Implementation of combinatorial optimization Techniques for Automated Fiber Placement through thickness defect stack-up minimization

Noah C. Swingle1, Alex R. Brasington1, Joshua A. Halbritter1, Ramy Harik1

1University of South Carolina  
McNAIR Center, Department of Mechanical Engineering  
1000 Catawba Street  
Columbia, SC 29201

ABSTRACT

The Computer Aided Process Planning (CAPP) module was developed to facilitate and accelerate the process planning workflow for Automated Fiber Placement (AFP). CAPP assists process planners in identifying optimal starting point locations and layup strategies for each ply of a laminate. Ply optimization operates on measurement and scoring of geometry-based defects such as gaps, overlaps, angle deviation, and steering. This paper expands on the established framework for analyzing defect stack-up through thickness of a laminate. Four different combinatorial optimization algorithms are implemented and evaluated: (1) genetic algorithm, (2) differential evolution, (3) particle swarm, and (4) greedy search. The algorithms identify the optimal combination of ply-level layup strategies, by scoring potential laminates on defect stacking, using two different objective functions. A final optimization approach is also presented which trades some performance for a large gain in efficiency. These approaches are compared to a randomized combination using a complex tool surface in a virtual case study. The result is a streamlined methodology for comparing different laminate-level manufacturing strategies and minimizing the through thickness defect stack up.

Keywords: Automated Fiber Placement, Process Planning, Combinatorial Optimization, CAPP, AFP Defects

Corresponding author: Ramy Harik ([harik@cec.sc.edu](mailto:harik@cec.sc.edu))

# INTRODUCTION

Advanced composite materials continue to become more commonplace in their application to primary aerospace structures. The use of such materials can create a stronger, yet lighter structures with reduced maintenance requirements compared to traditional metallic counterparts. Automated Fiber Placement (AFP) has become a leader in the manufacturing of composite aerospace structures. The AFP process consists of a robotic or gantry-based system coupled with a fiber placement end-effector to additively manufacture a given structure. The method typically utilizes strips of pre-impregnated composite material called tows. These tows are laid on a tool surface in groups, called courses, until the desired shape is reached, constituting a single ply. This process is repeated, laying up ply-by–ply, until the full laminate is created. For thermoset materials, this constitutes the completed “green state” of the manufactured part, which will undergo a final curing process before it is finished. Due to interactions between the axial stiffness of fibers within the tows, underlying tool surface geometry, and limitations of the process, a variety of fiber defects are likely to occur during the placement of material. The main defects are gaps, overlaps, wrinkles, puckers, bridging, angle deviation, steering, folds, twists, wandering tows, loose tows, missing tows, position errors, and foreign objects. Harik et al. describes the cause, anticipation, existence, significance, and progression of these defects while summarizing some of their effects on individual plies and the overall laminate [1]. Gap, overlap, angle deviation, and steering defects are primarily caused by the curvature of the tool surface allowing them to be predicted prior to manufacturing during the process planning phase of AFP. Process planning develops optimal inputs and efficient machine processes based upon the working material, composite design, and manufacturing resources.

## Defect Interactions and Significance

Gaps and overlaps are the most common AFP defects and their effect on the internal geometry and mechanical properties of a laminate has been studied at length. Harik et al. attributes the cause of gaps and overlaps to purposeful steering during the AFP process and laying up on complex, doubly curved tool surfaces [1]. Sawicki and Minguet found that laminates containing overlaps and gaps saw compression strength reductions upwards of 27% and recommended accounting for this effect by reducing design allowables [2]. Depending on the size of the defect, out-of-plane waviness occurs in adjacent plies, which promotes damage initiation and causes the reduction in strength. It was shown that while any gap size induced a strength decrease, the rate of this decrease was more prominent for smaller gap sizes (0.030”) and comparatively consistent for larger gaps.

Woigk et al. built on these works by experimentally analyzing the impact of different configurations of stacked gaps and overlaps [3]. This work verified the general impact of gaps and overlaps on mechanical properties while also highlighting some of the complexity of cross-defect interactions and through-thickness defect distribution. They concluded that if the defect distribution within a laminate is random, the impact will be relatively minor. However, if the defect is systemic and occurs repeatedly through thickness, there will be a knockdown in laminate strength as compared to pristine materials. Belnoue et al. augmented the domain knowledge by coupling a consolidation model to a curing model, thereby capturing the full fabrication process in a multi-scale, multi-physics model [4]. Figure 1 shows a sample from their experiments with a comparison between an actual manufactured sample and their FE model at different states. While testing their model on different combinations of defects and tooling, the authors noted a 10% difference in laminate thickness between the all-overlaps, soft tooling laminate and the all-gaps, hard tooling laminate.

Diagram

Description automatically generated

Figure . Gaps and overlaps specimen showing a) ply book, b) CT scan after consolidation, c) initial geometry for finite element (FE) analysis, and d) internal ply geometry of the FE model after forming [4].

These works highlight a few key takeaways when it comes to evaluating the cumulative effect of defects within a laminate. Gaps and overlaps induce out-of-plane fiber waviness and thickness variations which increase interlaminar shear stress, reducing the overall mechanical performance of a laminate. This impact is greatly affected by factors including ply thickness, stacking sequence, cure tooling, as well as defect size, distribution, and type. One of the most notable factors here is that when defects are staggered, or randomly distributed within a laminate, the mechanical properties approach nominal, pristine values. It is this behavior of defects that guides the laminate optimization approaches presented in this paper.

## Computer Aided Process Planning Software

The Computer Aided Process Planning (CAPP) software was developed to improve and accelerate the process planning workflow for AFP. Currently, CAPP assists process planners in identifying optimal starting point location and layup strategy for each ply of a laminate. The Ply Level Optimization (PLO) phase functions on the measurement and scoring of geometry-based defects such as gaps, overlaps, angle deviation, and steering. The algorithms behind this optimization have been presented in multiple publications by the authors [5]–[9]. Figure 2 shows a flow chart that summarizes functionality of CAPP and highlights the connection to CGTech’s VERICUT Composite Programming (VCP). This work extends the analysis with a Laminate Level Optimization (LLO) phase which identifies the best combination of ply level inputs to create a complete laminate. Previous work has developed the global scoring metrics for laminate level defect stacking, which will be expanded here with other scoring metrics [10].

LLO searches through different combinations of ply scenarios and produces a laminate scenario with minimal defect stacking. Section 2 describes the workflow inputs and initial data operations. Section 3 details the different metrics used to score laminate scenarios which are paired with the combinatorial optimization algorithm described in Section 4. A virtual case study is presented in Section 5 showcasing the effect of LLO on through-thickness defect stacking for a laminate with complex curvature.

Diagram

Description automatically generated

Figure . CAPP data workflow.

# Analysis Approach

The modeling of toolpaths and the accompanied prediction of gaps, overlaps, angle deviation, and steering is performed via the communication between CAPP and VCP. The result of this analysis is the identification and location of toolpaths and defects within individual plies. To initiate the fiber coverage computation, the general laminate specifications are imported into VCP. The laminate specifications indicate the extent of material coverage and primary fiber orientation for each ply. During the fiber coverage computation within each of the designated ply scenarios, the specific course paths are defined which are consistent with the fiber orientation and layup strategy.

The computed paths can be used to virtually reproduce the fiber placement and resulting geometry of the individual tows. The calculation of defects directly follows the virtual reproduction of tow paths [11]. The tow geometry was utilized to compute the area defects; gaps and overlaps (Figure 3(a)). It is important to note that gaps and overlaps were only computed between neighboring courses and not between individual tows within each course. This would require more advanced tow draping and deformation modeling. The resulting course to course overlaps were exported from VCP as 3D polygon boundaries defining each defect instance as a unique closed contour. Fiber angle deviation and steering are defined on a regular grid over the tool surface (Figure 3(b)). It is important to note that VCP has staggering functionality to avoid stacking of defects between plies with the same starting point location. This functionality is not as useful when considering the inputs to each ply are being optimized and the starting points can be in varying locations even when plies are the same shape. Therefore, selection of individual ply scenarios is required rather than the simple staggering of starting points.

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | A picture containing text, sky, traffic, different  Description automatically generated |
| (a) | (b) |

Figure . Diagrams showing (a) gaps and overlaps and (b) angle deviation.

## Laminate Scenarios

Once the best ply scenarios have been identified, the process planner must combine these to form a full laminate scenario. It is recommended that before beginning the laminate level analysis and optimization, any undesirable ply level scenarios are removed since the LLO step will have access to all scenarios available within a given ply. Consequently, the larger the number of ply scenarios in each ply, the more likely the solution will be slower in terms of convergence. This is due to a larger input space which creates more possible combinations of ply scenarios. Using five to ten scenarios in each ply has shown to lead to adequate optimization run times.

The creation of the laminate scenario focuses specifically on defect stacking rather than total defect area. This is because it is assumed that total defect area has been optimized in the previous ply optimization step [6]. The final laminate score is a combination of multiple sub-scores and attempts to describe the overall frequency and severity of defect stacking. The following sections will outline the methodology for analyzing and scoring laminate scenarios.

## Parametric Discretization of Defects

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. 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 CAD software packages. For creating 3D surfaces, it utilizes a 2D-to-3D mapping, . 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, 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 then projected onto the tool surface obtaining their 2D representations in the parametric domain. The latter is then subdivided into a regular rectangular grid for the discretization step. The area enclosed within the projected defect polygon is then 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 4. With the newly parametrically discretized defects, the evaluation of stacked defects can be performed efficiently for laminate scenarios of varying size.

Chart

Description automatically generated

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

# Objective Functions

## Identifying and Measuring Stacked Defects

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 areas. 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 desired scale of comparison, but in general they are stacked on top of each other creating a 3D array modeling the entire defect distribution within a laminate. Summing through the third dimension of this array adds all the through-thickness defects and reveals areas on the tool surface which are subject to high stacking. This process is summarized in Figure 5.

A picture containing table

Description automatically generated

Figure . Schematic of the process for identifying and measuring stacked defects.

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 5 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. The scores include a threshold, which corresponds to the level at which the process planner is concerned about defect stacking. Choosing a threshold requires some judgement, for instance, if the desired part requires tight geometrical constraints a lower threshold may be chosen.

This levels concept is primarily used in the global scoring method described in Section 3.1.1. “Global” refers to the way defects in all plies of the laminate are compared to each other. With this approach, a defect on ply 1 would still stack with a defect on ply 100 despite the 98 intermediate plies. While this may not realistically represent the way defects interact within a laminate, this approach is useful for efficient optimization purposes. The local scoring method takes all of the individual ply-level defect arrays as an input and windows through them instead of summing through-thickness. With this approach, defect interaction is analyzed on a more local scale (i.e., defect stacking is limited to a few plies above and below the ply being analyzed rather than the entire laminate).

### Global Scoring Method

Two metrics are used to globally evaluate defect stacking: frequency and severity. Frequency describes the percentage of defect area stacked beyond a chosen level threshold relative to the total through-thickness defect area on the laminate. Starting at the threshold level, defect level area is summed and then divided by the total defect area. Defect level area is defined by the number of cells belonging to each level. While a simple cell count is used for the optimization, it is possible to calculate an approximate area for each level for user analysis purposes. Frequency is mathematically defined by:

|  |  |
| --- | --- |
|  | () |

In Equation (1), is the total number of defect levels in the laminate and is the number of cells in the level. A frequency score of one denotes the case where all defects are stacked beyond the chosen threshold. If the number of defect levels falls below the threshold, a frequency score of zero is defaulted. The second metric, severity score, is similar to the frequency score in that it compares defect levels above the threshold to total defect 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. Severity score is mathematically defined as:

|  |  |
| --- | --- |
|  | () |

The main addition here is the squared term in the numerator which attempts to weight the score with the magnitude of variation from the threshold and results in a quadratic behavior. This is balanced similarly in the denominator of the fraction. The metric is sensitive to different distributions of defect area stacked beyond the threshold and will penalize laminate scenarios with more severe defect stacking. 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 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.

### 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 [12]. 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:

|  |  |
| --- | --- |
|  | () |

Here represents the probability of a discrete random variable 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. It can be thought of as a window that determines how far in-plane and through the thickness defect interaction is considered. In this case, it is a rectangular cuboid of size , where and are the in-plane directions and determines how many plies through-thickness defect interaction is considered.

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) to have 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. 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. This value can then be compared amongst others to determine the stacking of defects.

### Gap and Overlap Score Combination

Both the global and local metrics score laminates based on their gap and overlap distributions separately. For the local metric, 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 metric, there are four sub-scores corresponding to gap frequency, gap severity, overlap frequency, and overlap severity. These scores are combined using an analytic hierarchy process (AHP) matrix where the user defines pairwise comparisons between individual sub-scores and the matrix transforms these weights into rankings for the final combination [13], [14]. This customization allows process planners to achieve very specific defect distributions within their laminate corresponding to their own preferences and part requirements. In both cases a higher score is better and denotes less severe defect stacking.

### Generalized Objective Function

The scoring metrics described are used as objective functions when coupled with the combinatorial optimization algorithms presented in Section 4. The algorithms constitute different ways of generating candidate solutions which are then evaluated using the objective function. The optimization problem is then defined by:

|  |  |
| --- | --- |
|  | () |

where is the objective function, is a candidate ply level scenario, is the number of plies, where is the number of ply scenarios to be considered, and is the collection of feasible inputs. Note that is required to have a single ply scenario from each ply. The behavior of is then defined by the selection of evaluation strategy (global or local).

# Optimization Methodologies

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 frequency and severity or entropy 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 . This creates the need for efficient combinatorial optimization algorithms to search through that design space and converge on an optimal manufacturing strategy.

## Greedy Search

A Greedy Search (GS) is first 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 6. 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 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 (4). 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 plies resulting in the selection of ply scenarios to make up the entire laminate. The GS algorithm requires 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.

A picture containing icon

Description automatically generated

Figure . Greedy search algorithm for laminate optimization.

## Evolutionary Algorithms

### 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 genetic algorithm (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 , with a length () equal to the population size and a width () 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.

|  |  |
| --- | --- |
|  | () |

Each row of is a chromosome with each gene () being a continuous variable in the range [0, 1] where denotes variable n in chromosome m. can also be written in a shorthand notation as a list of vectors . The continuous variables 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 (6) and (7) respectively where and are the lower and upper bounds of .

|  |  |
| --- | --- |
|  | () |
|  | () |

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 [15]. 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 (8) where variable in the child chromosome is equal to the variable from the first parent () if the mask has a value of 0, and vice versa from the second parent (). Doing this for the total number of variables in the chromosome results in the full child chromosome.

|  |  |
| --- | --- |
|  | () |

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 () is calculated with Equation (9) where () is a randomly selected number between 0 and 1.

|  |  |
| --- | --- |
|  | () |

The value used in Equation (9) is defined below in Equation (10) where is the current generation number, is the total number of generations, and is a user defined parameter. Further details on the effect of and can be found in parameters [16].

|  |  |
| --- | --- |
|  | () |

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 elitism 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. For this study, a max number of generations was used as the termination criterion.

### Differential Evolution

Differential evolution (DE) is a type of evolutionary algorithm belonging to stochastic population-based evolutionary methods and uses a population of candidate solutions [17]. DE operates on a population of size and involves two stages. The first is initialization which generates an initial population . 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 , scenarios from each ply (parameter bounds), population size , mutation factor , crossover probability , maximum iterations , and convergence tolerance . 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 (4) and the best vector 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 in the range ] and another random integer in the range . Each parameter in the vector is then mutated according to:

|  |  |
| --- | --- |
|  | () |

where represents a vector in the current population and 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. The candidate population is then evaluated and the best solutions are selected to create the next population. Also, the current best vector is updated if necessary. Again, for this study a max number of generations is used as termination criteria.

### Particle Swarm

Particle Swarm (PS) optimization was initially developed by Kennedy and Eberhart, and it mimics the behavior of bird and fish swarms [18]. 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. The steps of this algorithm are summarized below.

Initialization of the algorithm consists of generating a defined number of particles and randomly setting their initial positions () and velocities (). 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 (12) and (13) [19], [20].

|  |  |
| --- | --- |
|  | () |
|  | () |

In these equations, is an inertial weight parameter, and are random numbers in the range (0, 1), and are acceleration coefficients, is the particles best position, and 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 , , and will influence the accuracy of the resulting solution. Also, as the dimensions of a problem increases, the optimization will become more complex and the probability of finding the global optimum decreases.

## Global Hot Spot Detection

The final optimization method presented is slightly distinct from the approach discussed thus far. Due to the nature of the combinatorial optimization algorithms, numerous objective function evaluations must be performed to achieve convergence. All this computation takes time, especially as the number of plies and ply scenarios increases. Global Hot Spot Detection (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 frequency, severity, and entropy. Once all of 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 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, similar to the threshold in frequency/severity. 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 7.

A picture containing text, wall, indoor

Description automatically generated

Figure . Illustration of the GHSD approach.

The scoring equation in Figure 7 is represented mathematically as:

|  |  |
| --- | --- |
|  | () |

Here, is the binary array associated with defects in a given ply scenario. 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 (14). 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 7, 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 calculations.

# Virtual Case Study

## Experimental Setup

Virtual case studies were performed on a doubly curved tool surface with a 20-ply laminate. The surface (Figure 8) measures approximately 2.0m in length and 0.85m at the widest point. A stacking sequence was randomly generated from 0°, +45°, and -45° fiber angles, excluding 90° since the curvature didn’t produce any area defects for this angle. A constant ply boundary was used for every ply. Five ply scenarios were generated for each ply with a random starting point and rosette path geometry. The rosette strategy generated the most defects and creates a worst-case scenario for defect stacking. This resulted in the creation of 100 ply scenarios which were then processed through VCP to generate fiber paths and compute geometrical fiber defects. Courses were generated according to the ply scenario parameters, 8 individual ¼” (6.35mm) tows per course, and default VCP values.

Logo

Description automatically generated

Figure . Complex tool surface used for virtual case study with ply area (blue).

The data from the randomized design space of 20 plies, 5 scenarios per ply, was given to the four different combinatorial optimization algorithms (GS, DE, GA, PS) and evaluated using the two different objective functions (global and local). Each of the evolutionary optimization algorithms works by creating a population of candidate laminate scenarios, evaluating them, and then creating a new population using information from the previous generation. For this study, total generations for each algorithm were limited to 50, and each algorithm was run 10 times. The first set of trials used global score as the objective function, with the second set using local score. Note that convergence criteria have not been implemented and the optimizations are allowed to run to the max number of generations. For both the global and local score optimizations, a lower score is better and indicates less gap and overlap defect stacking. The global score is scaled from [0-1] and the local score values depend on the size of the tool surface and resolution of the discretization used but were roughly [1.5-6] for this virtual case study.

## Global Score Trials

Figure 9 shows the progression of each of the algorithm iterations with the X-axis showing generations and the Y-axis showing the best global score of that generation. A dashed line is superimposed on each graph showing the average best score found by each algorithm after 50 generations. The dotted line represents the score of the best laminate found by GS. The scoring threshold was set at one and an equal weighting was used for gap/overlap frequency and severity to calculate the final combined global score.

Chart

Description automatically generated

Figure . Convergence graphs for DE, GA, and PS using the global score.

The first notable outcome of the global trials was the underperformance of the GS algorithm. All the other algorithms outperformed GS on every iteration after just 16 generations. The GS laminate scenario’s global score was 0.173 compared to average scores from DE, GA, and PS of 0.145, 0.143, and 0.155 respectively. The DE and GA operated very similarly, converging quickly on laminate scenarios with scores between 0.140 and 0.150. Both algorithms improved rapidly initially before plateauing and rarely finding a better laminate scenario after 35 generations. PS was the second worst performer overall and most distinct among the evolutionary algorithms. It progressed slower and converged faster than the DE and GA, with none of the ten iterations finding a better laminate scenario after 22 generations. Another notable characteristic of the PS trials was the wide range of final scores. While DE and GA had very consistent final laminate scores, the PS had scores as low as 0.146 (nearly as low as DE) and as high as 0.171 (higher than some of the random laminate scenarios). The time per generation for each evolutionary algorithm was consistent with DE, GA, and PS averaging 6.48, 6.46, and 6.57 seconds respectively. This is due to the objective function evaluations dominating the computation time with differences in internal algorithm operations not adding significant time. The GS algorithm took a total of 396.19 seconds to run, which is longer than any of the individual evolutionary algorithm trials.

It's clear that GS does not perform as well as the evolutionary algorithms in terms of efficiency or fitness. This is likely due to the interactions between the global scoring metric and the way GS operates. GS starts at the first ply proceeds sequentially, choosing the scenario that results in the laminate with the lowest global score. For the beginning plies, the algorithm has no information on where the defects will be for subsequent plies. It likely made poor decisions early on and improved as more ply scenarios were chosen and more data was available. Among the evolutionary algorithms, DE and GA performed very similarly while PS lagged behind. In addition to finding worse laminate scenarios on average, there was a significantly larger range in final scores with PS. This suggests that the algorithm is getting stuck on local minimums and may be improved with further parameter tweaking. The overlap stack-ups for the best laminate scenarios found by DE, GA, PS, GS, and the random laminate scenario are shown in Figure 10(a)-(e) respectively. These images are 2D projections of all the overlap defects within the 20-ply laminate scenario. Defects are color scaled from dark to light blue to yellow indicating increased severity of defect stacking (i.e., one defect through thickness is dark blue, two is light blue, and so on). Comparing each of the optimized laminates to the random laminate scenario, there is a clear progression from a more condensed defect distribution towards more dispersed distributions. It is also possible to see a slight improvement from the GS and PS laminates to the DE and GA laminates.

|  |
| --- |
| Text  Description automatically generated |
| (a) |
| A screenshot of a computer  Description automatically generated with medium confidence |
| (b) |
| Text  Description automatically generated with medium confidence |
| (c) |
| A picture containing text  Description automatically generated |
| (d) |
| Diagram  Description automatically generated |
| (e) |

Figure : Overlap stack-up for best laminate scenario found by (a) DE, (b) GA, (c) PS, (d) GS, and (e) random with global score

## Local Score Trials

The above experiment was repeated using entropy (local) as the objective function instead of frequency and severity (global). From the literature it was determined that gap and overlap defects induce fiber waviness for about 2-3mm in-plane and about 2 plies through-thickness, though this is heavily dependent on stacking sequence, material properties, cure tooling, and several other factors. For this tool surface, and the resolution of the defect discretization employed, a window size of 11x11x5 cells was used for the entropy calculation. Figure 11 shows the results of these trials with the same three progression graphs for DE, GA, and PS with lines superimposed for the average final score of each algorithm and the GS final score.

A picture containing chart

Description automatically generated

Figure . Convergence graphs for DE, GA, and PS using local score.

Using the local score as the objective function greatly changed the behavior and outcome of the combinatorial optimization algorithms. The first notable difference is a protraction of convergence across the board. DE and GA flattened out only at the very end of 50 generations, compared with 35 generations for global score. PS still converged quicker than the others, but also took longer with local score rather than global. Another interesting difference is the improvement in performance of the GS algorithm. For local score, GS outperformed the average score of the best laminates found by DE and PS while coming very close to the GA. The average best laminate scenario local scores found by DE, GA, PS, and GS were 2.722, 2.655, 2.781, and 2.702 respectively. DE progressed gradually, with an almost linear march until 30 generations when improvements were found more slowly. The final scores of each DE trial were also remarkably similar and suggest that parameters may need to be tweaked to increase the algorithm’s ability to find better laminate scenarios. The GA had a more pronounced progression initially, followed by slow improvement for the final 30 generations. PS started similarly to the GA but tended to level off more completely, failing to find many better laminates in second half of the generations. The time per generation for each of the evolutionary algorithms was also consistent between them and again points to the objective function evaluations dominating the computation time. The average time per generation for DE, GA, and PS was 44.28, 45.21, and 45.43 seconds respectively. GS took 2108.48 seconds to complete.

Perhaps the most interesting result of the local trials was the improvement of the GS algorithm. Considering the difference between the two objective functions and how GS works this isn’t a wholly surprising outcome. The global score considers every ply in the laminate at once, making it difficult for the algorithm to make good decisions early on when it is only considering a few plies. The local score only looks at defect interaction within a set range of plies, meaning it always has at least half of the information needed to make an informed decision. In other words, the global score is affected by interaction between plies 2, 3, and 17 equally whereas the local score is only affected by interaction between plies 2 and 3, not 2 and 17 or 3 and 17. This gives GS an advantage when paired with the local score over the global score. Another interesting result is the extension of convergence and increased likelihood for each algorithm to find smaller, more frequent improvements. This points to the increased complexity and sensitivity of the local score. Not only does the local score address through thickness defect interaction but also in-plane. This enables it to discern between two laminate scenarios that might have the same magnitude of through thickness defect stacking but different in-plane distributions. Looking at time efficiency, the local score trials took about seven times as long to run as the global score trials. Due to this increased time per generation, the GS algorithm produces an optimized laminate quicker than any of the evolutionary algorithms on average. Because of this time savings, and its increased performance, the GS algorithm was best suited for local score optimization.

Figure 12(a)-(d) shows the overlap stack-ups for the best laminate scenario found by DE, GA, PS, and GS respectively. It’s important to note that while these images are useful for seeing through thickness defect stack-up, they are much better suited for the global score as they show the entire laminate rather than a smaller window of plies. Nevertheless, there is still a clear progression from the random laminate shown in Figure 10(e) to the locally optimized laminates. It is also possible to identify the GA laminate as the most evenly distributed laminate scenario. While the DE and GS laminates may appear more stacked, this could be the result of a more global stack-up of overlaps rather than a local interaction.

|  |
| --- |
| A picture containing text  Description automatically generated |
| (a) |
| Text  Description automatically generated |
| (b) |
| A picture containing text  Description automatically generated |
| (c) |
| Text  Description automatically generated with low confidence |
| (d) |

Figure : Overlap stack-up for best laminate scenario found (a) DE, (b) GA, (c) PS, and (d) GS with local score

## Global Hot Spot Detection

The GHSD method is separate from the previously discussed approaches because it is not an objective function to be paired with a combinatorial optimization algorithm. GHSD is a way of scoring the ply scenarios based on their likelihood of stacking within the design space of possible laminate scenarios. After all the ply scenarios are scored using GHSD, the lowest scoring ply scenario (indicating least amount of stacking) from each ply is chosen for the laminate scenario. This greatly reduces the amount of computation necessary and subsequently the amount of time needed to find a suitable laminate. Given the same randomized ply-level design space as the previous trials, GHSD produced a laminate in 20.55 seconds, a drastic reduction from the combinatorial algorithms. The final laminate is shown in Figure 13. While the algorithm was efficient, it was a worse performer than the evolutionary algorithms. Scoring the laminate with the global score (which is most suitable for this approach) reveals a final score of 0.207.

Text

Description automatically generated

Figure . Overlap stack-up for the best laminate found by GHSD.

## Comparisons

In terms of efficiency, the GHSD vastly outperforms the other approaches. While the combinatorial optimization algorithms don’t take a lot of time to perform their internal operations, the objective functions are slow to compute, and with a population size of ten and 50 generations per iteration, the number of evaluations adds up. Improvements in the way the objective functions operate could certainly be made, but the nature of GHSD requiring significantly less evaluations means it will likely always outperform the other methods in terms of efficiency. Comparing the best laminate found by global, local, and GHSD with the random laminate scenario reveals somewhat different results. Zooming in on the densest defect area on the tool surface, Figure 14 shows the overlap stack-up for the best laminate from each approach. From left to right there is a clear progression from more spread-out defects towards more condensed, stacked defects. The best laminate scenarios found using global and local score both came from the GA and appear very similar. It is important to note that while laminate (d) appears to have less defect area, this is because the defects are more severely stacked and less spread out than the optimized laminates. All the optimized laminates clearly feature less stacked defects in comparison to the random laminate, and the evolutionary laminates appear to outperform the GHSD laminate.

A picture containing chart

Description automatically generated

Figure . Zoomed overlap stack-up for best laminate found using a) global score, b) local score, c) GHSD, and d) the random laminate scenario.

Looking at the best global (Figure 14(a)) and local (Figure 14(b)) laminate scenarios from Figure 14 it appears that the locally optimized laminate has less overall defect stacking and a more evenly dispersed distribution. Table 1 shows the distribution of through thickness overlap defect stacking for the best globally and locally optimized laminates. The 1 defect column shows the number of cells of the tool surface array that have 1 overlap defect and no stacking. The 2 defects column represents two overlap defects stacked on top of each other through thickness, and so on. From the table it becomes clear how these different objective functions push the algorithms towards different laminate scenarios. The global laminate had less area with three or more defects stacked through thickness but more area with one or two defects. This is because the global score penalizes higher levels of defect stacking through the severity metric outlined in Section 3.1.1. The locally optimized laminate featured more defect area at the higher levels of stacking likely because the three or four plies with those stacked defects were separated by several intermediate plies. This doesn’t mean that the locally or globally optimized laminate is better, they just aren’t directly comparable.

Table . Overlap stack-up data for globally and locally optimized laminates.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1 Defect** | **2 Defects** | **3 Defects** | **4 Defects** |
| **Global Laminate (a)** | 262112 | 65988 | 6792 | 17 |
| **Local Laminate (b)** | 260849 | 65069 | 7126 | 56 |
| **Percent Difference** | -0.48% | -1.4% | +4.9% | +229.4% |

The average and overall best score found by each approach, as well as the total runtime is summarized in Table 2 below. Looking at the global score data, it’s clear that the evolutionary algorithms were able to find better laminates in less time than GS. However, GHSD was able to find a laminate scenario significantly better than a random laminate in a fraction of the time. The local score data tells a very different story. Due to the increased computation time of the objective function, and the nature of GS, the evolutionary algorithms were largely outperformed. The GS was able to produce a laminate scenario that was better on average than either of the DE or PS laminates in less time. While the GHSD approach is more geared towards global optimization, the local score is still significantly better than a random laminate scenario and took just 1% of the time of the other optimization algorithms.

Table . Summary of optimization results.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Global Score** | | | **Local Score** | | |
|  | Average | Best | Total Time | Average | Best | Total Time |
| **DE** | 0.145 | 0.143 | 323.90s | 2.722 | 2.701 | 2214.05s |
| **GA** | 0.143 | 0.142 | 323.25s | 2.655 | 2.594 | 2260.65s |
| **PS** | 0.154 | 0.145 | 328.55s | 2.781 | 2.683 | 2271.80s |
| **GS** | N/A | 0.173 | 396.19s | N/A | 2.702 | 2108.48s |
| **GHSD** | N/A | 0.207 | 20.55s | N/A | 3.287 | 20.55s |
| **Random** | N/A | 0.404 | N/A | N/A | 4.344 | N/A |

# Conclusion

The work presented above represents a robust framework for approaching laminate optimization and defect stacking from a process planning perspective. Two different scoring systems were paired with four different combinatorial optimization algorithms showcasing several options for minimizing the total defect stack-up area within a laminate. The global score allows process planners to analyze total through thickness defect stacking while the local score takes a more restricted approach and also addresses in-plane defect distribution. For global optimization the GA produced the best laminate in the least amount of time. For local optimization the GS algorithm performed much better and was able to find similar or better laminates in less time than the evolutionary algorithms. In either case, GHSD was able to produce an acceptably optimized laminate scenario in a fraction of the time. It is important to note however that the total time for the evolutionary algorithms is slightly misleading, as many of the global trials converged much earlier than the full 50 generations. Compared with the random laminate scenario, all optimized laminates showed clear improvement with more evenly distributed defect distributions. While 2D images were included throughout the majority of this paper, these images can also be projected back onto the tool surface to identify specific locations of defect stacking. An example of this is shown in Figure 15. The impact of this optimization is the possibility for increased design allowables, eliminating knockdowns for gap and overlap defects. This will allow designers to achieve more efficient designs, saving manufacturing time and material cost.

Chart

Description automatically generated with medium confidence

Figure . A random laminate scenario with overlap stack-up visualized on the tool surface.

## Future Work

Currently, LLO treats gap and overlap defects separately and does not account for cross-defect-type interactions. Future work needs to address these interactions and create a more holistic methodology for laminate optimization. LLO also only optimizes laminates for gap and overlap defects, not any of the other multitude of defects possible with AFP. Future work will also include the implementation of convergence criteria for all of the algorithms. While forcing them to run for 50 generations was useful for this paper, in practice this number could be reduced to improve efficiency. Another natural next step of this work is utilizing the defect stack-up data to generate a green state thickness model of the part. This could be coupled with a curing model to predict final part tolerances and more accurate mechanical behavior. Finally, physical trials comparing manufactured laminates with and without LLO are planned for the future to validate this approach. These trials will be conducted using industrial scale AFP machines like the ones shown in Figure 15.

|  |  |
| --- | --- |
| A picture containing indoor, miller  Description automatically generated |  |
| (a) | (b) |

Figure . AFP machines at (a) NASA Langley [21] and (b) USC McNAIR Center.

# Acknowledgements

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.

# References

[1] R. Harik, Z. Gurdal, C. Saidy, S. J. Williams, and B. Grimsley, “Automated Fiber Placement Defect Identity Cards: Cause, Anticipation, Existence, Significance, and Progression,” in *SAMPE 2018*, 2018. Accessed: Mar. 24, 2020. [Online]. Available: https://www.researchgate.net/publication/326464139

[2] A. J. Sawicki and P. J. Minguet, “Effect of intraply overlaps and gaps upon the compression strength of composite laminates,” in *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 1998, vol. 1, pp. 744–754. doi: 10.2514/6.1998-1786.

[3] 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.

[4] 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.

[5] A. Brasington, “Integration of Process Planning into the Automated Fiber Placement Design for Manufacturing Cycle,” University of South Carolina, Columbia, SC, 2021.

[6] A. Brasington, J. Halbritter, R. Wehbe, and R. Harik, “Bayesian optimization for process planning selections in automated fiber placement,” *J Compos Mater*, vol. 2022, no. 0, pp. 1–22, 2019, doi: 10.1177/00219983221129010.

[7] A. Brasington, C. Smith, J. Halbritter, R. Wehbe, and R. Harik, “Surrogate Based Methods for Rapid Starting Point Optimization in Automated Fiber Placement,” in *SAMPE 2022 Conference Proceedings*, 2022.

[8] J. A. Halbritter, “Automation of Process Planning for Automated Fiber Placement,” University of South Carolina, 2020.

[9] J. 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*, 2019, vol. 2019-May. doi: 10.33599/nasampe/s.19.1592.

[10] J. Halbritter, N. Swingle, R. Wehbe, and R. Harik, “MINIMIZING THROUGH THICKNESS DEFECT STACK-UP FOR AUTOMATED FIBER PLACEMENT OF COMPOSITE LAMINATES VIA FIBER PATH OPTIMIZATION,” 2022.

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

[12] “Entropy.” scikit-image: image processing in python, 2022.

[13] 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.

[14] 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.

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

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

[17] 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.

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

[19] 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.

[20] 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.

[21] S. McDonald, “Robot ISAAC will help NASA Langley Speed Toward Innovation,” *NASA*, Aug. 07, 2017.