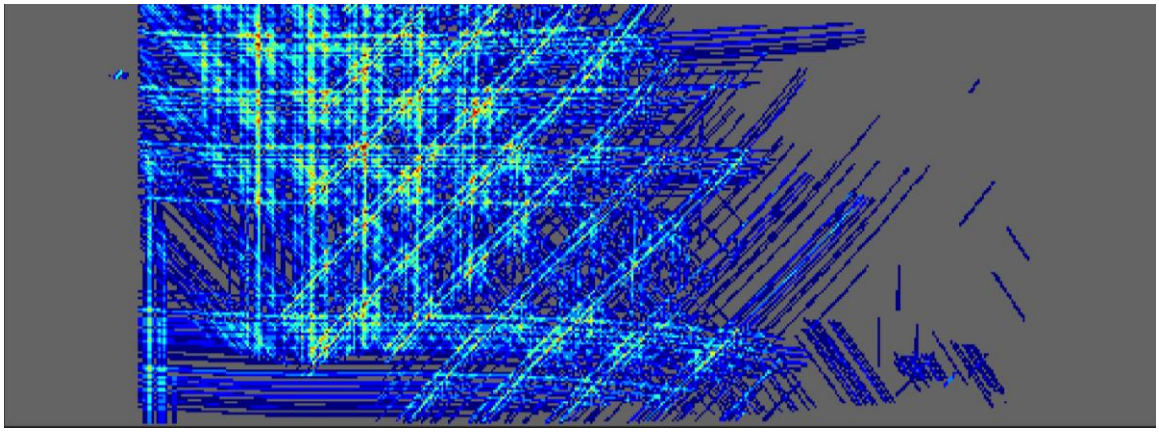


Figure 5.15: 2D overlap stack-up for PREDICT laminate optimized by GHSD

The GHSD method includes one thresholding parameter which determines what degree of stacking within the total design space is undesirable, similar to the global score threshold. This parameter is somewhat difficult to set as it is sensitive to the total number of plies and ply scenarios within the design space. An experiment was conducted evaluating different threshold values for GHSD on the PREDICT laminate. Values from 5-56 were tested and the global and local score of the optimized laminate scenario were recorded. The results are shown in Figure 5.16. GHSD's ply scenario scoring system is much closer to global score than local score, so the local score results are rather poor. Both score distributions fluctuate significantly across the range of threshold values. This highlights the difficulty of determining an optimal value without iterating through a set, reducing the overall efficiency and reliability of the approach.

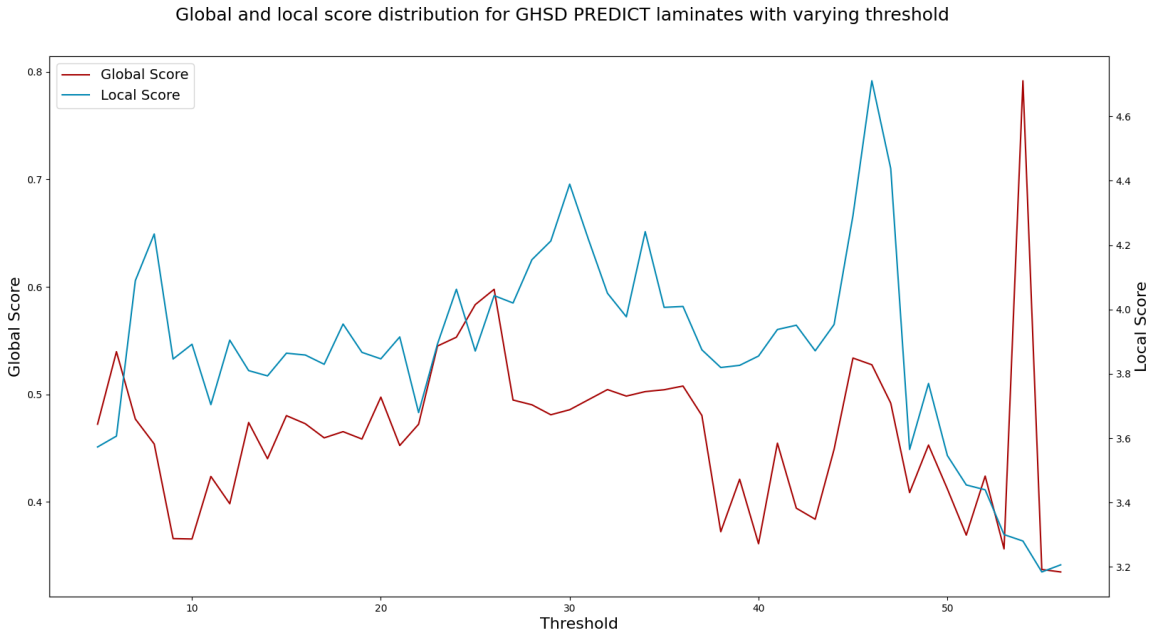


Figure 5.16: Global and local score distribution for GHSD with varying threshold

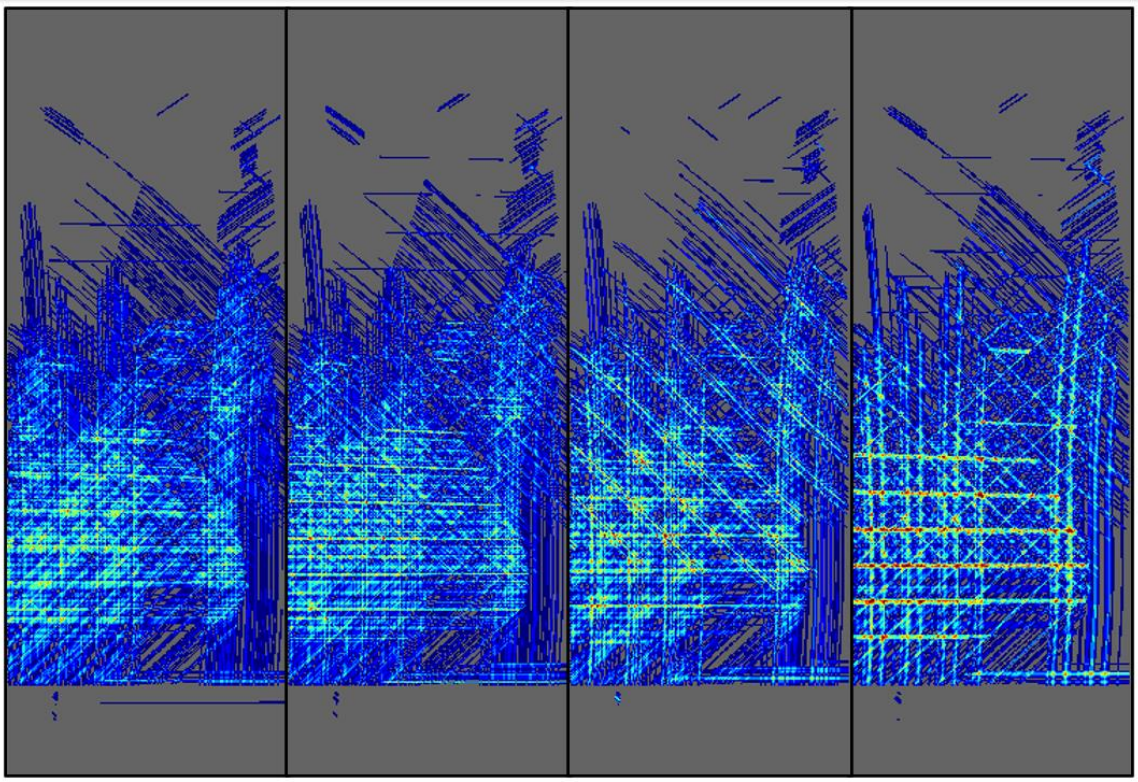
The optimal threshold value for the PREDICT laminate design space based on global score was 56. This yielded a global score of 0.335 and a local score of 3.206. Using the GA with global score would have produced a laminate scenario with a better global score in approximately ten generations. Using the GA with local score would have produced a laminate scenario with a better local score in about five generations. This negates the efficiency benefit gained from using GHSD over one of the traditional optimization approaches like GA or GS. However, if the goal is not optimality but just better than random, then this approach serves as a quick way to produce an acceptable laminate scenario.

5.5 COMPARISONS

Comparing the best laminate scenarios produced using global score, local score, and GHSD with the randomly generated laminate scenario reveals a few interesting

conclusions. Figure 5.17 shows the overlap stack-ups for the best scoring laminate scenarios identified during the global score trials, local score trials, GHSD experiment, and the random laminate scenario. It's clear that all three of the optimized laminate scenarios represent a significant improvement from the random laminate scenario regarding through-thickness overlap stacking. The random laminate scenario includes large regions of severe defect stacking, particularly between 90° plies. GHSD was able to reduce the amount of defect stacking among common angle plies, with most of the highly stacked regions occurring at intersections of defects from all four ply angles. The best results were achieved by the more traditional optimization methods.

The difference in computation time required to run the optimization algorithms is rather significant. The fastest method was GHSD which finished in 9.623s, compared to the slowest method (GS with local score) which took over nine hours. These numbers are not concrete however, and more serve to show the general difference in efficiency between the approaches. As discussed previously, the GA with global score could produce a laminate scenario with a better score in the same amount of time as GHSD if terminated early. Efficiency gains within the GS algorithm could be achieved particularly when paired with local score through reducing repeated evaluations of previously analyzed plies and threading. While it seems unlikely that local score optimization runs will ever be quite as efficient as global score runs, the gap could certainly be narrowed with further development.



a) GA + Global Score
Time: 2m 18s

b) GS + Local Score
Time: 9h 17m 2s

c) GHSD
Time: 9.623s

d) Randomly Generated
Time: ~0s

Figure 5.17: 2D overlap stack-up images for PREDICT laminate scenarios optimized by a) GA with global score, b) GS with local score, c) GHSD, and d) a randomly generated laminate scenario

The local score and global score laminate scenarios show strong dispersion of overlap defects and very few regions of severe stacking. Comparing the two, the global score laminate perhaps has more light blue and teal area while the local score laminate has a few orange and red pixels scattered throughout. Again, this 2D representation lends itself better to visualizing the global score results since the objective function measures this data directly. The more highly stacked regions of the local score laminate are likely separated by intermediate plies, which fail to qualify as stacking within the local score objective function. Figure 5.18 shows the distribution of overlap stack-up data for the four PREDICT laminate scenarios. The graph shows a strong trend from the random laminate scenario, to

GHSD, local, and global score optimized laminates. GHSD was able to avoid some of the most severe levels of stacking but failed to eliminate regions stacked with ten or eleven overlaps through-thickness. The global and local score laminates reduced the overlap stack-up further, identifying a laminate scenario with as few as eight plies that share an overlap defect location. Again, the data presented is in a slightly different format than the local score objective function directly uses. Consequently, this representation fails to capture the defect distribution as local score evaluates it.

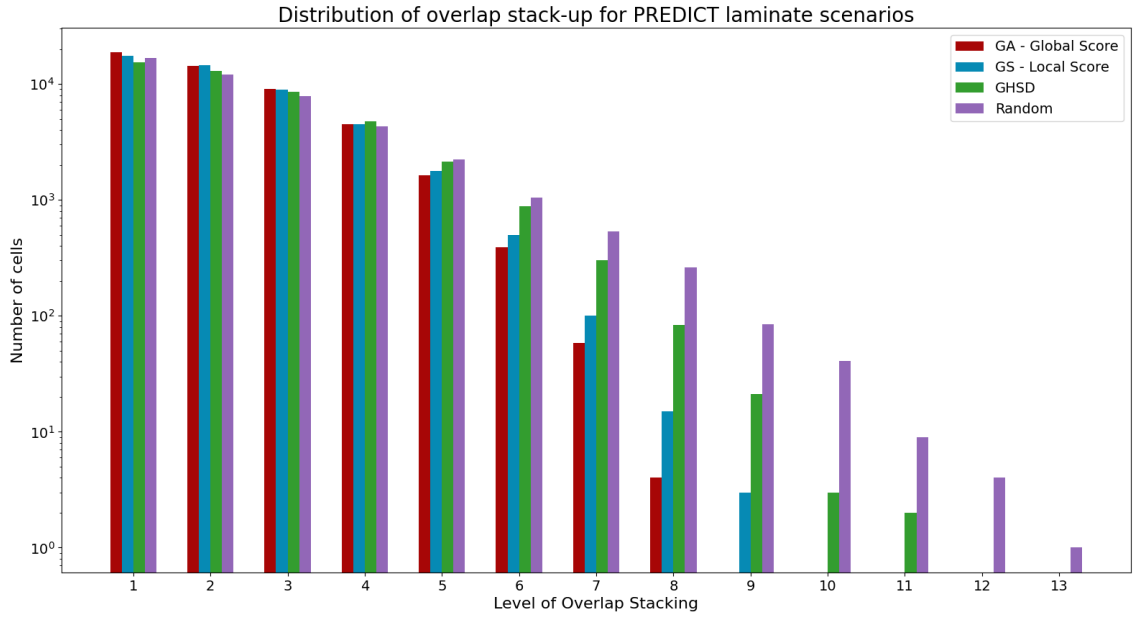


Figure 5.18: Distribution of overlap stack-up data for optimized and randomly generated PREDICT laminate scenarios

While the optimized laminate scenarios may be better than a single randomly generated laminate scenario, it's worth investigating how they compare to a more robust random sampling. This would be comparable to current methodologies used by process planners to eliminate defect stacking. Figure 5.19 shows the distribution of global and local scores for 500 randomly generated PREDICT laminate scenarios. The vertical orange line on each graph denotes the score of the best laminate scenario identified during the global

and local score trials. Each score distribution is roughly normal with a slight right skew. The global score sample had a mean of 0.418 and a standard deviation of 0.0633. The local score sample had a mean of 3.343 and a standard deviation of 0.166. This puts the globally and locally optimized laminate scenarios at 2.48 and 4.76 standard deviations below their respective means. Assuming a normal distribution, this means there was a less than 1% chance of producing a laminate scenario with a better score if chosen randomly. This result gives good confidence in the performance of the optimization approaches presented.

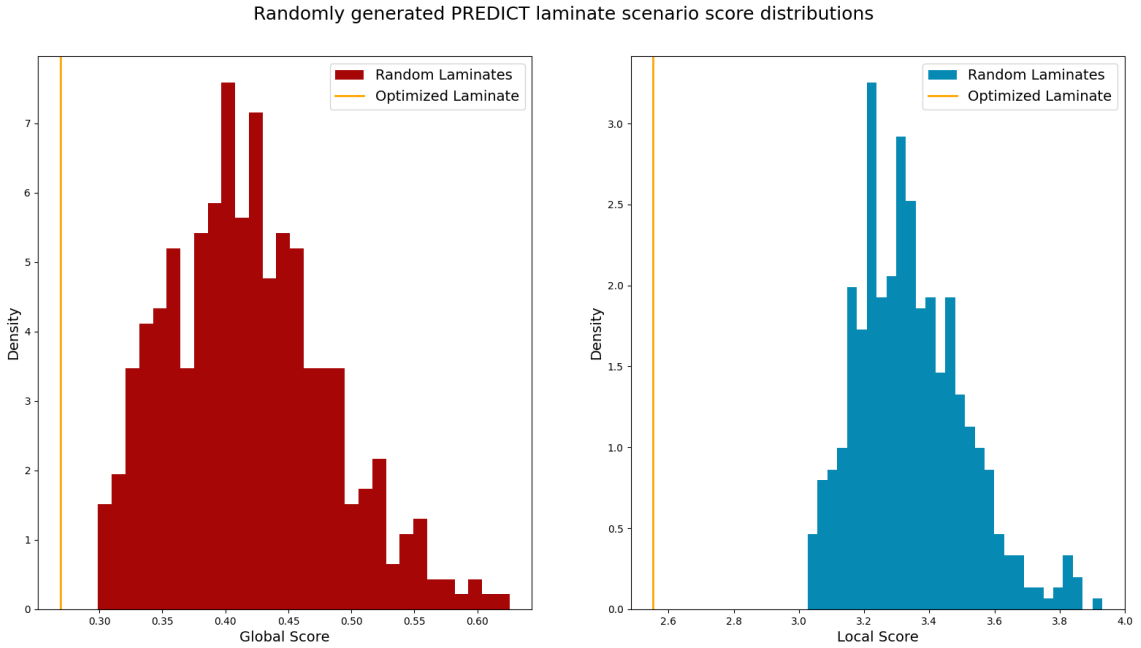


Figure 5.19: Global and local score distribution for randomly generated PREDICT laminate scenarios

It is largely unknown why the optimization algorithms were able to push further from the mean with local score than with global score. One possible explanation lies with the sensitivity and range of each score. Local score is based on a measure of entropy and is unbounded. The typical range of values for a laminate will vary depending on the size, curvature, density of defects, and discretization used. Local score is also sensitive to in-

plane defect distribution, making it a much more complex objective function. Conversely, the global score is normalized from zero to one and only measures defect stacking through-the-thickness. This means there is more likely to be different laminate scenarios with very similar scores, and less range overall. It is possible this simpler objective function gave less room for optimization than the more complex local score.

5.6 CONCLUSIONS

Multiple optimization trials were performed evaluating different approaches to achieve the optimal laminate-level layup strategy for a wind tunnel blade laminate. Four traditional combinatorial optimization algorithms were paired with two different objective functions and tested alongside a novel optimization approach. The first objective function, global score, was best utilized by the GA which identified a laminate scenario with a global score of 0.269 in 137.76s. All of the optimized laminate scenarios had defect distributions that were significantly less stacked than the randomly generated laminate scenario. The GS algorithm performed very similarly to the metaheuristic algorithms when paired with global score. This trend changed with the local score trials where GS outperformed all the other algorithms. GS was able to identify a laminate scenario with a local score of 2.554 but took a staggering nine hours to do so. The GHSD showed promise as an approach that trades performance for efficiency. However, this efficiency was never realized due to the unreliability of the threshold parameter and lack of performance compared to the traditional optimization approaches.

The two lowest scoring laminate scenarios identified during the global and local score trials had significantly lower scores than a sampling of 500 randomly generated laminate scenarios. Comparing the stack-up images of the optimized laminates with a

random laminate reinforces the benefit achieved by this process. Through LLO, multiple levels of defect stacking were able to be eliminated. Based on the literature presented in Chapter 2, this is predicted to result in laminates with closer to pristine mechanical properties and less thickness variation. In addition to identifying the optimal laminate scenario and producing a 2D image of the defect stack-up, this data can be mapped back to the tool surface for further analysis. Figure 5.20 shows the laminate scenario produced by GS with local score mapped back onto the PREDICT tool within the CAPP module. With this representation, a process planner could identify where the regions of severe defect stacking are occurring and decide between near-optimal laminate scenarios based on downstream impacts such as assembly interfaces or critically loaded zones.

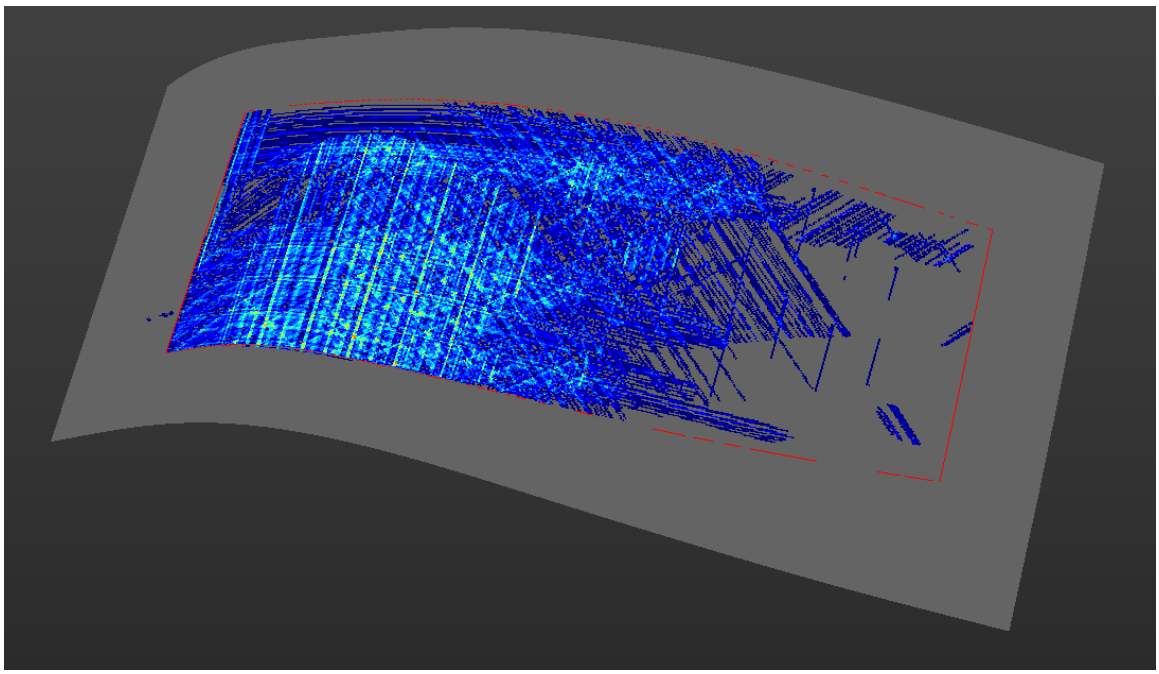


Figure 5.20: GS locally optimized PREDICT laminate scenario with overlap stack-up and ply boundary shown on tool surface within CAPP module

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This final chapter will provide a summary of the presented work, short discussion, and highlight potential paths forward.

6.1 SUMMARY OF WORK

Process planning is an essential phase in the product lifecycle that connects design intent with manufacturing resources. One of the key process planning workflows for AFP is toolpath generation which determines exactly how and where material will be laid up on a tool surface. The CAPP module was created to help automate and optimize process planning for AFP and began with ply-level optimization. This thesis presented an expansion of CAPP to optimize toolpaths based on the severity of defect stacking through-the-thickness of the laminate. The motivation for this work lies in the knowledge that the greatest effects of gap and overlap defects on part mechanical performance are realized on the laminate level. To achieve this goal several optimization algorithms and objective functions were developed. These approaches were refined through experiments on complex tool surfaces and then evaluated on a realistic laminate.

Several traditional optimization algorithms were developed alongside two custom objective functions. The algorithms included greedy search, genetic algorithm, differential evolution, and particle swarm optimization. These were paired with two objective functions, global and local score, which evaluate gap and overlap stacking at different

scales within the laminate. A separate novel optimization approach called global hot spot detection was created to reduce computation time without sacrificing performance. The traditional optimization algorithms were tuned through virtual experiments performed on two complex tool surfaces. These experiments reinforced the "No free lunch" theorem and highlighted the difficulty in imposing convergence criteria without prior knowledge of optimality.

Using the updated algorithms, a virtual case study was performed optimizing a 56-ply wind tunnel blade laminate. Of the traditional optimization approaches, global score optimization operated much quicker than local score trials. This difference in runtime was driven by the complexity of the local score objective function which measures not only through-thickness defect stacking but also in-plane distribution. Within the global score trials, the GS algorithm performed about as well as the metaheuristic algorithms. The GA was the highest performer overall and produced a laminate scenario with significantly less overlap stacking than a randomly generated laminate scenario. GS performed the best within the local score trials and substantially outperformed all the metaheuristic algorithms. This performance came at an efficiency cost and the total runtime for the GS using local score was over nine hours. GHSD succeeded in being efficient but at the cost of sub-par performance. The laminate scenario produced was better than a randomly generated laminate, but the approach suffered from reliability issues that affected its applied efficiency.

Overall, the work presented constitutes a robust framework for addressing toolpath dependent gap and overlap defects during the process planning phase. The experiments performed suggest that GS paired with the local score objective function provide the

highest quality results. While the drawback of this approach is a prolonged runtime, improvements could certainly be made in the efficiency of the GS algorithm and more advanced computing hardware would reduce the total time required. Additionally, the end use case for this process is in the initial design and manufacturing of composite laminates. Within the aerospace industry, this only happens once every decade or so and time constraints are less important than having an optimal design and manufacturing plan. Within the range of timeframes observed during this work, choosing to use local score and spending a couple hours to have that optimal plan is likely worth the wait.

6.2 FUTURE WORK

The work described in this manuscript constitutes a starting point for through-thickness process planning optimization and analysis. Currently, the methodologies optimize for gap and overlap defects separately. The objective functions can be tuned to focus more on one defect type than another but there is no system in place for accounting for cross-defect-type interactions. Further experimentation would be necessary to fully understand the physics and implications of these interactions before expanding LLO to account for them. Additionally, other defects are predictable during the process planning phase that could be included. Angle deviation, steering, and wrinkling can all be predicted, to varying degrees of accuracy, and could be added to the optimization framework.

Experiments have been planned to investigate and validate the benefits of LLO through physical manufacturing trials. While the virtual case studies show promising results, there is much to learn about the impact of LLO on internal laminate geometry, defect propagation, thickness variation, and downstream mechanical performance. The infrastructure that was required to perform LLO also enables the creation of a green-state

thickness model. Combining the gap and overlap distributions into one model could provide a reasonably accurate description of where thickness variation is likely to occur. A preliminary version of this is shown in Figure 6.1. Coupling this with a cure model could enable tolerance analyses to predict interfacing issues before manufacturing ever takes place.

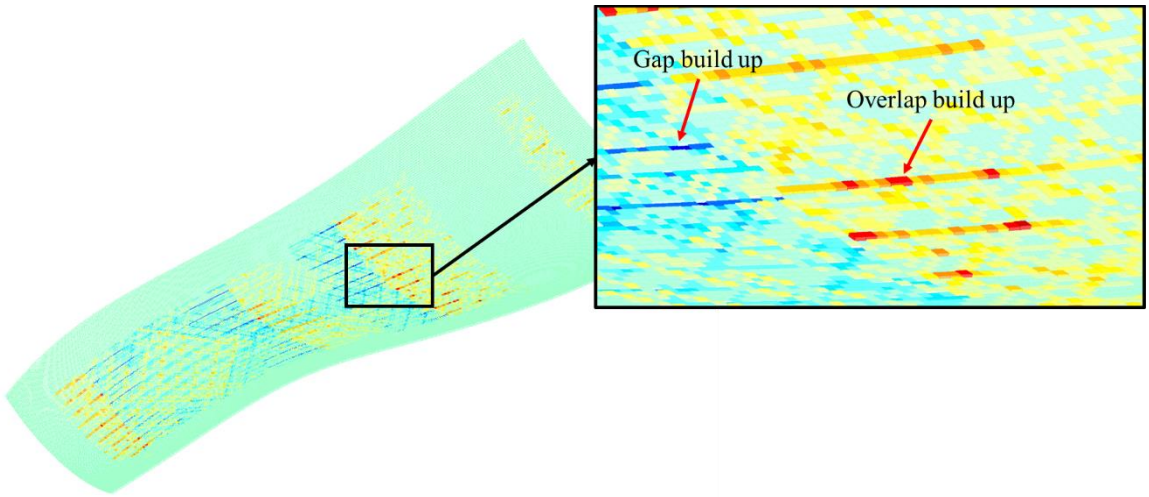


Figure 6.1: Preliminary green-state thickness model created through combining the gap and overlap stack-up models

6.3 SITUATION OF RESEARCH

The creation of LLO builds directly on years of effort undertaken at the University of South Carolina's McNAIR Center concentrated on applying industry 4.0 concepts to the AFP lifecycle. Beginning with design, work has been done investigating composite laminate design optimization [131] and topology optimization for 3D printed reinforced composites [132]. Moving to process planning, early research focused on categorizing process planning functions [49], [133] and AFP defects [21]. These works laid the foundation for the creation of CAPP and development of PLO [19], [20]. Within manufacturing, recent efforts have focused on process parameter modelling and

optimization, specifically looking at AFP heat, speed, and compaction [18]. The cycle is closed with inspection, where custom algorithms have been developed to automatically identify AFP defects [134]. These isolated efforts combine to form a framework for a new paradigm of composite manufacturing, summarized by Figure 6.2. The work described in this manuscript lives within the process planning pillar but attempts to strengthen the digital connection between design and manufacturing.

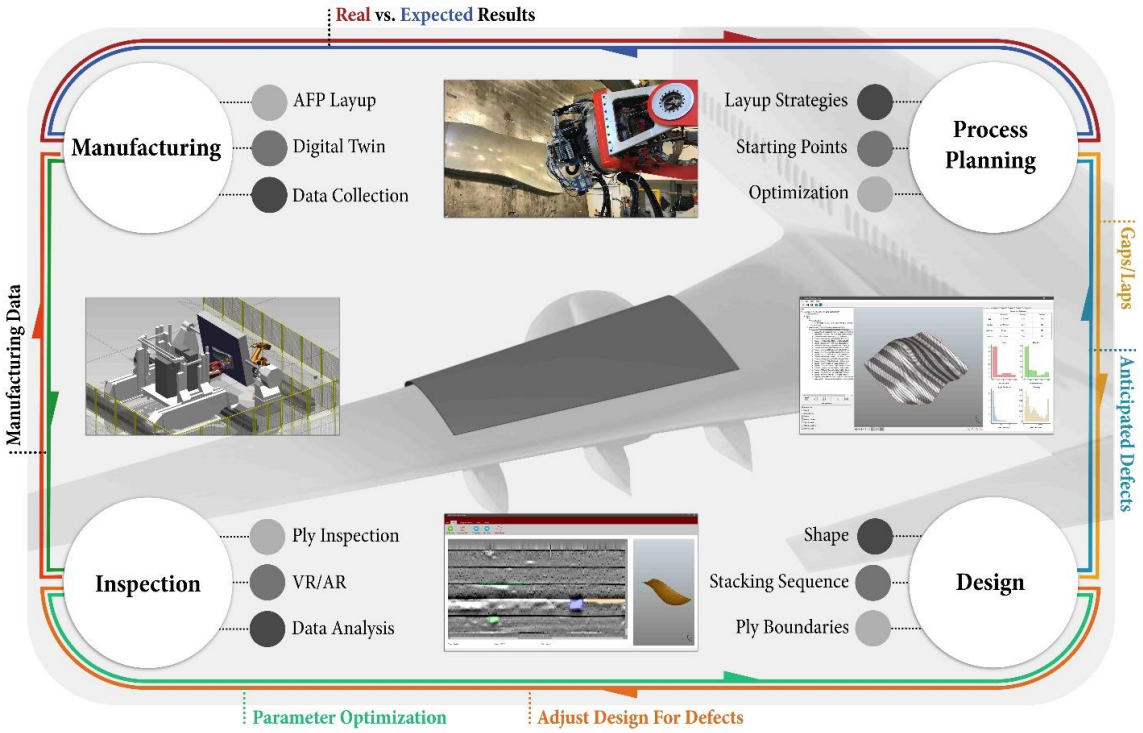


Figure 6.2: Industry 4.0 AFP lifecycle workflow diagram [135]

REFERENCES

[1] R. R. Nagavally, “Composite Materials - History, Types, Fabrication Techniques, Advantages, and Applications,” *International Journal of Mechanical And Production Engineering*, no. 2, 2016.

[2] G. Dorey, “Carbon fibres and their applications,” in *Journal of Physics D: Applied Physics*, 1987. doi: 10.1088/0022-3727/20/3/002.

[3] W. Watt, L. Phillips, and W. Johnson, “High Strength High Modulus Carbon Fibres,” *The Engineering*, vol. 221, pp. 815–816, 1966.

[4] P. Hörmann, D. Stelzl, R. Lichtinger, S. van Nieuwenhove, G. Mazón Carro, and K. Drechsler, “On the numerical prediction of radiative heat transfer for thermoset automated fiber placement,” *Compos Part A Appl Sci Manuf*, vol. 67, 2014, doi: 10.1016/j.compositesa.2014.08.019.

[5] M. di Francesco, L. Veldenz, G. Dell’Anno, and K. Potter, “Heater power control for multi-material, variable speed Automated Fibre Placement,” *Compos Part A Appl Sci Manuf*, vol. 101, 2017, doi: 10.1016/j.compositesa.2017.06.015.

[6] P. K. Mallick, *Processing of Polymer Matrix Composites*. 2017. doi: 10.1201/9781315157252.

[7] S. Ekşi and K. Genel, “Comparison of mechanical properties of unidirectional and woven carbon, glass and aramid fiber reinforced epoxy composites,” in *Acta Physica Polonica A*, 2017. doi: 10.12693/APhysPolA.132.879.

[8] J. T. Quinlivan and H. R. Fenbert, “Composite Applications in Commercial Transport Aircraft,” in *Polymers and Other Advanced Materials*, 1995. doi: 10.1007/978-1-4899-0502-4_1.

[9] F. Rubino, A. Nisticò, F. Tucci, and P. Carlone, “Marine application of fiber reinforced composites: A review,” *Journal of Marine Science and Engineering*, vol. 8, no. 1. 2020. doi: 10.3390/JMSE8010026.

[10] J. Njuguna, *Lightweight Composite Structures in Transport: Design, Manufacturing, Analysis and Performance*. 2016. doi: 10.1016/C2014-0-02646-9.

[11] D. K. Rajak, D. D. Pagar, P. L. Menezes, and E. Linul, “Fiber-reinforced polymer composites: Manufacturing, properties, and applications,” *Polymers*, vol. 11, no. 10. 2019. doi: 10.3390/polym11101667.

[12] E. Hardesty, W. Goldsworthy, and H. Karlson, “Geodesic path length compensator for composite-tape placement head,” 3810805, May 14, 1974

[13] R. L. Anderson and C. G. Grant, “ADVANCED FIBER PLACEMENT OF COMPOSITE FUSELAGE STRUCTURES,” in *NASA Advanced Composites Technology Conference, Part 2*, 1991. Accessed: Feb. 09, 2023. [Online]. Available: https://ntrs.nasa.gov/citations/19930021675

[14] H. Ghiasi, D. Pasini, and L. Lessard, “Optimum stacking sequence design of composite materials Part I: Constant stiffness design,” *Composite Structures*, vol. 90, no. 1. 2009. doi: 10.1016/j.compstruct.2009.01.006.

[15] H. Ghiasi, K. Fayazbakhsh, D. Pasini, and L. Lessard, “Optimum stacking sequence design of composite materials Part II: Variable stiffness design,” *Composite Structures*, vol. 93, no. 1. 2010. doi: 10.1016/j.compstruct.2010.06.001.

[16] S. Setoodeh, M. M. Abdalla, and Z. Gürdal, “Design of variable-stiffness laminates using lamination parameters,” *Compos B Eng*, vol. 37, no. 4–5, 2006, doi: 10.1016/j.compositesb.2005.12.001.

[17] A. Brasington, C. Sacco, J. Halbritter, R. Wehbe, and R. Harik, “Automated fiber placement: A review of history, current technologies, and future paths forward,” *Composites Part C: Open Access*, vol. 6. 2021. doi: 10.1016/j.jcomc.2021.100182.

[18] A. Brasington, B. Francis, M. Godbold, and R. Harik, “A review and framework for modeling methodologies to advance automated fiber placement,” *Composites Part C: Open Access*, vol. 10. Elsevier B.V., Mar. 01, 2023. doi: 10.1016/j.jcomc.2023.100347.

[19] J. Halbritter, “Automation of Process Planning for Automated Fiber Placement,” Open Access Thesis, University of South Carolina, Columbia, 2020.

[20] A. Brasington, J. Halbritter, R. Wehbe, and R. Harik, “Bayesian optimization for process planning selections in automated fiber placement,” *J Compos Mater*, vol. 56, no. 28, pp. 4275–4296, Dec. 2022, doi: 10.1177/00219983221129010.

[21] R. Harik, C. Saidy, S. J. Williams, Z. Gurdal, and B. Grimsley, “Automated fiber placement defect identity cards: Cause, anticipation, existence, significance, and progression,” *International SAMPE Technical Conference*, vol. 2018-May, 2018.

[22] A. T. Noevere and C. S. Collier, “Mapping AFP manufacturing data from VCP to hypersizer for stress analysis and optimization,” in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2018. doi: 10.2514/6.2018-0228.

[23] A. Noevere, C. Collier, and R. Harik, “Integrated design and manufacturing analysis for automated fiber placement structures,” in *International SAMPE Technical Conference*, 2019. doi: 10.33599/nasampe/s.19.1500.

[24] S. Rajan *et al.*, “Experimental investigation of prepreg slit tape wrinkling during automated fiber placement process using StereoDIC,” *Compos B Eng*, vol. 160, 2019, doi: 10.1016/j.compositesb.2018.12.017.

[25] E. Oromiehie, B. G. Prusty, P. Compston, and G. Rajan, “Automated fibre placement based composite structures: Review on the defects, impacts and inspections techniques,” *Composite Structures*, vol. 224. Elsevier Ltd, Sep. 15, 2019. doi: 10.1016/j.compstruct.2019.110987.

[26] J. Halbritter, C. Sacco, A. Brasington, M. Kirkpatrick, R. Wehbe, and R. Harik, “EXPERIMENTAL INVESTIGATION OF THE VALIDITY OF AUTOMATED FIBER PLACEMENT DEFECT PREDICTIONS,” in *Composites and Advanced Materials Expo, CAMX 2021*, 2021.

[27] E. Oromiehie, B. G. Prusty, P. Compston, and G. Rajan, “Automated fibre placement based composite structures: Review on the defects, impacts and inspections techniques,” *Composite Structures*, vol. 224. 2019. doi: 10.1016/j.compstruct.2019.110987.

[28] R. Anay *et al.*, “An experimental investigation concerning the effect of AFP defects on progressive damage in CFRP coupons,” *Compos Struct*, vol. 279, 2022, doi: 10.1016/j.compstruct.2021.114725.

[29] S. Rajan *et al.*, "Experimental investigation of prepreg slit tape wrinkling during automated fiber placement process using StereoDIC," *Compos B Eng*, vol. 160, pp. 546–557, Mar. 2019, doi: 10.1016/j.compositesb.2018.12.017.

[30] A. Halbritter and R. Harper, "Big parts demand big changes to the fiber placement status quo," in *SME Composites Manufacturing*, Mesa, Arizona: SME, Mar. 2012.

[31] T. Rudberg, J. Nielson, M. Henscheid, and J. Cemenska, "Improving AFP Cell Performance," *SAE International Journal of Aerospace*, vol. 7, no. 2, pp. 317–321, Dec. 2014, doi: 10.4271/2014-01-2272.

[32] M. Assadi and T. Field, "AFP Processing of Dry Fiber Carbon Materials (DFP) for Improved Rates and Reliability," in *SAE Technical Papers*, SAE International, Mar. 2020. doi: 10.4271/2020-01-0030.

[33] M. H. Nguyen, P. Davidson, and A. M. Waas, "Experimental and numerical study on the tensile failure behavior of toughened-interlayer composite laminates with automated fiber placement (AFP) induced gap and overlap defects," *International Journal of Material Forming*, vol. 14, no. 1, 2021, doi: 10.1007/s12289-020-01581-w.

[34] A. J. Sawicki and P. J. Minguet, "Effect of intraply overlaps and gaps upon the compression strength of composite laminates," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, 1998, pp. 744–754. doi: 10.2514/6.1998-1786.

[35] H. M. Hsiao and I. M. Daniel, "EFFECT OF FIBER WAVINESS ON STIFFNESS AND STRENGTH REDUCTION OF UNIDIRECTIONAL COMPOSITES UNDER COMPRESSIVE LOADING," 1996.

[36] L. E. Turoski, “Effects of manufacturing defects on the strength of toughened carbon/epoxy prepreg composites,” 2000.

[37] K. Croft, L. Lessard, D. Pasini, M. Hojjati, J. Chen, and A. Yousefpour, “Experimental study of the effect of automated fiber placement induced defects on performance of composite laminates,” *Compos Part A Appl Sci Manuf*, vol. 42, no. 5, pp. 484–491, May 2011, doi: 10.1016/J.COMPOSITESA.2011.01.007.

[38] A. W. Blom, C. S. Lopes, P. J. Kromwijk, Z. Gürdal, and P. P. Camanho, “A theoretical model to study the influence of tow-drop areas on the stiffness and strength of variable-stiffness laminates,” *J Compos Mater*, vol. 43, no. 5, pp. 403–425, Mar. 2009, doi: 10.1177/0021998308097675.

[39] O. Falcó, J. A. Mayugo, C. S. Lopes, N. Gascons, and J. Costa, “Variable-stiffness composite panels: Defect tolerance under in-plane tensile loading,” *Compos Part A Appl Sci Manuf*, vol. 63, pp. 21–31, 2014, doi: 10.1016/j.compositesa.2014.03.022.

[40] X. Li, S. R. Hallett, and M. R. Wisnom, “Modelling the effect of gaps and overlaps in automated fibre placement (AFP)-manufactured laminates,” *Science and Engineering of Composite Materials*, vol. 22, no. 2. Walter de Gruyter GmbH, pp. 115–129, Mar. 01, 2015. doi: 10.1515/secm-2013-0322.

[41] J. P. Marrouzé, J. Housner, and F. Abdi, “THE 19TH INTERNATIONAL CONFERENCE ON COMPOSITE MATERIALS EFFECT OF MANUFACTURING DEFECTS AND THEIR UNCERTAINTIES ON STRENGTH AND STABILITY OF STIFFENED PANELS.”

[42] M. Lan, D. Cartié, P. Davies, and C. Baley, “Microstructure and tensile properties of carbon–epoxy laminates produced by automated fibre placement: Influence of a

caul plate on the effects of gap and overlap embedded defects," *Compos Part A Appl Sci Manuf*, vol. 78, pp. 124–134, Nov. 2015, doi: 10.1016/J.COMPOSITESA.2015.07.023.

[43] M. Lan, D. Cartié, P. Davies, and C. Baley, "Influence of embedded gap and overlap fiber placement defects on the microstructure and shear and compression properties of carbon-epoxy laminates," *Compos Part A Appl Sci Manuf*, vol. 82, pp. 198–207, Mar. 2016, doi: 10.1016/j.compositesa.2015.12.007.

[44] J. P. H. Belnoue *et al.*, "Understanding and predicting defect formation in automated fibre placement pre-preg laminates," *Compos Part A Appl Sci Manuf*, vol. 102, pp. 196–206, Nov. 2017, doi: 10.1016/J.COMPOSITESA.2017.08.008.

[45] Y. M. Elsherbini and S. v. Hoa, "Experimental and numerical investigation of the effect of gaps on fatigue behavior of unidirectional carbon/epoxy automated fiber placement laminates," *J Compos Mater*, vol. 51, no. 6, pp. 759–772, Mar. 2017, doi: 10.1177/0021998316655393.

[46] A. Marouene, P. Legay, and R. Boukhili, "Experimental and numerical investigation on the open-hole compressive strength of AFP composites containing gaps and overlaps," *J Compos Mater*, vol. 51, no. 26, pp. 3631–3646, Nov. 2017, doi: 10.1177/0021998317690917.

[47] W. Woigk, S. R. Hallett, M. I. Jones, M. Kuhtz, A. Hornig, and M. Gude, "Experimental investigation of the effect of defects in Automated Fibre Placement produced composite laminates," *Compos Struct*, vol. 201, pp. 1004–1017, Oct. 2018, doi: 10.1016/J.COMPSTRUCT.2018.06.078.

[48] C. Sacco *et al.*, "On the effect of manual rework in AFP quality control for a doubly-curved part," *Compos B Eng*, vol. 227, Dec. 2021, doi: 10.1016/j.compositesb.2021.109432.

[49] G. Rousseau, R. Wehbe, J. Halbritter, and R. Harik, "Automated fiber placement path planning: A state-of-the-art review," *Computer-Aided Design and Applications*, vol. 16, no. 2. CAD Solutions, LLC, pp. 172–203, 2019. doi: 10.14733/cadaps.2019.172-203.

[50] L. Li, X. Wang, D. Xu, and M. Tan, "A Placement Path Planning Algorithm Based on Meshed Triangles for Carbon Fiber Reinforce Composite Component with Revolved Shape," *International Journal on Control Systems and Applications*, vol. 1, no. 1, 2014.

[51] X. Wang, W. Zhang, and L. Zhang, "Intersection of a ruled surface with a free-form surface," *Numer Algorithms*, vol. 46, no. 1, 2007, doi: 10.1007/s11075-007-9118-y.

[52] L. Yan, Z. C. Chen, Y. Shi, and R. Mo, "An accurate approach to roller path generation for robotic fibre placement of free-form surface composites," *Robot Comput Integr Manuf*, vol. 30, no. 3, 2014, doi: 10.1016/j.rcim.2013.10.007.

[53] B. Shirinzadeh, G. Cassidy, D. Oetomo, G. Alici, and M. H. Ang, "Trajectory generation for open-contoured structures in robotic fibre placement," *Robot Comput Integr Manuf*, vol. 23, no. 4, 2007, doi: 10.1016/j.rcim.2006.04.006.

[54] M. Favaloro and D. Hauber, "Process and design considerations for the automated fiber placement process," in *International SAMPE Technical Conference*, 2007.

[55] K. Schueler, J. Miller, and R. Hale, “Approximate geometric methods in application to the modeling of fiber placed composite structures,” *J Comput Inf Sci Eng*, vol. 4, no. 3, 2004, doi: 10.1115/1.1736685.

[56] I. B. Land and K. Ulrich, “Design and Manufacture of Advanced Composite Aircraft Structures Using Automated Tow Placement Certified By,” 1996.

[57] N. M. Patrikalakis and T. Maekawa, *Shape interrogation for computer aided design and manufacturing*. 2010. doi: 10.1007/978-3-642-04074-0.

[58] Z. Gürdal, B. F. Tatting, and K. Chauncey Wu, “Tow-placement technology and fabrication issues for laminated composite structures,” in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005. doi: 10.2514/6.2005-2017.

[59] A. W. Blom, B. F. Tatting, J. M. A. M. Hol, and Z. Gürdal, “Fiber path definitions for elastically tailored conical shells,” *Compos B Eng*, vol. 40, no. 1, 2009, doi: 10.1016/j.compositesb.2008.03.011.

[60] S. Setoodeh, M. M. Abdalla, S. T. IJsselmuiden, and Z. Gürdal, “Design of variable-stiffness composite panels for maximum buckling load,” *Compos Struct*, vol. 87, no. 1, 2009, doi: 10.1016/j.compstruct.2008.01.008.

[61] M. W. Hyer and R. F. Charette, “Use of curvilinear fiber format in composite structure design,” *AIAA Journal*, vol. 29, no. 6, 1991, doi: 10.2514/3.10697.

[62] O. Stodieck, J. E. Cooper, P. M. Weaver, and P. Kealy, “Improved aeroelastic tailoring using tow-steered composites,” *Compos Struct*, vol. 106, 2013, doi: 10.1016/j.compstruct.2013.07.023.

[63] B. F. Tatting, Z. Gürdal, M. W. Hyer, E. R. Johnson, L. Librescu, and L. Meirovitch, "Analysis and Design of Variable Stiffness Composite Cylinders in Engineering Mechanics," 1998.

[64] Z. Zamani, H. Haddadpour, and M. R. Ghazavi, "Curvilinear fiber optimization tools for design thin walled beams," *Thin-Walled Structures*, vol. 49, no. 3, 2011, doi: 10.1016/j.tws.2010.08.002.

[65] D. C. Jegley, B. F. Tatting, and Z. Gürdal, "Optimization of elastically tailored tow-placed plates with holes," in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2003.

[66] M. Arian Nik, K. Fayazbakhsh, D. Pasini, and L. Lessard, "Surrogate-based multi-objective optimization of a composite laminate with curvilinear fibers," *Compos Struct*, vol. 94, no. 8, 2012, doi: 10.1016/j.compstruct.2012.03.021.

[67] Z. Gürdal, B. F. Tatting, and C. K. Wu, "Variable stiffness composite panels: Effects of stiffness variation on the in-plane and buckling response," *Compos Part A Appl Sci Manuf*, vol. 39, no. 5, 2008, doi: 10.1016/j.compositesa.2007.11.015.

[68] A. W. Blom, M. M. Abdalla, and Z. Gürdal, "Optimization of course locations in fiber-placed panels for general fiber angle distributions," *Compos Sci Technol*, vol. 70, no. 4, 2010, doi: 10.1016/j.compscitech.2009.12.003.

[69] B. C. Kim, K. Potter, and P. M. Weaver, "Continuous tow shearing for manufacturing variable angle tow composites," *Compos Part A Appl Sci Manuf*, vol. 43, no. 8, 2012, doi: 10.1016/j.compositesa.2012.02.024.

[70] S. Honda and Y. Narita, “Vibration design of laminated fibrous composite plates with local anisotropy induced by short fibers and curvilinear fibers,” *Compos Struct*, vol. 93, no. 2, 2011, doi: 10.1016/j.compstruct.2010.07.003.

[71] Z. Wu, P. M. Weaver, G. Raju, and B. Chul Kim, “Buckling analysis and optimisation of variable angle tow composite plates,” *Thin-Walled Structures*, vol. 60, 2012, doi: 10.1016/j.tws.2012.07.008.

[72] J. Yang and F. Stern, “A highly scalable massively parallel fast marching method for the Eikonal equation,” *J Comput Phys*, vol. 332, 2017, doi: 10.1016/j.jcp.2016.12.012.

[73] M. Bruyneel and S. Zein, “A modified Fast Marching Method for defining fiber placement trajectories over meshes,” *Comput Struct*, vol. 125, 2013, doi: 10.1016/j.compstruc.2013.04.015.

[74] E. Lemaire, S. Zein, and M. Bruyneel, “Optimization of composite structures with curved fiber trajectories,” *Compos Struct*, vol. 131, 2015, doi: 10.1016/j.compstruct.2015.06.040.

[75] C. J. Brampton and H. A. Kim, “Optimization of Tow Steered Fiber Orientation Using the Level Set Method,” in *10th World Congress on Structural and Multidisciplinary Optimization*, 2013.

[76] C. Waldhart, “Analysis of Tow-Placed , Variable-Stiffness Laminates,” *Thesis*, 1996.

[77] B. C. Kim, K. Hazra, P. Weaver, and K. Potter, “Limitations of fibre placement techniques for variable angle tow composites and their process-induced defects,” in *ICCM International Conferences on Composite Materials*, 2011.

[78] A. T. Noevere and C. S. Collier, “Design for Manufacturing Tool for Automated Fiber Placement Structures-Verification and Validation.”

[79] A. T. Noevere, C. S. Collier, R. Harik, and J. Halbritter, “Development of a Design for Manufacturing Tool for Automated Fiber Placement Structures.”

[80] M. Lee, S. Jones, A. Noevere, and B. Stier, “An Aerospace Structural Sizing Tool with Sectional Equivalent Beam Requirements,” in *AIAA SCITECH 2023 Forum*, Reston, Virginia: American Institute of Aeronautics and Astronautics, Jan. 2023. doi: 10.2514/6.2023-2593.

[81] E. H. Forman and S. I. Gass, “The analytic hierarchy process - An exposition,” *Oper Res*, vol. 49, no. 4, 2001, doi: 10.1287/opre.49.4.469.11231.

[82] J. R. R. A. Martins and S. A. (Simeon A. Ning, *Engineering design optimization*. Cambridge University Press, 2021.

[83] B. A. Conway, “A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems,” *J Optim Theory Appl*, vol. 152, no. 2, pp. 271–306, 2012, doi: 10.1007/s10957-011-9918-z.

[84] R. Chelouah and P. Siarry, “Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions,” *Eur J Oper Res*, vol. 148, no. 2, pp. 335–348, Jul. 2003, doi: 10.1016/S0377-2217(02)00401-0.

[85] K. Socha and M. Dorigo, “Ant colony optimization for continuous domains,” *Eur J Oper Res*, vol. 185, no. 3, pp. 1155–1173, Mar. 2008, doi: 10.1016/j.ejor.2006.06.046.

[86] T. Ibaraki, “Integer programming formulation of combinatorial optimization problems,” *Discrete Math*, vol. 16, no. 1, 1976, doi: 10.1016/0012-365X(76)90091-1.

[87] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems for Optimization,” 1996.

[88] J. Paulus and A. Klapuri, “Music structure analysis using a probabilistic fitness measure and a greedy search algorithm,” *IEEE Trans Audio Speech Lang Process*, vol. 17, no. 6, pp. 1159–1170, 2009, doi: 10.1109/TASL.2009.2020533.

[89] C. Wilt, J. Thayer, and W. Ruml, “A Comparison of Greedy Search Algorithms,” 2010. [Online]. Available: www.aaai.org

[90] H. Weng *et al.*, “An enhanced greedy algorithm for failure resistant material design with application to composite delamination,” *Compos Struct*, vol. 278, Dec. 2021, doi: 10.1016/j.compstruct.2021.114681.

[91] M. Li, H. Zhang, J. Ma, S. Li, W. Zhu, and Y. Ke, “Greedy-based approach for generating anisotropic random fiber distributions of unidirectional composites and transverse mechanical properties prediction,” *Comput Mater Sci*, vol. 218, Feb. 2023, doi: 10.1016/j.commatsci.2022.111966.

[92] N. Kokash, “An introduction to heuristic algorithms,” 2005.

[93] S. Desale, A. Rasool, S. Andhale, and P. Rane, “Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey,” *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS*, vol. 2, pp. 296–304, 2015, [Online]. Available: http://www.ijcert.org

[94] Z. Beheshti, S. Mariyam, and H. Shamsuddin, “A Review of Population-based Meta-Heuristic Algorithm,” 2013. [Online]. Available: www.i-csrs.org

[95] G. Laporte and I. H. Osman, “Routing problems: A bibliography,” 1995.

[96] Y. He and A. J. Aref, “An optimization design procedure for fiber reinforced polymer web-core sandwich bridge deck systems,” *Compos Struct*, vol. 60, no. 2, pp. 183–195, 2003, doi: 10.1016/S0263-8223(02)00311-2.

[97] J. Wu and R. Burgueño, “An integrated approach to shape and laminate stacking sequence optimization of free-form FRP shells,” *Comput Methods Appl Mech Eng*, vol. 195, no. 33–36, pp. 4106–4123, Jul. 2006, doi: 10.1016/j.cma.2005.07.015.

[98] B. Liu, R. T. Haftka, M. A. Akg• Un, and A. Todoroki, “Permutation genetic algorithm for stacking sequence design of composite laminates,” 2000. [Online]. Available: www.elsevier.com/locate/cma

[99] G. Soremekun, Z. G• Urdal, R. T. Haftka, and L. T. Watson, “Composite laminate design optimization by genetic algorithm with generalized elitist selection,” 2001. [Online]. Available: www.elsevier.com/locate/compstruc

[100] X. Legrand, D. Kelly, A. Crosky, and D. Crépin, “Optimisation of fibre steering in composite laminates using a genetic algorithm,” *Compos Struct*, vol. 75, no. 1–4, pp. 524–531, Sep. 2006, doi: 10.1016/j.compstruct.2006.04.067.

[101] J. M. J. F. van Campen, C. Kassapoglou, and Z. Gürdal, “Generating realistic laminate fiber angle distributions for optimal variable stiffness laminates,” *Compos B Eng*, vol. 43, no. 2, pp. 354–360, Mar. 2012, doi: 10.1016/j.compositesb.2011.10.014.

[102] M. Rouhi, H. Ghayoor, S. v. Hoa, and M. Hojjati, “Effect of structural parameters on design of variable-stiffness composite cylinders made by fiber steering,” *Compos Struct*, vol. 118, no. 1, pp. 472–481, 2014, doi: 10.1016/j.compstruct.2014.08.021.

[103] X.-She. Yang, *Nature-inspired optimization algorithms*, First. Elsevier Inc., 2014.

[104] L. Le-Anh, T. Nguyen-Thoi, V. Ho-Huu, H. Dang-Trung, and T. Bui-Xuan, “Static and frequency optimization of folded laminated composite plates using an adjusted Differential Evolution algorithm and a smoothed triangular plate element,” *Compos Struct*, vol. 127, pp. 382–394, Sep. 2015, doi: 10.1016/j.compstruct.2015.02.069.

[105] C. M. C. Roque and P. A. L. S. Martins, “Maximization of fundamental frequency of layered composites using differential evolution optimization,” *Compos Struct*, vol. 183, no. 1, pp. 77–83, Jan. 2018, doi: 10.1016/j.compstruct.2017.01.037.

[106] T. Khajah and S. Natarajan, “Layup optimization of tow-steered composite laminates for maximum fundamental frequency and flutter speed using differential evolution,” *Compos Struct*, vol. 310, Apr. 2023, doi: 10.1016/j.compstruct.2023.116748.

[107] V. Ho-Huu, T. D. Do-Thi, H. Dang-Trung, T. Vo-Duy, and T. Nguyen-Thoi, “Optimization of laminated composite plates for maximizing buckling load using improved differential evolution and smoothed finite element method,” *Compos Struct*, vol. 146, pp. 132–147, Jun. 2016, doi: 10.1016/j.compstruct.2016.03.016.

[108] M. Akçair, M. Savran, L. Aydın, O. Ayakdaş, S. Öztürk, and N. Küçükdoğan, “Optimum design of anti-buckling behavior of graphite/epoxy laminated composites by differential evolution and simulated annealing method,” *Research on Engineering Structures and Materials*, 2019, doi: 10.17515/resm2019.66is0909.

[109] T. Vo-Duy, V. Ho-Huu, H. Dang-Trung, and T. Nguyen-Thoi, “A two-step approach for damage detection in laminated composite structures using modal strain energy method and an improved differential evolution algorithm,” *Compos Struct*, vol. 147, pp. 42–53, Jul. 2016, doi: 10.1016/j.compstruct.2016.03.027.

[110] R. Eberhart and J. Kennedy, “A New Optimizer Using Particle Swarm Theory,” 1995.

[111] L. A. Schmit and B. Farshi, “Optimum laminate design for strength and stiffness,” *Int J Numer Methods Eng*, vol. 7, no. 4, 1973, doi: 10.1002/nme.1620070410.

[112] S. Adali, A. Richter, and E. Verijenko, “Optimization of shear-deformable laminated plates under buckling and strength criteria,” 1997.

[113] R. Kathiravan and R. Ganguli, “Strength design of composite beam using gradient and particle swarm optimization,” *Compos Struct*, vol. 81, no. 4, pp. 471–479, Dec. 2007, doi: 10.1016/j.compstruct.2006.09.007.

[114] S. Suresh, P. B. Sujit, and A. K. Rao, “Particle swarm optimization approach for multi-objective composite box-beam design,” *Compos Struct*, vol. 81, no. 4, pp. 598–605, Dec. 2007, doi: 10.1016/j.compstruct.2006.10.008.

[115] S. N. Omkar, D. Mudigere, G. N. Naik, and S. Gopalakrishnan, “Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures,” *Comput Struct*, vol. 86, no. 1–2, pp. 1–14, Jan. 2008, doi: 10.1016/j.compstruc.2007.06.004.

[116] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. Narayana Naik, and S. Gopalakrishnan, “Quantum behaved Particle Swarm Optimization (QPSO) for

multi-objective design optimization of composite structures," *Expert Syst Appl*, vol. 36, no. 8, pp. 11312–11322, Oct. 2009, doi: 10.1016/j.eswa.2009.03.006.

[117] M. H. FarzanehKaloorazi, I. A. Bonev, and L. Birglen, "Simultaneous path placement and trajectory planning optimization for a redundant coordinated robotic workcell," *Mech Mach Theory*, vol. 130, pp. 346–362, Dec. 2018, doi: 10.1016/j.mechmachtheory.2018.08.022.

[118] Z. Liu, J. Lu, and P. Zhu, "Lightweight design of automotive composite bumper system using modified particle swarm optimizer," *Compos Struct*, vol. 140, pp. 630–643, Apr. 2016, doi: 10.1016/j.compstruct.2015.12.031.

[119] S. Van Der Walt *et al.*, "Scikit-image: Image processing in python," *PeerJ*, vol. 2014, no. 1, 2014, doi: 10.7717/peerj.453.

[120] E. H. Forman and S. I. Gass, "The Analytic Hierarchy Process-An Exposition," *Oper Res*, vol. 49, no. 4, pp. 469–486, 2001, doi: 10.1287/opre.49.4.469.11231.

[121] R. W. Saaty, "The analytic hierarchy process—what it is and how it is used," *Mathematical Modelling*, vol. 9, no. 3–5, pp. 161–176, 1987.

[122] R. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, 2nd ed. Hoboken, NJ: John Wiley & Sons, Inc., 2004. [Online]. Available: www.MatlabSite.com

[123] A. R. Parkinson, R. J. Balling, and J. D. Hedengren, *Optimization Methods for Engineering Design: Applications and Theory*. Brigham Young University, 2013.

[124] A. R. Parkinson, R. J. Balling, and J. D. Hedengren, "Optimization Methods for Engineering Design Applications and Theory," 2013.

[125] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011, doi: 10.1109/TEVC.2010.2059031.

[126] S. Das and P. N. Suganthan, “Differential evolution: A survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, Feb. 2011, doi: 10.1109/TEVC.2010.2059031.

[127] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” in *Proceedings of IEEE International Conference on Neural Networks*, 1995.

[128] İ. B. Aydilek, “A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems,” *Applied Soft Computing Journal*, vol. 66, pp. 232–249, May 2018, doi: 10.1016/j.asoc.2018.02.025.

[129] W. H. Lim and N. A. Mat Isa, “An adaptive two-layer particle swarm optimization with elitist learning strategy,” *Inf Sci (N Y)*, vol. 273, pp. 49–72, 2014, doi: 10.1016/j.ins.2014.03.031.

[130] K. Zielinski, D. Peters, and R. Laur, “Stopping Criteria for Single-Objective Optimization,” 2008.

[131] M. A. Albazzan, R. Harik, B. F. Tatting, and Z. Gürdal, “Efficient design optimization of nonconventional laminated composites using lamination parameters: A state of the art,” *Composite Structures*, vol. 209. 2019. doi: 10.1016/j.compstruct.2018.10.095.

[132] L. G. B. Jácome *et al.*, “An analysis framework for topology optimization of 3D printed reinforced composites,” in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, 2018. doi: 10.2514/6.2018-0093.

[133] J. A. Halbritter, R. Harik, C. Saidy, A. Noevere, and B. W. Grimsley, “Automation of AFP process planning functions: Importance and ranking,” in *International SAMPE Technical Conference,* Soc. for the Advancement of Material and Process Engineering, 2019. doi: 10.33599/nasampe/s.19.1592.

[134] C. M. Sacco, “Hybrid Theory-Machine Learning Methods for the Prediction of AFP Layup Quality,” 2021. [Online]. Available: https://scholarcommons.sc.edu/etd

[135] R. Harik, “Advancing Composites Manufacturing Towards a New Paradigm,” *SAMPE,* vol. 56, no. 6, pp. 6–14, Dec. 2020.