Global/Local Iteration for Blended Composite Laminate Panel Structure Optimization Subproblems

David B. Adams[†], Layne T. Watson[‡], Omprakash Seresta[+], Zafer Gürdal[*]
Department of Computer Science[†]
Departments of Computer Science and Mathematics[‡]
Department of Aerospace and Ocean Engineering[+]
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 USA
Aerospace Structures Chair[*]
Delft Institute of Technology, The Netherlands
Contact e-mail: daadams3@vt.edu

Keywords—global/local, composite laminates, genetic algorithms, parallel computing, combinatorial optimization, decomposition, blending

Abstract—Composite panel structure optimization is commonly decomposed into panel optimization subproblems. Previous work applied a guide based design approach to the problem for a structure where the local loads were assumed to be fixed for each panel throughout the design process. This paper examines the application of guide based design to a more realistic representation of the structure where the local loads for each panel are determined through a global level analysis that is coupled with the stacking sequence for every design panel. A small problem is selected for which an exhaustive search of the subproblem design space verifies the optimality of the solution found through the global/local iteration process introduced in this work. The efficient discovery of solutions to these guide based design subproblems creates an opportunity to incorporate the solutions into a global level optimization process. A parallel genetic algorithm is proposed to control global optimization in which evaluating the fitness of each member of the population requires the solution of a guide based design subproblem where parallelism is solely within fitness evaluations. Results are presented for a wingbox design problem and compared with known solutions for the same problem to demonstrate weight reductions in a problem thought to already be near optimally solved.

1: Introduction

The design of fiber-reinforced composite laminates is a discrete optimization process involving the specification of material type and orientation of ply layers in the stacking sequence. For large structures, such as the design of a wing or fuselage, the design process is partitioned into panel level optimizations for which local improvements can be made in reducing global structure weight without compromising strength and buckling constraints at the local level. An example illustrating the primary reason for this partitioning is that the local loadings for the panels can vary widely from the root to the tip of a wing and any laminate design thick enough to satisfy loading constraints at the root of the wing will be much heavier than required at the wing tip. By designing the structure as multiple panels, each optimized as a minimal weight design satisfying local loading requirements, large gains can be made in the global reduction of weight from that of a single stacking sequence design covering the entire structure.

The initial approach to this problem involved the specification of a fixed load for each panel in the structure [1], [2], [3], [4]. Isolated local panel optimization results in stacking sequence orientations that vary widely between adjacent panels, which would cause serious manufacturing difficulties, generating the need for a globally blended solution. By fixing the local loads on each panel, the complexity of the global problem is greatly reduced, but the issue of blending the optimized local (panel) designs into a complete structure design that is manufacturable is far from trivial. Earlier work on this blending problem included a genetic algorithm (GA) method using the edit distance metric to allow a set of independently evolving panel populations to evolve to a blended global solution using reference migration [1], the use of sublaminate definitions and

design variable zones by Soremekun et al. [2], and the addition of continuity constraints proposed by Liu and Haftka [5]. The edit distance method utilizes a multiple-deme parallel GA approach that obtains blended designs through evolutionary pressures from neighboring populations. It is a nonstandard parallel decomposition that works asynchronously in real time to emulate semi-isolated populations with random migration to produce blended global designs from a pool of globally unconstrained local stacking sequence design possibilities. Evolutionary pressures are controlled through a user defined scaling factor that modifies the severity of penalties imposed for blending mismatches. These penalties, however, were found to hinder convergence to a global optimum by creating local optima in the search space that are artifacts of the algorithm itself [1]. The approach used in [2] is a two step procedure that relies on first optimizing the individual panels followed by identifying common thickness zones across multiple panels that are redefined and reoptimized using blended stacking sequences. Because of the heuristic nature of the approach, however, it is possible that suboptimal designs are generated, although blended designs are obtained with little weight penalty compared to unblended minimum weight structural design.

A more recent approach to this problem is a guide based design methodology [6]. In guide based design, the search space of the problem is implicitly reduced by forcing the generation of global (defining the entire structure) individuals that are always completely blended across panel boundaries. The particular focus of the present work is in extending the guide based design methodology to a more realistic representation of the problem where the local loadings for individual panels can only be determined through a global level analysis that is coupled with the exact stacking sequence for each panel in the structure. This extension creates a challenging subproblem to determine the fitness of a guide design given the additional coupling of panel designs. A global/local iterative process is presented here that quickly converges to a near optimal solution and with a small amount of sub-graph expansion the optimal solution for a given guide can be found. The solution to this subproblem appears to be sufficiently fast that it can be incorporated into a GA for global optimization of the structure where each member of the population is an instance of the subproblem for which a fitness value can be determined.

Section 2 provides some background on genetic algorithms. Section 3 examines the blending problem for composite laminate structures and reviews the guide based design methodology [6] for the fixed local loading scenario. Section 4 describes a global/local approach to the subproblem created by observing that a multi-panel wing structure is statically indeterminate. Changing the stacking sequence of one panel can change its stiffness properties, which affect the internal load distribution, possibly changing the in-plane load of every other panel in the structure. Section 5 supports the accuracy of the global/local approach through comparison to smaller problems with known optima discovered through exhaustive search. A parallel genetic algorithm is proposed to incorporate the global/local subproblem evaluation in a global optimization process and demonstrates the effectiveness of the method by finding several designs better than the best previously discovered in the literature. Section 6 presents a short lessons learned section about the approach and the high computational cost associated with it.

2: Genetic Algorithms

A genetic algorithm (GA) is a nondeterministic directed search algorithm using ideas based on natural selection to guide the exploration of the search space toward a global optimum. Common elements that occur in most genetic algorithms are those of population initialization, parent selection, crossover, mutation, and the selection of successive generations. Each element of the algorithm has many variations, modified to suit the needs of the problem at hand, including but

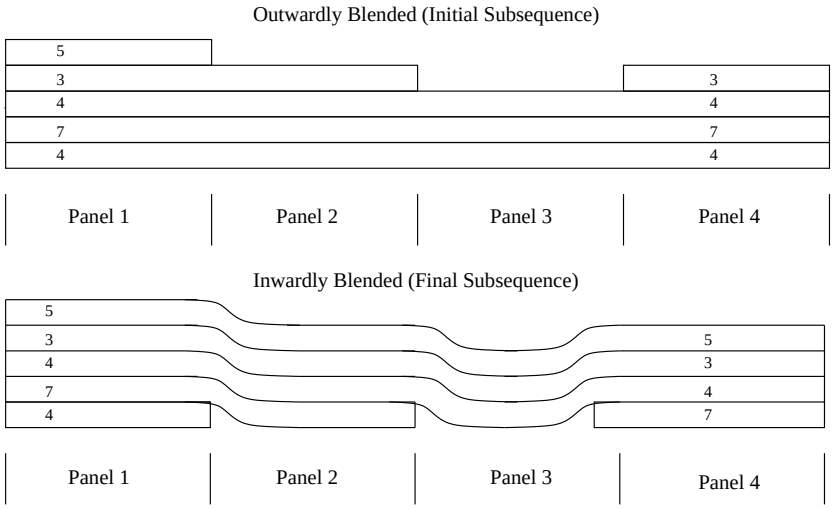

Figure 1. Outwardly/inwardly blended distinction example.

not limited to attempts to mimic natural genetics in every phase. The original work on genetic algorithms is attributed to Holland [7] in 1975, with application work following soon after in static function optimization by DeJong [8]. Goldberg [9] popularized the idea with his book in 1989, and is cited extensively in the literature defining genetic algorithms as search procedures based on the mechanics of natural selection and natural genetics. Bäck [10] is an excellent recent reference. Much of the work today using genetic algorithms can still fit this definition, though the concepts of natural selection and genetics are expanded to encompass some unnatural elements beyond the pristine translations from biology. In practice, genetic operators are tailored to specific problems in ways that have no analog in nature [3][11][12].

3: The Blending Problem

The blending problem here is defined as a laminate stacking sequence optimization on multiple interconnected panels with given local loading constraints. Each laminate is assumed to be balanced, symmetric, and constructed with a finite number of possible fixed orientations for each ply. The goal is to minimize the weight of the structure, determining the material type and orientation of each ply layer of every panel such that the loading constraints are satisfied locally and the panels form a blended global design. A blending measure for this work is defined in Section 3.2, but in general can be any measure of the manufacturability of a global design. Maintaining the continuity of material type and orientation across panel design boundaries, in part or in whole, is the primary concern in determining the blendedness of a global design. Ply orientation mismatches across panel design boundaries can cause manufacturing costs to rise in addition to the structural integrity issues associated with continuity breaks in the material.

3.1: Laminate Encoding

Each laminate must be encoded for use in the genetic algorithm. Following the coding scheme used by McMahon [13], integer values from zero to seven represent the orientation of each ply. The positive integers map to orientation angles 0, 15, 30, 45, 60, 75, and 90 degrees from one to seven, respectively, with the zero encoding representing an empty ply. Successive occurrences of a gene map to the orientation with alternating plus and minus signs, starting from the center of the (symmetric) laminate. For example, the first occurrence of a 2 encoding maps to $15°$, the second 2 maps to $-15°$, and so on. This is the meaning of balanced. Symmetric means that the

stacking sequence is symmetric with respect to the mid-plane of the laminate, and therefore only half of the orientations actually need to be encoded. For example, the stacking sequence design $[-60\ 90\ 0\ \pm 45\ 60\ 0_3]_s$, where s refers to the center of symmetry, is encoded as $(5, 7, 1, 4, 4, 5, 1, 1, 1)$. Similarly, the material type for each ply is represented by an integer, but here the material type is restricted to a single value meaning the material type for each ply is the same.

3.2: Guide Based Design

A simplified definition of blending is used in this work. Two adjacent panel designs are outwardly (inwardly) blended if one design is obtained by deleting a contiguous series of outermost (innermost) plies from the other. A global design is perfectly outwardly (inwardly) blended if and only if blending holds for every pair of adjacent panels starting from the innermost layer near the mid-plane (starting from the outermost layer and moving inward toward the mid-plane).

Using the simplified definition of blending, a perfectly outwardly (inwardly) blended global design can be generated by designing each local panel as an initial (final) sequence (a_1, ..., a_k is an *initial sequence*, and a_{n-k+1}, ..., a_n is a *final* sequence, of the sequence a_1, ..., a_k, ..., a_n) of a single larger guide design. Figure 1 displays the distinction between outward and inward blending rules. A GA can be created to operate on and generate a population of individuals that guide the global design process so that only globally blended designs are in the problem domain. During the analysis phase of the GA a guide design (a single individual from the population) is evaluated to determine the optimal initial sequence for each local panel satisfying loading constraints and minimizing weight. This is accomplished by stripping ply layers from the guide design, starting from the outermost (for outward blending) layers, one layer at a time and analyzing the resulting designs according to the constraints for each local panel. (More efficient variations of this are possible, e.g., starting with the initial sequence length equal to that of the last optimal design analyzed for this panel, and then adding/subtracting layers from this length.) The guide designs in the population acquire fitness values through the combination of optimal local designs generated by the guide itself so that each guide design has a single best series of initial sequences corresponding to each panel that can be used to construct a usable fitness function value. An optimal initial sequence is discovered for each local panel and the fitness of the global design becomes the sum of the individual fitness values for every optimized local panel. This modified analysis method provides the means to assign fitness values to guide designs themselves and allows a GA to operate on populations of guide designs to drive a global optimization process to an optimal blended design. An illustrative fictitious example of guide based design is presented in Figure 2.

4: Guide Design Subproblem

Guide based design is important because it reduces the global search space by examining only those designs that are completely blended. The reason that it works well in the above context is that it is computationally cheap to discover the optimal design corresponding to any particular guide design. Since the loads on all panels are fixed, a one dimensional optimization performed on each panel independently from the other panels is sufficient to discover the lightest possible design that satisfies the loading constraints. For fixed loading, this design is unique and can be easily assigned a fitness by the controlling GA. Guide based design can be summarized as a methodology to construct blended designs from a given thick laminate, wrapping this process into a global optimization GA to search for the best possible guide design laminate in the space of possible designs.

The problem with the above method is that loads are not fixed locally. A fixed load at the tip of the wing will distribute that load to each panel according to the stiffness and strength properties

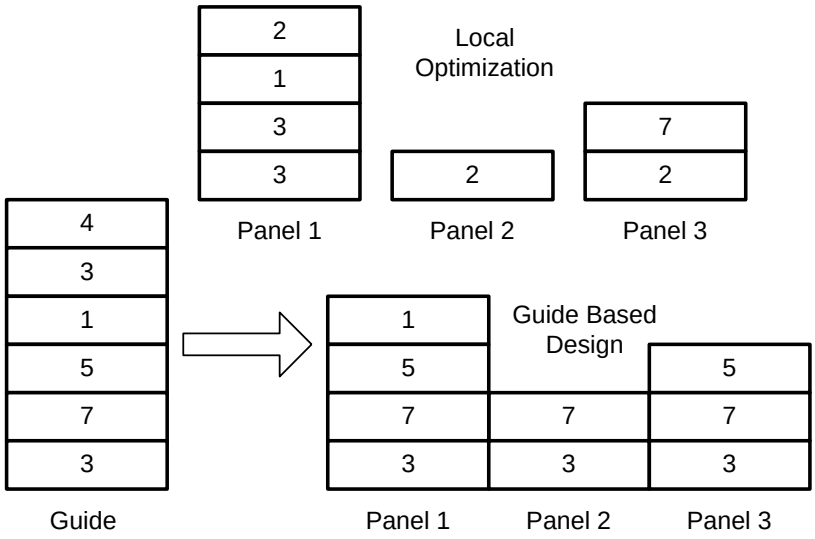

Figure 2. Guide based optimization example. The figure illustrates how isolated optimization creates designs that are locally optimal but are not easily manufacturable due to the ply orientation mismatches at through-the-thickness layers of the laminate. A side view of three sample panels, listing ply orientation encodings for each layer, is presented for both local optimization and a possible guide based design. Isolated local optimization generally produces light weight designs as less ply layers are required for Panels 2 and 3. The guide based design is optimized and constructed from the 6-ply thick guide design given on the left. The optimal initial sequence for the given guide produces a global design that is completely (outwardly) blended and therefore manufacturable. By operating on populations of guide designs a GA can discover an optimal blended global design minimizing weight and maximizing constraint satisfaction margins.

of each panel in the entire structure. For a given guide laminate, the fitness value (to be maximized) for the guide is defined as

$$-(W)(P+1),$$

where W is the total weight of the structure and P is the total penalty applied to the structure for being unbalanced. As stated in Section 3.1, the encoding for each laminate is a string of integers where successive occurrences of a gene map to the corresponding orientation with alternating plus and minus signs. A balanced stacking sequence contains an even number of each individual gene such that every occurrence of a positive orientation angle has a complimentary negative orientation somewhere in the laminate. The value of P is the sum of the number of genes occurring an odd number of times in the stacking sequence for a panel summed across all panels in the structure. If, however, any of the panels in the structure fail under their local loading requirements an additional multiplicative penalty is applied to the fitness calculation making failed designs highly undesirable. The precise fitness value for a guide design is defined as the minimum weight design such that all local panel loading requirements are satisfied and in cases where multiple configurations exist at the minimum weight for a given guide, a design minimizing the value of P from among the equal configurations is selected to represent the fitness. Note that this definition of fitness for a guide may not represent the best possible fitness value attainable from a given guide since it favors weight

before considering fitness value. This choice was made with the intention of steering the GA search toward designs using fewer layers even if they are not yet balanced.

The determination of the fitness of a guide is itself a complex optimization problem where the discovery of the optimal solution, as described above, may not be computationally feasible. The following sections describe methods used in this work to approximate the fitness value of an arbitrary guide.

4.1: Global/Local Iteration

The first proposed modification of the guide based design algorithm is to iterate between an exact global analysis and local level panel optimizations for specific loading conditions until convergence — a *global/local iteration*. The evaluation of a single guide begins with the global analysis of a completely thick design: each panel in the structure is assumed to use every possible layer from the given guide. A completely thick design can be represented as a pair with a particular guide encoding and a string of zeroes corresponding to the number of ply layers removed from each panel in the structure:

$$[GuideEncoding]_s; [0, 0, ..., 0].$$

Every configuration of values for ply layers removed, given in the second string above, represents a possible panel design configuration for the entire structure with corresponding local loadings on each panel. Since the determination of these local loads requires an expensive global finite element analysis, it is desirable to do as few global analyses as possible. The global analysis of the completely thick design yields local loading information for each panel in the structure used as input to the local level analyses to determine the failure status of each panel. Though the local loads given from the global analysis are only valid assuming the stacking sequences for each panel in the structure remain unchanged, these loads can allow for a local level linear optimization to guess a new design point to examine with a global analysis.

The local level optimizations take a single panel with the given loads returned from the most recent global optimization and determine the minimum length initial subsequence from the guide that can satisfy the given loading requirements. This operation can be performed independently on each panel and the local optimization results in a tuple of nonnegative integers corresponding to the number of ply layers removed from the guide for each panel to form the minimum weight structure possible under the given loads. For a given guide, this tuple fully describes an instance in the design space of all possible global designs derivable from the guide. This tuple becomes the best guess at which the next global analysis should be performed. The global analysis is required because by removing ply layers from the panels the stiffness properties change and the load distribution for the entire structure can change. The global analysis will yield new loading parameters for each panel in the structure and new linear local searches can be performed to determine the lightest possible structure based on the new loads. The process iterates between a global level analysis and the local level panel optimization under the following rules:

1) After a global analysis, the local loadings on all panels are exact until a change is made in the stacking sequence of a panel. If, under the given loads, a panel fails at the local level, then that panel must be made thicker and it is marked with a Boolean variable dissallowing the removal of ply layers during future iterations. Continue iterating between the global level analysis and independent local optimizations.

2) If, after the global analysis, local optimization returns a tuple that satisfies local loading constraints for each panel and does not try to reduce the thickness of any remaining panels unmarked from Rule 1 then convergence is declared and the iteration terminates. Otherwise, continue iterating between the global level analysis and independent local optimizations.

The above finite process, summarized in Figure 3, produces a unique design for each guide obtainable in a very quick iterative process (generally less than five iterations). There is no reason to believe that this design is optimal, given the criteria for determining the fitness of a guide above, but assume (at least for now) that the unique design found through this global/local search is the source of a directed acyclic subgraph containing the optimal solution (optimal design derived from the given guide). Each guide design can be encoded as a string. A corresponding design for the structure is encoded as an array of integers representing the number of layers removed from the guide for each panel in the structure. This creates an acyclic directed graph structure (with a source node) of all possible designs (encodings) derivable from a given guide by manipulating the corresponding array of panel descriptors. The source for a four panel structure would be the fully thick design designated by the list (0,0,0,0), where no ply layers are removed from the guide for any panel. The position of a number in the list corresponds to the design of a specific panel. The children of a specific node in the digraph correspond to all possible designs obtainable by removing a single ply layer from each panel in turn. For this example, the four children of the source node are the designs (1,0,0,0) (0,1,0,0) (0,0,1,0) and (0,0,0,1). An exhaustive search of the directed acyclic subgraph using the node corresponding to the unique design obtained through the global/local iterative approach as the source, stopping at nodes that correspond to infeasible designs, will discover the optimal solution under the following two assumptions:

1) the unique design discovered through this global/local iteration approach corresponds to the source node of a directed acyclic subgraph containing the (node describing the) optimal solution;

2) a directed path exists from the source node to any node describing a feasible design (under loading calculated by a global analysis) containing only nodes also describing feasible designs.

While no proof of 1) and 2) is known, no counterexample to these two assumptions has been found yet. Lacking theory concerning 1) and 2), this work is focused on finding the node (describing the optimal solution) in the directed subgraph using the converged design from global/local iteration as the source node. Since there is no proof of the above assumptions, the solution discovered through this process may not represent the exact fitness of the guide as given by the definition in Section 4, but can still be used as an approximation to the fitness and consequently in an outer optimization loop.

4.2: Middle Tier Addition

For problems with few panels, the combination of the global/local iteration (top tier) and an exhaustive search of the directed subgraph (bottom tier) resulting from the converged design was computationally feasible. However, for larger structures it was quickly realized that the directed subgraph could contain hundreds of thousands of nodes for some guide configurations and the need for a middle tier search method was highlighted. The basic idea of the global/local iteration is to get a design that is close enough to the optimal solution that the latter can be found in a reasonable number of analyses of the exhaustive digraph search. The problem is that the global/local iteration, since it tries to optimize all panels in parallel at the local level, can converge at a node too far from the node corresponding to the fitness value of the guide. Consider, for example, a case where a structure with four panels is at the node

$$[12, 13, 14, 14]$$

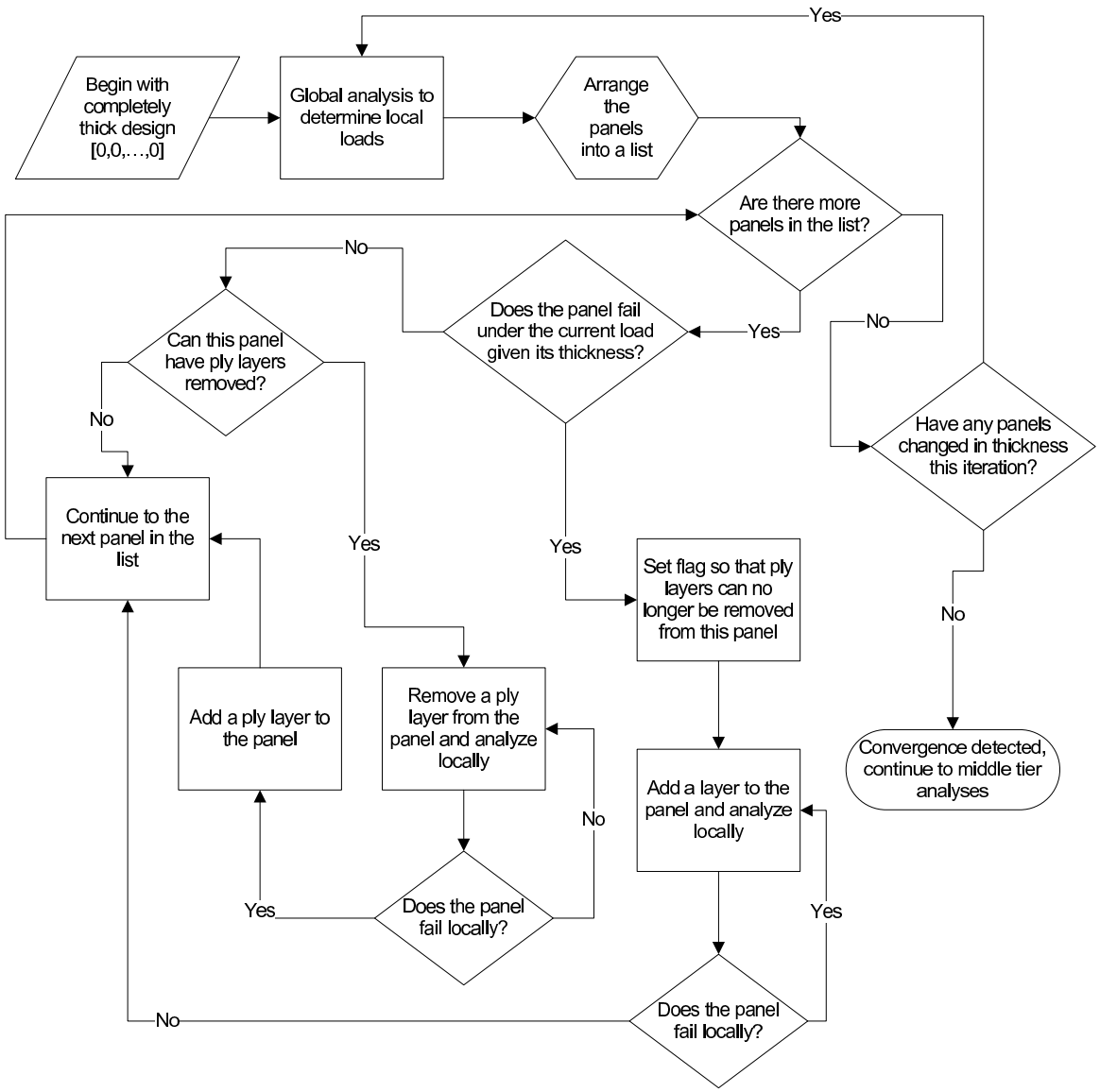

Figure 3. Top tier flow chart.

in a particular global/local search iteration. When the node is analyzed with the global analysis, loads are determined for each panel corresponding to that particular node. It is possible that for those particular loads each panel, through local level analysis, decides that it can remove an additional layer without failing. The new node for examination then becomes

$$[13, 14, 15, 15].$$

When this node is evaluated with the global analysis, loads are again determined for each panel corresponding to this new configuration. If, for example, all four panels now fail at the local level, then the next node to be considered would again be [12, 13, 14, 14], except this time all panels are marked with a failure flag and can not have further ply layers removed. If the optimal design happens to be [13, 14, 14, 15], three levels below the source node, then the directed subgraph search would have to go three levels deep to find the fitness value of the guide. For a problem

with only four panels this is not really an issue, but, for the eighteen panel problem compared later here, the same situation could cause the source node to be up to 17 levels above the optimal node corresponding to the fitness of the particular guide in question. This would create a directed subgraph that would be far too large to exhaustively search.

The first middle tier modification is to create a computational budget for each guide evaluation. Any subdigraph search that takes more computational time than the allotted amount will need to be dealt with in a different way. This issue is combatted in three different ways, with the first being a middle tier greedy algorithm intended to force the source node further down in the guide graph structure. Instead of starting the exhaustive subdigraph search immediately, the source node is evaluated with its given loads and the constraint margins for all panels are sorted into a list. The greedy approach then tries to remove a ply layer from the panel with the greatest constraint margin. The resulting design is analyzed globally to create new local loading values and then examined for failure. If the design is valid then the new design has its local constraint margins sorted and the process starts again, but if it fails in some way, then the panel with the next highest constraint margin has a ply layer removed and that design is examined. This process continues until an iteration where the entire sorted list of panels has a single ply layer removed and all of them fail in some way. This design corresponds to the rock bottom of a greedy approach since no more layers can be removed from the design. An exhaustive search starting from this node would be pointless, but in order to find the most balanced node possible the computational budget allows for an exhaustive search of about four levels deep. This is accomplished by recording the last four moves of the greedy algorithm and backtracking up the digraph to that node, or to the node corresponding to the design resulting from the top tier algorithm, whichever is closer to the final greedy middle tier node. This node becomes the source node for the exhaustive search (bottom tier algorithm). Figure 4 summarizes the middle tier algorithm just described.

The greedy middle tier modification is sufficient for making most guide designs fall within the computational budget, but some outliers still produce large subdigraphs. Two additional rules were introduced to save computational time. If a guide design, after global/local iteration and the middle tier greedy search, still falls 10 levels or more above the best design found so far then it is highly unlikely that this guide is going to produce a solution that is even competitive for the search. These designs are not even passed to the subdigraph search routine but rather return a fitness value corresponding to that given by the greedy approach and no more compute time is wasted. The second rule used is just a computational time kill switch placed in the subdigraph search routine. If a list of nodes to examine at a particular level of the subdigraph grows beyond a fixed length, 3600 for the eighteen panel example given later or about an hour of compute time, then the search is aborted. The best fitness value discovered at that point in the search is reported as the fitness for the guide, and this guide is also reported in the output file as being relatively close to the best known solution but producing a subdigraph that was too large to search in the context of the larger optimization run. Most guide designs can be fully evaluated in less than two minutes (for the problems considered here) so cutting searches off that take more than an hour seems reasonable.

5: Results

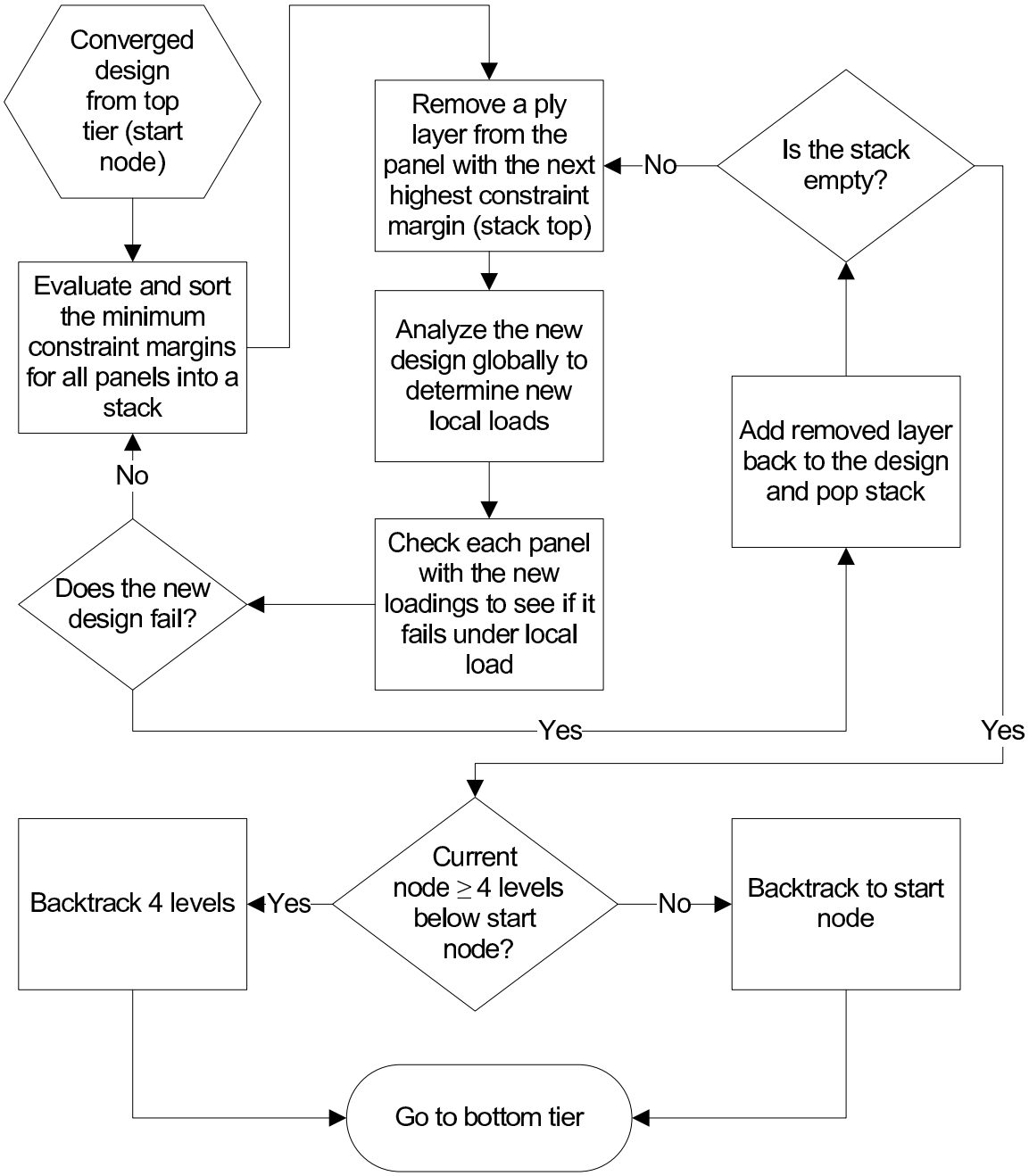

Figure 4. Middle tier flow chart.

5.1: Toy Problem

For the initial work, a simple wing structure model is used. The wing model is an unswept, untapered, rectangular wing box with dimensions 354.33 cm × 224.03 cm × 38.1 cm (139.5 in × 88.2 in × 15 in) (Figure 5) [14]. The spars and the ribs divide the top and bottom skin of the wing box into four panels of equal size. The top and bottom skin panels are modeled using membrane elements. Only the top skin panels are being optimized, all other sections are fixed to the design ($[\pm 45_5 45]_s$). The upper panel arrangement is shown in Figure 6. The root of the wing is fixed and load is applied at the free end. Global finite element analysis is performed using the commercial finite element package GENESIS, developed by Vanderplaats Research & Development, Inc. The

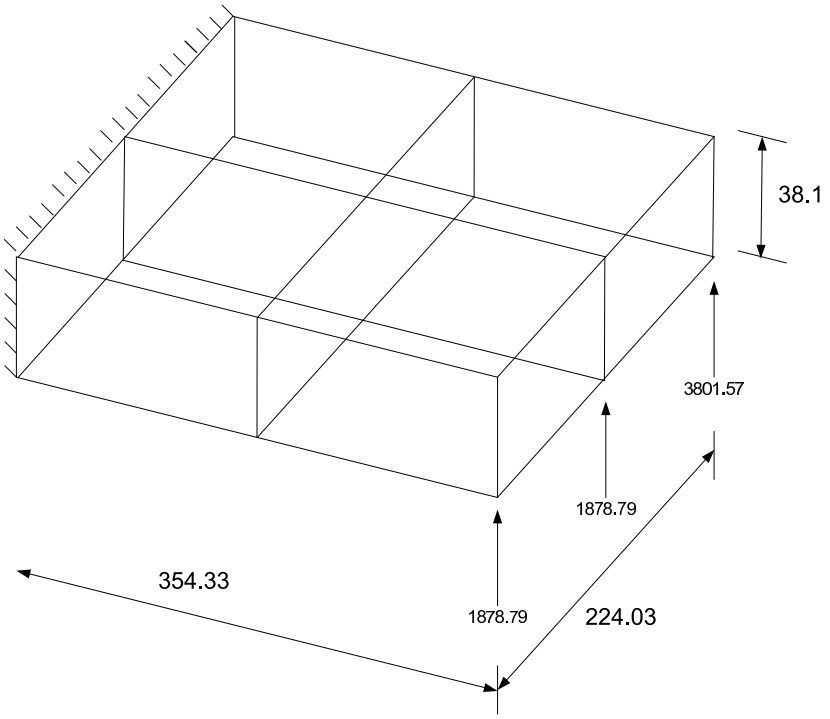

Figure 5. Wing structure model.

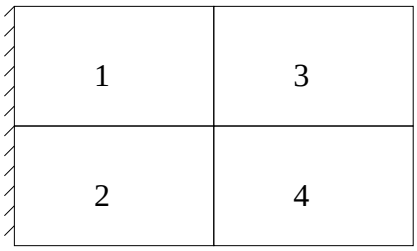

Figure 6. Panel layout for top and bottom skin panels.

upward lift force acting on the wing tip is modeled by three concentrated loads of magnitudes 3801.57 *nt* (at the leading edge wing tip), 1878.79 *nt*, and 1878.79 *nt* (854.67 *lbf*, 422.39 *lbf*, and 422.39 *lbf*).

Two guide designs were selected and fixed for the top four panels in the wing and examined extensively. All possible structures derivable from the guides consisting of 160,000 design points each (nodes in the digraph) were explored to determine which node should represent the fitness of the guide. In both cases the global/local method and subdigraph search described in Section 4 discovered the optimal solution for each subproblem in fewer than 27 global analyses.

A parallel GA, using the local/global method to evaluate guide designs, is constructed where the determination of fitness for a member of the population is assigned to a worker node during the analysis phase of each generation while the GA itself is controlled by a single master process. A static load distribution scheme is used in which each worker node receives an equal number of population members to analyze each generation. This is the same work distribution scheme used in the initial guide based design work by Adams et al. [6], but in this case the amount of computation required to determine the fitness of a single member of the population is not static. The variability in analysis times between individuals creates some load imbalances that will need to be corrected dynamically.

All testing for this problem was performed on a 200 node (400 processor) 1.4 GHz dual Opteron cluster using an MPI (message passing interface) based Fortran 90 code. However, the small problem size and the fact that the determination of fitness of an individual is viewed as an atomic operation prevented the use of more than 51 processors at a time. The average run time for a GA using a population size of 50 running for 200 generations was about 12 hours (612 node hours). Initial results are promising. As a point of comparison, the same problem above was examined using the algorithm by Seresta et al. [14]. For a run of 200 generations of the GA, Seresta's algorithm discovered in 50 generations a solution that it could not improve after an additional 150 generations. The guide for this design is given as the encoding

$$(4, 7, 1, 4, 4, 7, 4, 4, 4, 7, 1, 1, 4, 4, 1, 1, 7, 4, 1, 4),$$

using the final ten layers of the guide for Panel 1, 11 layers for Panel 2, six layers for Panel 3, and eight layers for Panel 4, yielding a combined mass of 12.371 *kg* (27.274 *lbs*). The greedy global/local with bounded implicit enumeration (GGLBIE) algorithm presented here discovered a different solution of equal mass in one to four generations of the GA and a lighter design was discovered after 100 iterations. This lighter design is given as the guide encoding

$$(7, 4, 7, 4, 4, 7, 7, 4, 1, 7, 4, 4, 1, 1, 1, 4, 1, 4, 1, 7),$$

using the final ten layers of the guide for Panel 1, 11 layers for Panel 2, six layers for Panel 3, and seven layers for Panel 4, yielding a combined mass of 12.018 *kg* (26.494 *lbs*).

5.2: Load balancing mechanism

Since the time to analyze a single guide can take between two minutes and multiple days, a load balancing scheme had to be introduced. Part of the load balancing scheme is reflected in the computational budget allocated to each analysis. Any analysis that expands to more than 3600 nodes to analyze during the subdigraph search is reported, with the nodes beyond the first 3600 being discarded. The initial distribution scheme was to use 400 processors and a population size of 400. Each processor was given a single guide to analyze each iteration, but for most iterations this resulted in a single processor computing for about an hour and the other processors remaining idle, as they finished the work very quickly. The efficiency suffered in the determination of fitness yielding about 0.2 guides per second on average. Using the same 400 processors and a population size of 400, a cut off point was introduced for all designs that were ten or more levels above the best known design after the middle tier greedy algorithm. The cut off point is just a mechanism for recognizing searches that most likely will not be competitive with the best known solution for the problem so far and even if they were to be competitive, their subdigraph searches would be too large to realize their potential. This process was accomplished by passing the information about the best known design found so far along with each unit of work and was updated throughout the optimization run even within a single generation. This increased the average number of guides analyzed to 0.8 guides per second. The major load balancing change though was in an increased population size and a modified work distribution scheme for the master-slave algorithm. In previous work [6] the master process would distribute work in equal sized chunks when the population size was larger than the number of worker processors available. This scheme was modified so that each worker processor would request a single guide to evaluate at a time and request new work when it finished the work it was given. At the same time the population size was increased by a factor of 50 to 20,000 individuals per generation. Using this basic load balancing scheme and a higher population allowed for the guides falling into the category of outliers to be analyzed for the entire duration of a generation while other workers could process many more routine guide evaluations sequentially. This method allowed the average number of guides analyzed per second to reach 9.5 and was sufficient for the following runs.

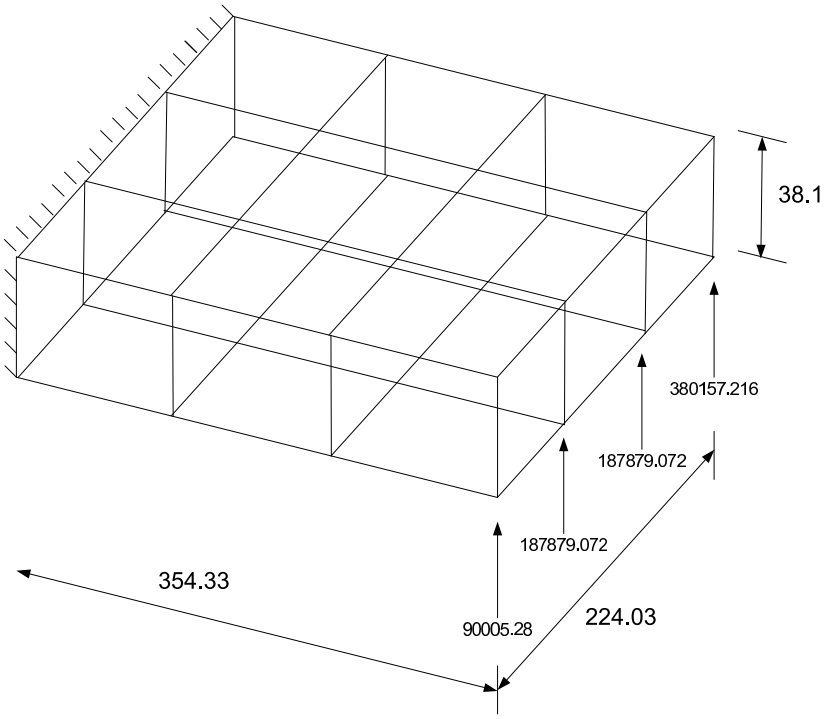

Figure 7. 18 panel wing structure model.

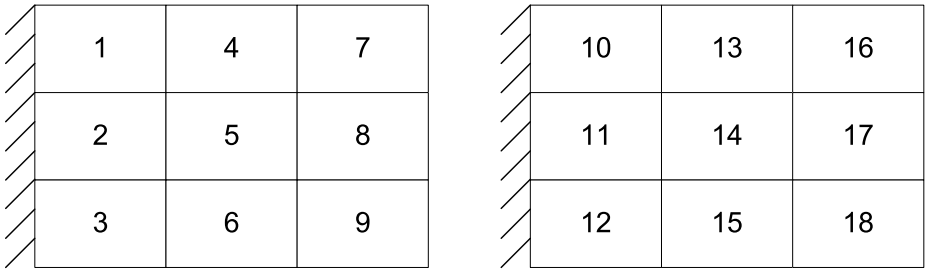

Figure 8. Panel layout for top (left) and bottom (right) skin panels for the 18 panel problem.

5.3: Comparison with other published work

A previous section displayed a success story for a toy problem, but a larger test problem is required to compare the GGLBIE algorithm with other methods published in the literature. This section describes the results and difficulties in dealing with an eighteen panel problem originally studied by Liu and Haftka [5] and again by Seresta et al. [14]. It should be stated up front that this method requires significantly more computational power than the above mentioned published works, but is a more effective search mechanism. The trade off for improved design quality in exchange for additional computational time can be weighed by the developer to evaluate the cost/benefit ratio for this method.

All testing for this larger problem was performed at Virginia Polytechnic Institute and State University on System X, a 1100 node (2200 processor) 2.3 GHz PowerPC cluster using an MPI based Fortran 90 code. A simple wing structure model is again used. The wing model is an unswept, untapered, rectangular wing box with dimensions 354.33 cm $\times$ 224.03 cm $\times$ 38.1 cm (139.5 in $\times$ 88.2 in $\times$ 15 in) (Figure 7). The spars and the ribs divide the top and bottom skin

Table 1. Stacking sequence of top panels.

3		90	1	90	90	0	0	90	90	0	90	90	0	90	90	0	90	0	90	90
	0	0	0	−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
2						0	0	90	90	0	90	90	0	90	90	0	90	0	90	90
	0	0	0	−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
1									90	0	90	90	0	90	90	0	90	0	90	90
	0	0	0	−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
6																0	90	0	90	90
	0	0	0	−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
5																				90
	0	0	0	−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
4																				
				−45	45	0	0	90	0	0	90	0	0	0	90	0	90	0	90	α
9																				
										0	90	0	0	0	90	0	90	0	90	α
8																				
														0	90	0	90	0	90	α
7																				
																				α
α									−45	45	−45	90	45	−45	45	−45	45	−45	45	−45
	45	−45	45	−45	45	−45	90	0	45	0	90	−45	0	0	0	90	0	0	45	0

of the wing box into nine panels of equal size. The top and bottom skin panels are modeled using membrane elements. The root of the wing is fixed and load is applied at the free end. The upward lift force acting on the wing tip is modeled by four concentrated loads of magnitudes 380157.216 *nt*, 187879.072 *nt*, 187879.072 *nt*, and 90005.28 *nt* (85467 *lbf*, 42239 *lbf*, 42239 *lbf*, and 20235 *lbf*) (Figure 7). The inwardly blended design reported in Table 1 and 2 was found after 94 iterations of the GA using a population size of 20000 and consumed about 140,000 node hours of compute time on 400 processors. The tables illustrate that all panels share the ply layers α or β and should be read such that the entry

$4 \longleftarrow 3$
$2 \longleftarrow 1$

corresponds to the stacking sequence $[1 \rightarrow 2\ 3 \rightarrow 4]_s$ as described in Section 3.1. The best known solution for this problem was reported by Seresta [14] and had a mass of 255.91 *kg* (564.17 *lb*). The solution discovered by the GGLBIE algorithm had a mass of 252.73*kg* (557.16 *lb*) using the inwardly blended approach, which the careful reader will determine is called outer blending in [14]. For reference, Table 3 gives the number of ply layers used for each panel and the corresponding mass.

6: Conclusions

The first observation about genetic algorithms for blended composite laminate structure design is that some restriction of the design space (e.g., a guide based GA considers only perfectly globally blended designs) seems necessary to make significant progress towards optimality. Simply because of the number of different fairly good blended designs that exist, many different heuristic approaches ([1], [2], [3], [4], [5], [6]) are able to find fairly good designs with moderate computational costs. None of these approaches, however, as the results here show, found a solution

Table 2. Stacking sequence of bottom panels.

16											−45	45	90	90	0	0	90	0	90	90	0	90	90	0	90	90	β
13													90	90	0	0	90	0	90	90	0	90	90	0	90	90	β
14																			90	90	0	90	90	0	90	90	β
17																			90	90	0	90	90	0	90	90	β
10																					0	90	90	0	90	90	β
11																					90	90	0	90	90	β	
18																									90	β	
12																										β	
15																										β	
β																										0	
	0	90	0	0	0	0	90	90	90	90	0	90	0	90	90	0	90	90	90	90							

Table 3. Panel thickness and mass.

Panel No.	Ply Layers	Mass kg (lb)
1	126	22.27 (49.09)
2	122	21.56 (47.53)
3	140	24.74 (54.55)
4	96	16.96 (37.40)
5	104	18.38 (40.52)
6	112	19.80 (43.64)
7	64	11.31 (24.94)
8	76	13.43 (29.61)
9	84	14.85 (32.73)
10	126	9.54 (21.04)
11	52	9.19 (20.26)
12	42	7.42 (16.36)
13	70	12.37 (27.27)
14	58	10.25 (22.60)
15	42	7.42 (16.36)
16	76	13.43 (29.61)
17	58	10.25 (22.60)
18	44	7.77 (17.14)

within 1% of optimal. The second observation is that, consistent with the prevailing wisdom in the evolutionary computation community, considerable effort to improve, refine, or "tune" individuals within a generation is cost effective. Here the three-tier deterministic algorithm, mixing greedy and implicit enumeration components, can be construed as a form of local improvement applied simultaneously to all the blended designs associated with a guide.

The reality of this work is that this type of combinatorial optimization problem is very difficult to solve efficiently. There may be ways to find a decent solution in a relatively short amount of time using other methods simply because there are a lot of relatively good designs in the design space, but to increase the quality of a solution toward the optimal for this type of problem requires a much larger expenditure of computing resources. This is typical of combinatorial optimization problems, and is true also for the traveling salesman problem, for example.

Acknowledgement:

The authors are indebted to the Terascale Computing Facility at Virginia Tech for a generous grant of computer time on System X.

References:

[1] D. B. Adams, L. T. Watson, Z. Gürdal, Optimization and blending of composite laminates using genetic algorithms with migration, *Mechanics of Advanced Materials and Structures*, vol. 10, pp. 183–203, 2003.

[2] G. Soremekun, Z. Gürdal, C. Kassapoglou, D. Toni, Stacking sequence blending of multiple composite laminates using genetic algorithms, *Composite Structures*, vol. 56, pp. 53–62, 2002.

[3] M. T. McMahon, L. T. Watson, A distributed genetic algorithm with migration for the design of composite laminate structures, *Parallel Algorithms and Applications*, vol. 14, pp. 329–362, 2000.

[4] B. P. Kristinsdottir, Z. B. Zabinsky, M. E. Tuttle, S. Neogi, Optimal design of large composite panels with varying loads, *Composite Structures*, vol. 51, pp. 93–102, 2001.

[5] B. Liu, R. T. Haftka, Composite wing structural design optimization with continuity constraints: Proc. 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Paper Number AIAA-2001-1205, Seattle, WA, 2001.

[6] D. B. Adams, L. T. Watson, Z. Gürdal, C. M. Anderson-Cook, Genetic algorithm optimization and blending of composite laminates by locally reducing laminate thickness, *Advances in Engineering Software*, vol. 35, pp. 35–43, 2004.

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, 1975.

[8] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems*, Ph.D. dissertation, Univ. of Michigan, Ann Arbor, MI, 1975.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, MA, 1989.

[10] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, 1996.

[11] N. Kogiso, L. T. Watson, Z. Gürdal, R. T. Haftka, S. Nagendra, Design of composite laminates by a genetic algorithm with memory, *Mechanics of Composite Materials and Structures*, vol. 1, pp. 95–117, 1994.

[12] R. Le Riche, *Optimization of composite structures by genetic algorithms*, Ph.D. dissertation, Department of Aerospace Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1994.

[13] M. T. McMahon, L. T. Watson, A distributed genetic algorithm with migration for the design of composite laminate structures. Virginia Polytechnic Institute and State University Technical Report TR-98-20, Dept. of Computer Science, Blacksburg, VA, 1998.

[14] O. Seresta, Z. Gürdal, D. B. Adams, L. T. Watson, Optimal design of composite wing structures with blended laminates: Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, Aug. 30–Sep. 1, 2004.