Evaluation of Injection Island GA Performance on Flywheel Design Optimization

David Eby, R. C. Averill
Department of Materials Science and Mechanics
William F. Punch III, Erik D. Goodman
Genetic Algorithms Research and Applications Group (GARAGe)
Michigan State University, East Lansing, MI 48824 USA
Phone (517)355-6453 Fax (517)432-0704
goodman@egr.msu.edu

ABSTRACT

This paper first describes optimal design of elastic flywheels using an Injection Island Genetic Algorithm (iiGA). An iiGA in combination with a finite element code is used to search for shape variations to optimize the Specific Energy Density of flywheels (SED is the rotational energy stored per unit mass). iiGA's seek solutions simultaneously at different levels of refinement of the problem representation (and correspondingly different definitions of the fitness function) in separate subpopulations (islands). Solutions are sought first at low levels of refinement with an axisymmetric plane stress finite element code for high-speed exploration of the coarse design space. Next, individuals are injected into populations with a higher level of resolution that uses an axisymmetric three-dimensional finite element model to "fine-tune" the flywheel designs. Solutions found for these various "coarse" fitness functions on various nodes are injected into nodes that evaluate the ultimate fitness to be optimized. Allowing subpopulations to explore different regions of the fitness space simultaneously allows relatively robust and efficient exploration in problems for which fitness evaluations are costly. First the paper treats a greatly simplified case – one for which all two million possible solutions were enumerated, yielding a known global optimum. Then the success and speed of many methods, including several variations of an iiGA, in finding this known global optimum are compared. The iiGA methods always found the global optimum, and the other methods never did. Hybridizing the iiGA with a local search operator and a Threshold Accepting (TA) search at the end of each generation provided the fastest solutions, without sacrificing robustness. Finally, a problem with a large design space is presented and results are compared for a hybrid iiGA to a parallel GA that uses a topological "ring" structure. The hybrid iiGA greatly outperforms the topological "ring" GA in terms of fitness and search efficiency for this given problem.

1.0 INTRODUCTION

New optimization problems arise every day -- for instance, what is the quickest path to work? Where and how congested is the road construction? Am I better off riding my bike? If so, what is the shortest path? Sometimes these problems are easily solved, but many engineering problems cannot be handled satisfactorily using traditional optimization methods. Engineering involves a wide class of problems and optimization techniques. Many engineering design approaches such as "make-it-and-break-it" are simply out-of-date, and have been replaced by computer simulations that exploit various mathematical methods such as the finite element method to avoid costly design iterations. However, even with high-speed supercomputers, this design process can still be hindersome, producing designs that evolve slowly over a long period of time. The next step in the engineering of systems is the automation of optimization through computer simulation. If the desired performance factors for the system can be appropriately captured, then optimization over them is simply engineering on a grander scale.

Optimization approaches include hill climbing, stochastic search, directed stochastic search and hybrid methods. Hill climbing or gradient-based methods are single-point search methods that have been applied successfully to many shape optimization problems [1-3], and are extensible via neighborhood sampling even to cases in which derivatives are not analytically given. However, these methods are severely restricted in their application due to the likelihood of quickly converging to local exrema [4]. Random search methods simply evaluate randomly sampled designs in the search space, and are therefore generally limited to problems that have small search spaces, if practical search times are required. A directed random search method, such as a Genetic Algorithm (GA), is a multiple-point, directed stochastic search method that can be an effective optimization approach to a broad class of problems. The use of GA's for optimal design requires that a large number of possible designs be analyzed, even though this number generally still represents only a miniscule fraction of the total design space. When each evaluation is computationally intensive, a traditional simple or parallel GA can thus be difficult to apply. Injection Island Genetic Algorithms (iiGA's) can help reduce the computational intensity associated with typical GA's by searching at various levels of resolution within the search space using multiple analyses that can vary in levels of complexity, accuracy and computational efficiency.

Structural optimization via GA's is the main area of interest for this paper [5-13]. Recently, GA's have been successfully applied in the optimization of laminated composite materials [14-18]. The authors of the current paper have used an iiGA in the design of laminated composite structures [14,15]; others use different GA approaches [16-18]. iiGA's have also been applied to engineering problems such as [19]. [20-24] deal with the application of GA's to shape optimization problems. [20, 21] used GA's to find optimal shapes based on various polynomials while [22] presents the concept of fictitious domains to generate new shapes and [23] reduced computational costs associated with generating meshes for finite element evaluations by a point heat sink approach. [24] modeled flywheels as a series of concentric rings (see Figure 1) using a simple GA measuring fitness with a plane stress finite difference model. Although [24] has already performed optimization of flywheels using a simple GA, this paper differs in many respects: [24] *seeded* the initial population with flywheels that varied linearly in thickness from the inner to outer radii, allowing genetic operators to find new shapes, while this paper allows for ring thickness to be *randomly* chosen in the initial population; [24] searched for shapes using a simple GA while this paper will present various optimization approaches such as Threshold Accepting (TA), GA's, iiGA's and hybrid techniques; [24] based fitness on a *single objective* in each run while *multiple fitness* definitions where used *concurrently* in each iiGA run for this paper; [24] measures fitness only with a *plane stress evaluation* while the current paper presents techniques that *concurrently* use *multiple evaluations* that vary in levels of complexity, accuracy and computational efficiency.

Combining a GA with the finite element method is by now a familiar approach in the optimization of structures, but using an iiGA with multiple evaluation tools and with different fitness functions is a new approach aimed at decreasing computational time while increasing the robustness of a typical GA. Typically, a useful approximation to the overall response of most structures can be captured with a computationally efficient, simplified model, but often, these simplified models cannot capture all complex structural behaviors. If the model does not accurately capture the physics of the problem, then the results of any optimization technique will be an *artifact* of the simplified analysis, dooming the solution(s) to be incorrect. This forces the designer to use a more refined model, which can be computationally demanding, sometimes leading to evaluation times too long to be practical for use in GA search. These obstacles are nearly always present in interesting structural optimization problems. This paper will show how an efficient, simplified axisymmetric plane stress finite element model, when used to evaluate fitness in an optimization problem, produces solutions that are *artifacts* of the simplified analysis. The paper will also show that an ordinary parallel GA using the refined axisymmetric finite element model requires excessively long search times, in comparison to an iiGA approach which employs both types of FEA representation.

An eventual goal of this effort is to develop tools for multi-criterion optimization of large-scale, 3-dimensional composite structures, using an iiGA that searches at various levels of resolution and model realism. It incorporates several simultaneous and interconnected searches, including some that are faster (but often less accurate). This approach is designed to spend less time evaluating poor designs with computationally intensive fitness functions (this is to be done with the efficient, less accurate evaluations) and to spend more time evaluating potentially good designs with the computationally intensive fitness evaluation.

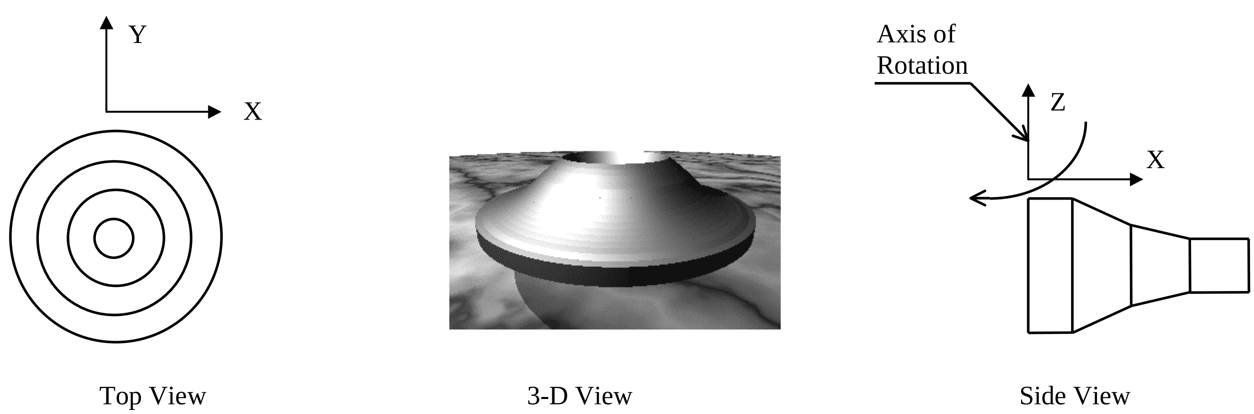


Figure1. Visual Display of Flywheel

For the flywheel problem treated here, the lowest level of the iiGA searches with a simple axisymmetric plane stress finite element model (with a "sub-fitness" function), which quickly finds "building blocks" to inject into a series of GA populations using several more refined, axisymmetric, three-dimensional finite element models. An optimal annular composite flywheel shape is sought using both this iiGA approach and, for comparison, using a "ring" topology parallel GA (PGA). The flywheel is modeled as a series of concentric rings (see Figure 1). The thickness within each ring varies linearly in the radial direction. A diverse set of material choices is provided for each ring. Figure 2 shows a typical planar finite element model used to represent a flywheel, in which symmetry about the transverse normal direction and about the axis of rotation is used to increase computational efficiency. The overall fitness function for the genetic algorithm GALOPPS [1] was the specific energy density (SED) of the flywheel, which is defined as:

$$SED = \frac{\frac{1}{2} I\omega^2}{mass} \tag{1.)}$$

where ω is the angular velocity of the flywheel ("sub-fitness" function), I is the mass moment of inertia defined by:

$$I = \int_V \rho \cdot r^2 dV \tag{2.)}$$

and ρ is the density of the material.

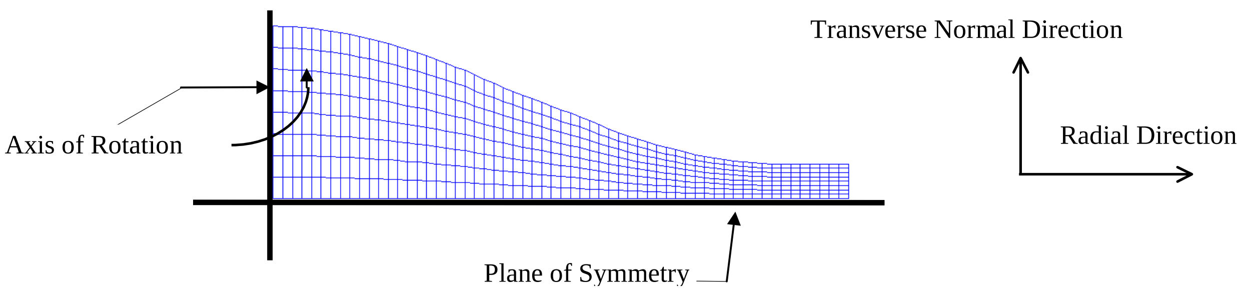


Figure 2. Typical Flywheel Model

1.1 SIMULATED ANNEALING AND THRESHOLD ACCEPTING

Simulated Annealing (SA) is a combinatorial optimization technique that is based on the statistical mechanics of annealing of solids [25]. To understand how such an approach can be used as an optimization tool, one must consider how to coerce a solid into a low energy state. Annealing is a process typically applied to solid materials to force the atomic structure of the material into a highly ordered state. Atomic structures that maintain a highly ordered state are also at a low energy state. In an annealing process, a material is heated to a temperature that allows many atomic arrangements, then cooled slowly, minimizing energy, while statistically allowing an occasional increase in atomic energy. When the material is extremely hot, the probability of an increase in atomic energy is very high. As the cooling continues, the probability of an increase in atomic energy decreases. Similarly, SA methods use analogous set of parameters that simulate controlled cooling effects found in the annealing of materials.

SA methods begin with an initial solution that is often generated randomly, and try to perturb the solution to improve it. If the perturbation improves the solution then it is accepted and the process of perturbing continues. In this manner, SA methods are like iterative methods that climb hills. As with hill climbing methods, this process of searching just for a better solution tends to force the process to a local optimum. However, SA methods are different in this respect: annealing occasionally allows perturbations that are harmful to the solution to be accepted. This allows SA methods to "climb out" of local optima to search for a global optimum. In real physical systems, jumps to a higher ("worse") state of energy actually do occur. Probability of these jumps is reflected in the current temperature. As the annealing process (cooling) continues, the probability that only better solutions will be accepted increases. At the beginning of the annealing process (associated with a high temperature), the chance that a worse solution is accepted is, while later in the annealing process (at a lower temperature) the chance that a worse solution is accepted is small. This probability of accepting worse solutions is based on a Boltzman distribution:

$$\Pr[Accept] = e^{-\frac{\Delta E}{T}} \qquad 3.)$$

By successively lowering the temperature T, the simulation of material coming into equilibrium at each newly reduced temperature can effectively simulate physical annealing.

Threshold Accepting (TA) is a simplified version of Simulated Annealing. The probability of accepting a worse solution is governed by the Boltzmann distribution for SA applications and the TA algorithm, but the TA algorithm is not dependent upon a specified temperature. Instead, the TA algorithm rate of cooling is based on a specified percentage of the current solution fitness. This percentage decreases over the set of generations. This causes the TA in earlier generations to have a higher probability of accepting a worse individual, while later generations in the optimization are less likely to accept a worse solution.

1.2 PARALLEL GENETIC ALGORITHMS

Two problems associated with GA's are their need for many fitness evaluations and their propensity to converge prematurely. An approach that ameliorates both of these problems is a parallel GA (PGA), which also produces a more realistic model of nature than a single large population. PGA's typically decrease processing time to a given solution quality, even when executed on a single processor, and better explore the search space. If they are executed using parallel processors, an additional speedup (in wall clock time) nearly linear with processor number may be achieved.

Unlike some specialized sequential GA's which may pay a nontrivial computational cost for maintaining a structured population (demes, etc.) based on similarity comparisons (niching techniques, etc.), PGA's maintain multiple, separate subpopulations which are allowed to evolve nearly independently. This allows each subpopulation to explore different parts of the search space, each maintaining its own high-fitness individuals and each controlling how mixing occurs with other subpopulations, if at all.

1.3 INJECTION ISLAND GA's

iiGA's represent an extension to the usual notion of parallel GA's, allowing each subpopulation to search at a different levels of resolution within a given space, or to search using representations or fitness functions which differ in some other way among subpopulations. This includes searching at low levels of resolution on some nodes (islands) and injecting their highest-performance individuals into islands of higher resolution for "fine-tuning". This injection occurs while all islands continue to search simultaneously, although it is also possible to stop or re-assign low-resolution islands once they have converged. The parallel GA environment in which the iiGA is run is based on the GALOPPS toolkit [26] developed by one of the authors. The software can be run on one or multiple workstations [26] (a single processor was used for all runs reported here). Figure 3a shows the iiGA used for this paper to perform multiple refinements in the geometric representation by increasing the number of rings in the flywheels. For composite analysis, material properties can vary from ring to ring of the flywheel. Islands with different levels of resolution evaluate fitness using either a simplified analysis that is computationally cheaper or a refined, computationally expensive analysis. Different GA parameters can be used for each population. The rates of crossover, mutation, and island interaction can all vary from island to island. For example, an island can exploit a simplified evaluation tool that is computationally cheap by increasing the island's population size. Also, islands using a computationally cheap evaluation function can be allowed to evaluate more generations before injecting their results into other islands.

Many engineering problems require satisfying multiple fitness criteria in some sort of weighted overall fitness function to find an optimal design, if not actually requiring multicriterion optimization. Each individual fitness measure may have its own optimal or suboptimal solutions. In an iiGA, it may be useful to use each individual criterion as the fitness function for some subpopulations, allowing them to seek "good" designs with respect to each individual criterion, as potential building blocks for the more difficult weighted fitness function, or as useful points for assessment of Pareto optimality. iiGA's take advantage of the low communication required to migrate individuals from island to island. Often, only the best individual in a population migrates to allow "good" ideas (building blocks) to be combined with other "good" ideas to find "better" ideas amongst islands using different "sub-fitness" functions. Finally, for weighted fitness evaluation, individuals may be injected into a set of nodes where the evaluation of an overall weighted fitness function is employed. This search method facilitates robust exploration of the search space for all aspects of the overall fitness. Of course, many variations on these injection island architectures can be custom tailored for specific problems.

iiGA's using islands of different resolutions have the following advantages over other PGA's:

(i) Building blocks of lower resolution can be directly found by search at that resolution. After receiving lower resolution solutions from its parent island(s), an island of higher resolution can "fine-tune" these solutions, but may also reject those inferior to better solution regions already located.

(ii) The search space in islands with lower resolution is proportionally smaller. This typically results in finding "fit" solutions more quickly, which are injected into higher resolution islands for refinement.

(iii) Islands connected in the hierarchy (islands with a parent-child relationship) share portions of the same search space, since the search space of the parent is typically contained in the search space of the child. Fast search at low resolution by the parent can potentially help the child find fitter individuals.

(iv) iiGA's embody a divide-and-conquer and partitioning strategy which has been successfully applied to many problems. In iiGA's, the search space is usually fundamentally divided into hierarchical levels with well-defined overlap (the search space of the parent is contained in the search space of the child).

(v) In iiGA's, nodes with smaller block size can find the solutions with higher resolution. Although Dynamic Parameter Encoding (DPE) [27] and ARGOT [28] also deal with the resolution problem, using a zoom or inverse zoom operator, they are different from iiGA's. First, they are working at the phenotype level and only for real-valued parameters. iiGA's typically divide the string into small blocks regardless of the meaning of each bit. Second, it is difficult to establish a well-founded, general trigger criterion for zoom or inverse zoom operators in PDE and ARGOT. Furthermore, the sampling error can fool them into prematurely converging on suboptimal regions. Unlike PDE and ARGOT, iiGA's search different resolution levels in parallel and may reduce the risk of zooming into the wrong target interval, although there remains, of course, a risk that search will prematurely converge on a suboptimal region.

2.0 FINITE ELEMENT MODELS OF FLYWHEELS

Two axisymmetric finite element models were developed to predict planar and three-dimensional stresses that occur in flywheels composed of orthotropic materials undergoing a constant angular velocity. Both finite element models were developed applying the principle of minimum potential energy. The finite element model that assumes a plane stress state is truly a one-dimensional finite element model, and is accurate when the gradient of the flywheel thickness is small. The finite element model that yields a three-dimensional stress state is truly a two-dimensional finite element model, and is accurate for all shapes. An automated mesh generator was written to allow for mesh refinement through the transverse normal and the radial directions. Therefore, the finite element code that predicts three-dimensional stresses can have various levels of refinement. A coarse mesh with a small number of degrees of freedom will be less accurate but more efficient than a refined mesh that contains more degrees of freedom. The mesh was also generated to minimize the time required to solve the set of linear equations created by the finite element code. By first assuming an initial angular velocity, the stresses and strains were calculated. Next, the initial angular velocity was scaled to the maximum failure angular velocity. The maximum stress failure criterion was used to predict the maximum failure angular velocity in the analysis of isotropic flywheels, while the maximum strain criterion was used for composite flywheels.

3.0 FINDING GLOBAL OPTIMA FOR A SIMPLIFIED FLYWHEEL

In order to explore how effective the iiGA search is in finding the global optimum for this sort of problem, and to compare the speed of finding it using iiGA's with various enhancements, a simplified flywheel problem was posed. A flywheel that contains 6 concentric rings (i.e., 7 heights) with 8 possible values for each height (see Figure 3b) created a design space of 8^7 or about 2 million possible designs. Using a coarse (962 DOF), three-dimensional finite element model, it was possible to calculate the SED of all of these designs, in about 50 hours on a SPARC Ultra processor. With the global optimum design known from exhaustive search, other search methods could be judged as to robustness and efficiency.

The TA algorithm alone began its search with a randomly initiated design. All hybrid algorithms that incorporated the TA algorithm were initiated with the best individual of the current generation, performing at most 10 TA operations, with the resulting solution always replacing the worst in the population. The local search method took the best individual of each generation and varied the thickness profile of whichever ring the FEA code found to fail first. The inner and outer thicknesses were increased and decreased independently, so a total of four evaluations occurred. When incorporating the local search method in any algorithm, the worst solution in the population was replaced only when a better solution was found by the local search. All multipoint search methods used the same total population size, 2,200 individuals. All GA runs used elitism (guaranteed survival of best individual) and one-point crossover. Typically, for larger, computationally expensive problems, each island would be located on a separate processor, but for this problem, only a single Sun Sparc Ultra workstation was used. All GA runs used a 1% mutation rate.

The motivation for the particular iiGA topology used here requires some explanation. The search space for the plane stress finite element model evaluation contains good building blocks for the iiGA. Also, the plane stress evaluation (0.001seconds per evaluation) is up to 1000 times faster than the three-dimensional evaluation of stress (for this analysis). To make the iiGA search less computationally intensive and more robust, the iiGA shown in Figure 3a was designed to exploit these facts. A full cycle in an iiGA consists of evaluating a specified number of generations (which varies from island to island) in each island. Genetic operations can also be varied from island to island. Islands 0 through 1 had a 75% rate of crossover, population size of 300, and completed 12 generations per cycle before migrating 3 individuals in accordance with Figure 3a. Islands 0 and 1 measured fitness with plane stress finite element code, basing fitness on the sub-fitness function (angular velocity alone). Islands 0 and 1 contained designs with 3 and 6 rings with 7 and 13 DOF, respectively. The choice of a high crossover rate was chosen to motivate those particular islands to discover new designs. A large population size and high number of generations per cycle was used due to the computational efficiency of the plane stress evaluation and to force the islands to converge quickly to potentially productive regions of the design space, presumably containing useful building blocks. Islands 2 and 3 had a crossover rate of 70%, population size of 200, and completed 6 generations per cycle before migrating 3 individuals, evaluating fitness with the three-dimensional axisymmetric finite element code basing fitness on SED (130 DOF). Islands 4 and 5 had a 65% crossover rate, population size of 200 and completed 4 generations before migrating individuals, measuring fitness with the three-dimensional axisymmetric finite element code basing fitness on SED (430 DOF). Islands 6 through 8 had a crossover rate of 60%, population size of 100, and received migrated individuals every 2 generations, measuring fitness with the three-dimensional axisymmetric finite element code basing fitness on SED (962 DOF). Islands 6 through 8 have a lower population size and number of generations per cycle to explore the space more slowly and to avoid a large number of costly evaluations. Islands 6 through 8 should fine tune potentially good designs (building blocks) received from the islands at a lower resolution. Figure 3a also displays a hybrid iiGA design that groups the islands according to the method by which they perform their specialized heuristic search (if any) at the end of each generation. Of course, many variations on these hybrid iiGA designs can be custom tailored for specific problems.

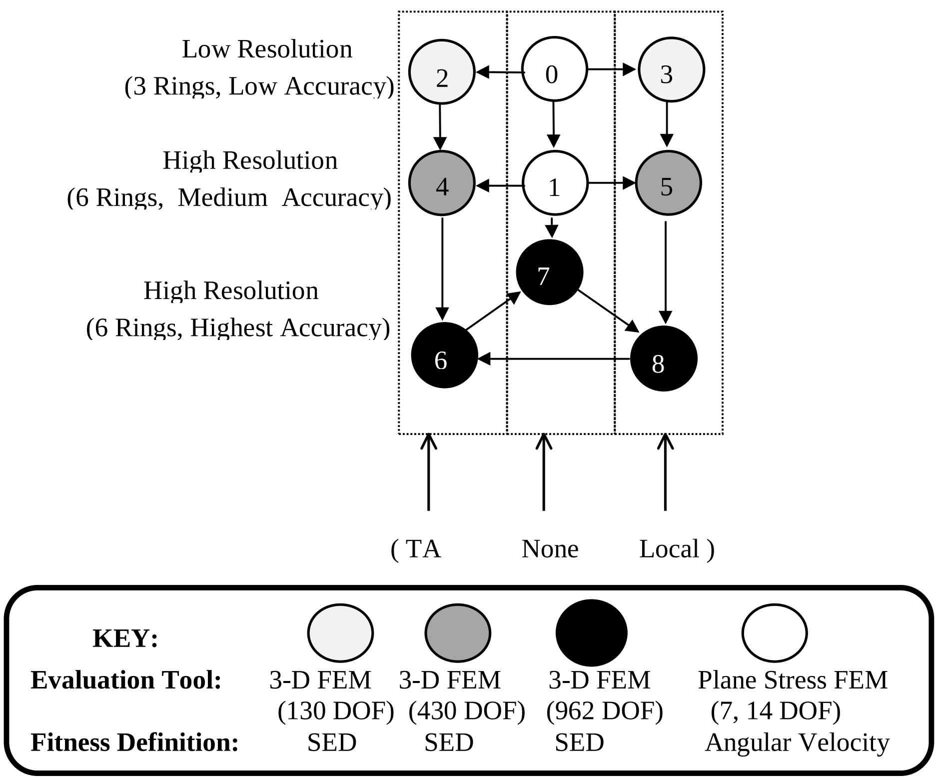


Figure 3a. Simplified iiGA Topology

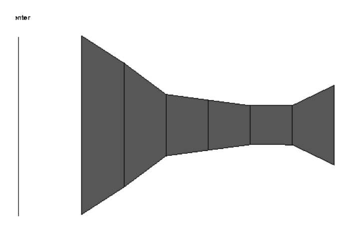


Figure 3b. Typical Coarse Flywheel Design (6 Rings).

Figure 3. Simplified Injection Island GA Topology with Coarse Flywheel Representation.

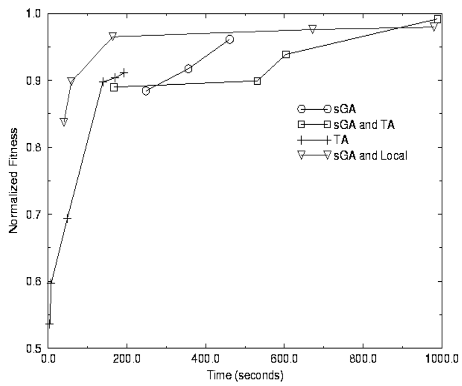
4.0 RESULTS OF GLOBAL OPTIMIZATION STUDY

Table 1 shows the results of the various methods. Each run lasted 6000 seconds on the same processor. In five runs of each method, the simple GA, with and without TA and local search heuristics, and the ring topology parallel GA, never found the global optimum. Figure 4 displays the fitness (reevaluated with the most accurate FEM evaluation) as a function of time of a typical run for a TA algorithm, simple GA and a simple GA that incorporated either a TA algorithm or a local search method. Elitism was used in all GA runs, so solutions are only plotted when better solutions are found (leading to the appearance of different run lengths).

Other hybrid iiGA topologies were tested that incorporated either Threshold Accepting or local search methods. Without the local search or TA heuristics, the iiGA took an average of 768 seconds to find the global optimum. The hybrid iiGA that also used local search found the global optimum in 715 seconds (average) while the iiGA that incorporated the TA found the global solution in 674 seconds (average). Figures 5 and 7 display the fitness as a function of time for the iiGA (same topology as Figure 3a) and hybrid iiGA (Figure 3a, TA/None/local), respectively. The iiGA alone found the global solution in 768 seconds (average), while the hybrid iiGA (Figure 3a, TA/None/Local) found the global optimum in 417 seconds (average). The hybrid iiGA that used the TA algorithm and local search method evaluated less than 5% of the entire search space, taking less than 0.5% of the time needed to enumerate the entire search space, measuring more than half of the evaluations with the plane stress finite element model to find the global optimum. Examination of Figure 4, shows that the local search and the TA help the simple GA find better solutions. Also, the TA alone quickly climbs to a suboptimal solution. Figure 5 shows the iiGA quickly finding "building blocks" at low levels of resolution that are injected into islands of higher resolution. Figure 6 displays the hybrid iiGA (Figure 3a, TA/none/Local) benefiting from the combination of TA and local search heuristics. Figures 4-6 only display the first 1000 seconds because no better solutions were ever found thereafter.

Table 1. Comparison of Optimization Approaches.

Optimization Technique	Average Time to Find Global Solution (5 Runs)
TA	Never Found
Simple GA	Never Found
Simple GA with Local Search	Never Found
Simple GA with TA	Never Found
Ring Topology GA	Never Found
iiGA	Always Found, 768 Seconds
Hybrid iiGA with Local Search	Always Found, 715 Seconds
Hybrid iiGA with TA	Always Found, 674 Seconds
Hybrid iiGA with Local Search and TA	Always Found, 417 Seconds



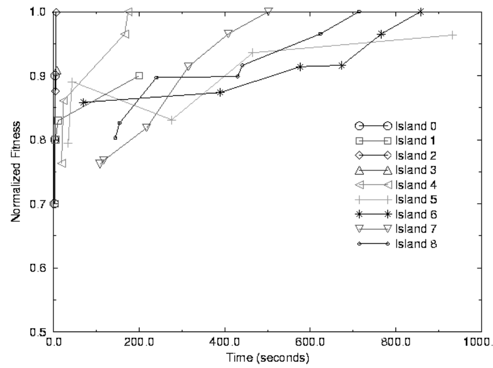


Figure 4. Fitness as a Function of Time on a Single Processor for a Typical Run of a Simple GA, Simple GA with TA, and Simple GA with Local Search Method.

Figure 5. Fitness as a Function of Time on a Single Processor for a Typical iiGA Run.

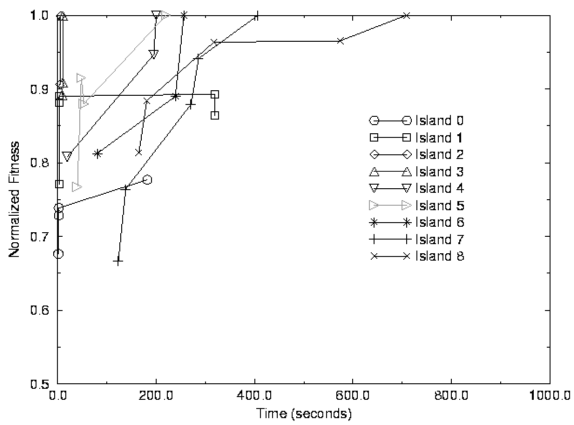


Figure 6. Fitness as a Function of Time on a Single Processor for a Typical Hybrid iiGA that Incorporated TA and Local Search Methods.

5.0 SEARCHING LARGER DESIGN SPACES USING iiGA's and PGA's

A larger search domain was created to compare the iiGA to a topological "ring" GA. A 24-ring flywheel with 1024 heights per thickness created a huge design space. Again, to make the GA search less computationally intensive and more robust, an iiGA as shown in Figure 7a was designed. Islands 0 through 2 evaluate fitness based on angular velocity with a simplified plane stress finite element model with varying geometric resolutions (3, 6 and 12 rings). Islands 0 through 2 have 7, 13 and 25 computational degrees of freedom, respectively. Islands 3 through 11 measure fitness based on SED using the three-dimensional axisymmetric finite element model. Islands 3 and 4 are low in geometric resolution (3 rings), but have 160 degrees of freedom. Islands 5 and 6 are medium in geometric resolution (6 rings), containing 558 degrees of freedom. Islands 7 and 8 are high in geometric resolution (12 rings), having 1,952 degrees of freedom. Islands 9 through 11 are the highest in geometric resolution (24 rings) with 9,982 degrees of freedom.

A full cycle consists of evaluating a specified number of generations (which varies from island to island) in the injection island topology. Islands 0 through 2 had a 75% rate of crossover, population size of 300, and completed 15 generations per cycle before migrating the island's best individual in accordance with Figure 7a. Islands 3 and 4 had a crossover rate of 70%, population size of 100, and completed 7 generations per cycle before migrating 3 individuals. Islands 5 and 6 had a 65% crossover rate, population size of 130 and completed 4 generations before migrating individuals. Islands 7 and 8 had a crossover rate of 65%, population size of 130 and migrated individuals after evaluating 3 generations. Islands 9 through 11 had a crossover rate of 60%, population size of 100 and received migrated individuals every 3 generations. Islands 0 through 2 can converge much faster to "good" building blocks when compared to the rest of the islands due to the simplification of the plane stress evaluation and the level of resolution. The iiGA topology design in Figure 7a uses this as an advantage because the topology injects building blocks from the simplified plane stress evaluation based on angular velocity into two isolated islands that evolve independently, searching separate spaces efficiently using the axisymmetric three-dimensional finite element model to evaluate SED.

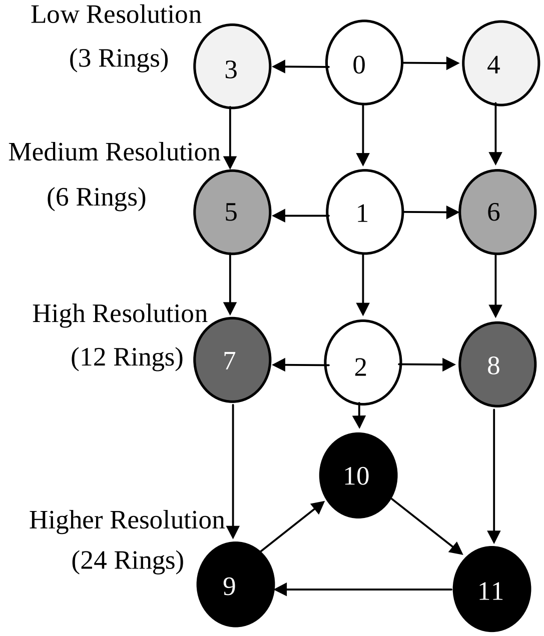


Figure 7a. Injection Island Topology

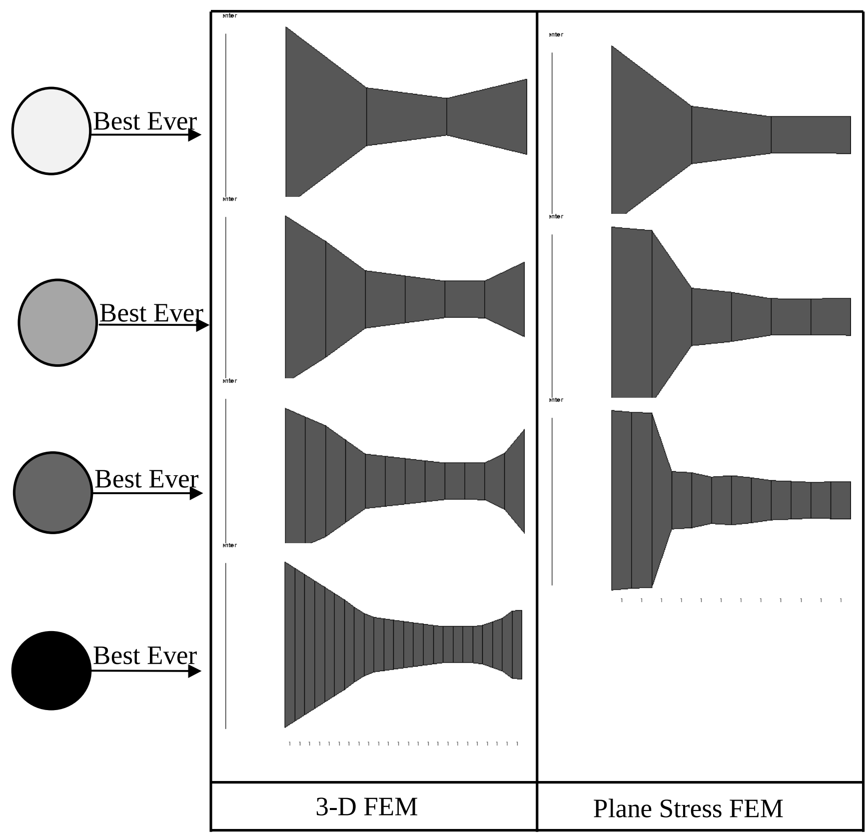


Figure 7b. Best Flywheel Found for Each Level of Resolution

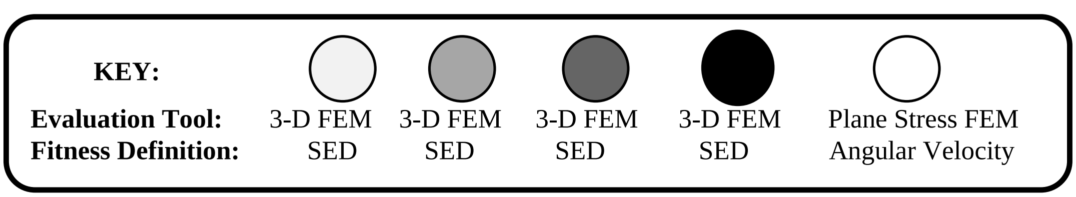
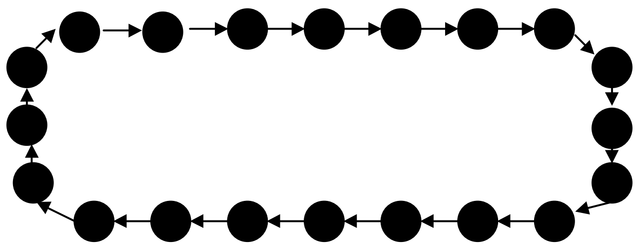
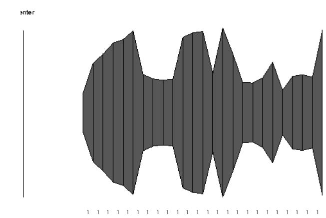


Figure 7. Injection Island GA Topology with Annular Composite Flywheel Results.

A “ring” PGA topology was also considered. This topology shown in Figure 8a, contains 20 islands with the same total number of individuals (divided equally among the islands) as the iiGA topology shown in Figure 7a. The flywheel geometric resolution was 24 rings, containing 9,982 computational degrees of freedom. In all studies, the composite flywheel material was E-glass.



8a. “Ring” PGA Topology.



8b. Best Ever Flywheel from “Ring” PGA within the time used by iiGA.

Figure 8. 20-Ring Island GA Topology with Annular Composite Flywheel Results.

6.0 RESULTS ON COMPOSITE ANNULAR FLYWHEELS

Figure 7b displays the “best ever” annular composite flywheel at all the levels of geometric resolution. Also, Figure 7b compares the three-dimensional to the plane stress axisymmetric results. The plane stress results based on angular velocity are exaggerated shapes that are *artifacts* of the analysis. However, the plane stress results cannot be dismissed because they are the building blocks that helped to form the final “finely tuned” flywheels rapidly. Figure 8b displays the best ever flywheel found in the “ring” topology, which ran for the same amount of time (3 hours). Clearly the iiGA is more efficient in its search technique than the “ring” PGA. Also, the iiGA annular composite flywheel has a 35% increase in SED when compared to the PGA results.

7.0 DISCUSSION AND CONCLUSIONS

The iiGA offers some new tools for approaching difficult optimization problems. For many problems, the iiGA can be used to break down a complex fitness function into “sub-fitness” functions, which represent “good” aspects of the overall fitness. The iiGA can build solutions in a sequence of increasingly refined representations, spatially or according to some other metric. The iiGA can also use differing evaluation tools, even with the same representation. A simplified analysis tool can be used to quickly search for good building blocks. This, in combination with searching at various levels of resolution, makes the iiGA efficient and robust. Mimicking a smart engineer, the iiGA can first quickly evaluate the overall response of a structure with a coarse representation of the design and finish the job off by slowly increasing the levels of refinement until a “finely tuned” structure has been evolved. This approach allows the iiGA to decrease computational time and increase robustness in comparison with a typical GA, or even a typical parallel GA. This was illustrated when the iiGA found a flywheel with a 35% increase in SED when compared to the best flywheel found in the same amount of time by the “ring” topology parallel GA. It was demonstrated much more strongly with the results for the simpler problem with the known global optimum, in which all variants of iiGA found the solution unerringly and rapidly, and all variants of the SGA with local search and threshold accepting heuristics, and the parallel ring GA, never found the solution. Of course, finding the global optimum for a problem with a reduced search space does not guarantee that the iiGA will find the global optimum for more complex cases, but it at least lends plausibility to the idea that the iiGA methods are helpful in searching such spaces relatively efficiently for near-optimal solutions. In many engineering domains in which each design evaluation may take many minutes (or hours), the availability of such a method, parallelizable with minimal communication workload, could make good solutions attainable for problems not previously addressable.

REFERENCES:

1. C. Soto and R. Diaz, 1993, “Optimum Layout and Shape of Plate Structures using Homogenization,” *Topology Design of Structures*, pp. 407-420.
2. K. Suzuki and N. Kikuchi, 1990, “Shape and Topology Optimization by a Homogenization Method,” *Sensitivity Analysis and Optimization with Numerical Methods*, Vol. 115, pp. 15-30.

3. K. Suzuki and N. Kikuchi, 1991, “A Homogenization Method for Shape and Topology Optimization,” *Computational Methods in Applied Mechanics in Engineering*, Vol. 93, pp. 291-318.
4. E. Sangren, E. Jensen and J. Welton, 1990, “Topological Design of Structural Components using Genetic Optimization Methods,” *Sensitivity Analysis and Optimization with Numerical Methods, Vol. 115*, pp.31-43.
5. P. Hajela and E. Lee, 1997, “Topological Optimization of Rotorcraft Subfloor Structures for Crashworthiness Considerations,” *Computers and Structures*, Vol. 64, No. 1-4, pp.65-76.
6. C. Mares and C. Surace, 1996, “An Application of Genetic Algorithms to Identify Damage in Elastic Structures” *Journal of Sound and Vibration*, Vol. 195, No. 2, pp. 195-215.
7. C. Chapman and M. Jakiela, 1996, “Genetic Algorithm-Based Structural Topology design with Compliance and Topology Simplification Considerations,” *Journal of Mechanical Design*, Vol. 118, pp. 89-98.
8. S. Rajan, 1995, “Sizing, Shape and Topology Design Optimization of Trusses using Genetic Algorithms,” *Journal of Structural Engineering*, Vol. 121, No. 10, pp. 1480-1487.
9. A. Keane, 1995, “Passive Vibration Control via Unusual Geometeries: The Application of Genetic Algorithm Optimization to Structural Design,” *Journal of Sound and Vibration*, Vol 185, No. 3, pp. 441-453.
10. Y. Nakaishi and S. Nakagiri, 1996, “Optimization of Frame Topology Using Boundary Cycle and Genetic Algorithm,” *JSME International Journal*, Vol. 39, No. 2, pp. 279-285.
11. N. Queipo, R. Devarakonda and J. Humphery, 1994, “Genetic Algorithms for Theromosciences research: Application to the Optimized Cooling of Electronic Components,” *International Journal of Heat and Mass Transfer*, Vol. 37, No. 6, pp.893-908.
12. R. Flynn and P. Sherman, 1995, “Multi-Criteria Optimization of Aircraft Panels: Determining Viable Genetic Algorithm Configurations,” *International Journal of Intelligent Systems*, Vol. 10, pp. 987-999.
13. H. Furuya and R. Haftka, 1995, “Placing Actuators on Space Structures by Genetic Algorithms and Effective Indicies,” *Structural Optimization*, Vol. 9, pp. 69-75.
14. W. F. Punch III, R.C. Averill, E.D. Goodman, S. C. Lin, and Y. Ding, February 1995,“Design Using Genetic Algorithms - Some Results for Laminated Composite Structures,” *IEEE Expert*, Vol. 10 (1), pp. 42-49.
15. W. F. Punch III, R.C. Averill, E.D. Goodman, S. C. Lin, Y. Ding, and Y.C. Yip, 1994, “Optimal Design of Laminated Composite Structures using Coarse-Grain Parallel Genetic Algorithms,” *Computing Systems in Engineering*, Vol. 5, No. 4-6, pp. 414-423.
16. N. Kosigo, L. T. Watson, Z. Gural, and R. T. Haftka, 1993, “Genetic Algorithms with Local Improvement for Composite Laminate Design,” *Structures & Controls Optimization*, ASME, Aerosp. Div., New York, Vol. 38, 13-28.
17. T. Le Riche and R. T. Haftka, 1993, “Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm,” *AIAA Journal*, Vol. 31, No. 5, pp. 951-956.
18. A. Todoroki, K. Watanabe and H. Kobayashi, 1995, “ Application of Genetic Algorithms to Stiffness Optimization of Laminated Composite Plates with Stress-Concentrated Open Holes,” *JSME Internat. Journal*, 38(4), 458-464.
19. I. Parmee and H. Vekeria, 1997, “Co-Operative Evolutionary Strategies for Single Component Design,” *Proceedings of the Seventh International Conference on Genetic Algorithms*, Michigan State University, pp. 529-536.
20. G. Fabbri, 1997, “A Genetic Algorithm for Fin Profile Optimization,” *International Journal of Heat and Mass Transfer*, Vol. 40, No. 9, pp. 2165-2172.
21. N. Foster and G. Dulikravich, 1997, “Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms,” *Journal of Spacecraft and Rockets*, Vol. 34, No. 1, pp. 36-42.
22. J. Haslinger and D. Jedelsky, 1996, “Genetic Algorithms and Fictitious Domain Based Approached in Shape Optimization,” *Structural Optimization*, Vol. 12, pp. 257-264.
23. J. Wolfersdorf, E. Achermann and B. Weigand, May 1997, “Shape Optimization of Cooling Channels Using Genetic Algorithms,” *Journal of Heat Transfer*, Vol. 119, pp. 380-388.
24. G. Genta and D. Bassani, 1995, “Use of Genetic Algorithms for the Design of Rotors,” *Meccanica*, Vol. 30, 707-717.
25. R. Ruthenbar, 1989, “Simulated Annealing Algorithms: An Overview,” *IEEE Circuits and Devices Magazine*, Vol. 5, No. 1, pp. 19-26.
26. E. D. Goodman, 1996, “GALOPPS, The Genetic Algorithm Optimized for Portability and Parallelism System, User’s Guide,” Technical Report, Genetic Algorithms Research and Applications Group (GARAGe), Michigan State University, East Lansing, July, 1996, 100pp.
27. E. D. Goodman, R. C. Averill, W. F. Punch III and D. J. Eby, 1997, “Parallel Genetic Algorithms in the Optimization of Composite Structures,” WSC2 Proceedings , World Wide Web Conference, http://garage.cps.msu.edu/papers/GARAGe97-05-02.ps.
28. N. N. Schraudolph and R. K. Belew, “Dynamic Parameter Encoding for Genetic Algorithms,” Technical Report LAUR 90-2795 (revised), Los Alamos National Laboratories, 1991.