The Optimization of Flywheels using an Injection Island Genetic Algorithm

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

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 Many engineering design approaches such as "make-it-and-break-it" are simply out-of-date, and have been replaced by computer simulations that exploit various mathematical methods such as the finite element method to avoid costly design iterations.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.

Shape optimization of flywheels for the maximization of Specific Energy Density (SED, rotational energy per unit weight) is an appealing thought that has received its fair attention by researchers. The concept of a flywheel is as old as the axe grinder's wheel, but could very well hold the key to tomorrow's problems of efficient energy storage. The flywheel has a bright outlook because of the recent achievement of high specific energy densities. A simple example of a flywheel is a solid, flat rotating disk. The SED of a flat solid disk can be increased by varying the shape of the disk to redistribute the inertial forces induced from rotation. A flywheel stores kinetic energy by rotating a mass about a constant axis, which makes it easy to integrate flywheels into energy conservation systems. Some vehicles currently use flywheels during braking for regenerating energy lost during deceleration. Another practical application is energy storage in low earth orbit satellites, where photoelectric cells are exposed to 60 minutes of light to charge, followed by 30 minutes of darkness when stored energy must be used. Electrochemical energy storage (e.g., in batteries) is limited by low cyclic lifetimes, low longtime reliability and low specific energies, all major concerns in satellite applications. The flywheel is well suited for this application due to high cyclic lifetimes, longtime reliability and high specific energies. Also, large-scale flywheels could be used in energy plants to store huge amounts of energy. Finding practical applications for flywheels is not a problem, but optimizing the SED of flywheels, given a problem-specific set of parameters and constraints, provides a challenge.

This chapter steps through various approaches that have been designed to optimize elastic flywheels. First, a simple Genetic Algorithm (sGA) searches for the well-known constant stress flywheel profile while measuring fitness with a plane stress finite element analysis. Shortcomings of the plane stress analysis are demonstrated via the sGA discovering solutions that are *artifacts* of the simplified plane stress analysis. Concepts of Simulated Annealing (SA) and Threshold Accepting (TA) are presented. The description of an Injection Island Genetic Algorithm (iiGA) is given for the optimization of flywheels. An iiGA in combination with a finite element code is used to search for shape variations to optimize the Specific Energy Density of flywheels. 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 measures fitness with 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.

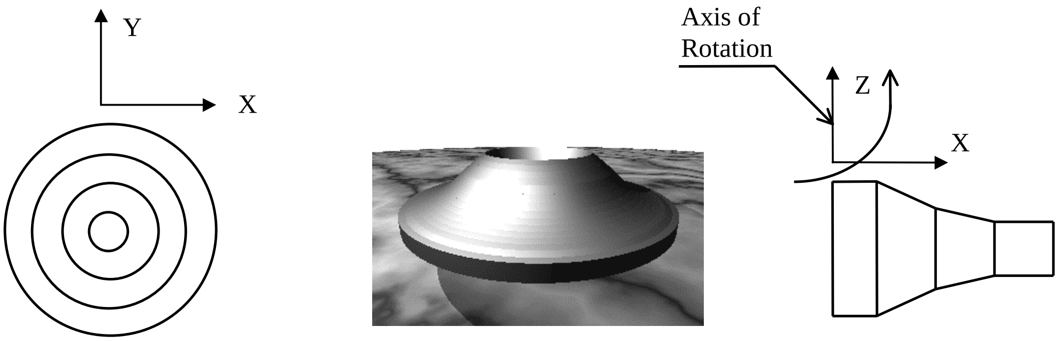
For the flywheel problem treated here, the lowest level of the iiGA searches with a simple axisymmetric plane stress finite element model (with a “sub-fitness” function), which quickly finds “building blocks” to inject into a series of GA populations using several more refined, axisymmetric, three-dimensional finite element models. 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 [Goodman, 1996] 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.



Top View 3-D View Side View

Figure 1. Visual Display of Flywheel.

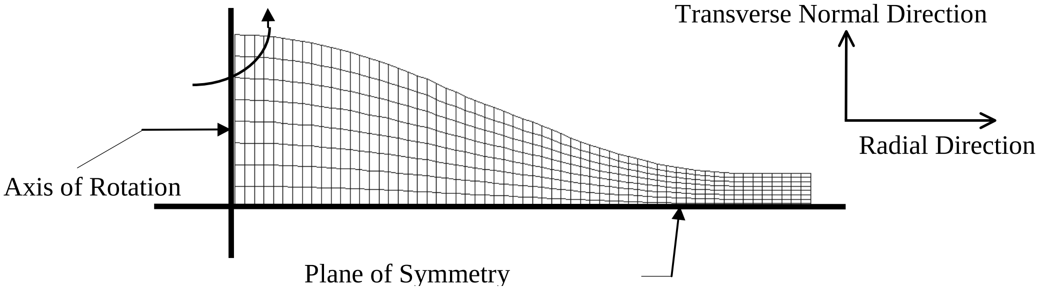


Figure 2. Typical Flywheel Model.

A greatly simplified design space (containing two million possible solutions) will be examined. The search space was enumerated, yielding the global optimum. The success and speed of many methods, including several variations of an iiGA can now be compared. The iiGA methods always locate the global optimum while other methods never locate the global optimum. Hybridizing the iiGA with a local search operator and a Threshold Accepting (TA) search at the end of each generation provide the fastest solutions, without sacrificing robustness. Results from an unconstrained optimization problem with a larger design space will be presented. Finally, a constraint will be placed on the maximum allowable angular velocity of the flywheel to create a more challenging optimization problem. Results from this more challenging problem are compared for various optimization approaches that include:

Parallel GAs (PGAs) that have various topological structures, iiGAs and hybrid iiGAs. The hybrid iiGA greatly outperforms the PGA in terms of fitness and search efficiency for this given problem.

2.0 Optimization Methods

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 [Soto and Diaz, 1993; Suzuki and Kikuchi, 1990; Suzuki and Kikuchi, 1991], 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 extrema [Sangren *et al.*, 1990]. 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 topic of this chapter: for other examples, see Hajela and Lee, [1997], Mares and Surace [1996], Chapman and Jakiela [1996], Rajan [1995], Keane [1995], Nakaishi and Nakagiri [1996], Queipo *et al.* [1994], Flynn and Sherman [1995], Furuya and Haftka [1995]. Recently, GA's have been successfully applied in the optimization of laminated composite materials [Punch *et al.*, 1994], [Punch *et al.*, 1995], [Kosigo *et al.*, 1993], [Le Riche and Haftka, 1993] [Todoroki *et al.*, 1995]). The authors of this chapter have used an iiGA in the design of laminated composite structures; and others have applied the iiGA to other engineering problems {*e.g.,* [Parmee and Vekeria, 1997])}. Others use different GA approaches (see Le Riche and Haftka [1993], Todoroki *et al.* [1995]). Several authors have dealt with the application of GA's to shape optimization problems. Fabbri [1997] and Foster and Dulikravich [1997] used GA's to find optimal shapes based on various polynomials, while Haslinger and Jedelsky [1996] present the concept of fictitious domains to generate new shapes. Wolfersdorf [1997] reduced computational costs associated with generating meshes for finite element evaluations by a point heat sink approach. Genta and Bassani [1995] 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 Genta and Bassani have already performed optimization of flywheels using a simple GA, this chapter differs in many respects: Genta and Bassani *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 chapter allows for ring thickness to be *randomly* chosen in the initial population; Genta and Bassani searched for shapes using a simple GA while this chapter will present various optimization approaches such as Threshold Accepting (TA), GA's, iiGA's and hybrid techniques; Genta and Bassani based fitness on a *single objective* in each run while *multiple fitness* definitions where used *concurrently* in each iiGA run for this chapter; Genta and Bassani measures fitness only with a *plane stress evaluation* while the current chapter 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 a GA search. These obstacles are nearly always present in interesting structural optimization problems. This chapter 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 chapter will also show that an ordinary parallel GA using the refined axisymmetric three-dimensional finite element model requires excessively long search times, in comparison to an iiGA approach which employs both the axisymmetric plane stress and three-dimensional finite element models.

An eventual goal of this effort is to develop tools for multi-criterion optimization of large-scale, three-dimensional composite structures, using an iiGA that searches at various levels of resolution and model realism. This technique incorporates several simultaneous and interconnected searches, including some that are faster (but often less accurate). This approach is constructed 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.

2.1 Simulated Annealing

Simulated Annealing (SA) is a combinatorial optimization technique that is based on the statistical mechanics of annealing of solids [Ruthenbar, 1989]. 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 an 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 greater, 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.

2.2 Threshold Accepting

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.

2.3 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.

Two primary classes of parallel GA’s are in common usage: coarse-grain and fine grain. Coarse-grain PGA’s divide the total population into independently breeding subpopulations with occasional migration among them. Fine-grain PGA’s typically distribute individuals in a multidimensional space (2-D or 3-D) and allow breeding only among near neighbors. 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.), coarse-grain 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. Note that this advantage is NOT shared by approaches (which we label "micro-grain" parallelism) which execute a sequential GA (i.e., perform exactly the same calculations) but distribute individuals among multiple processors for fitness evaluation – such an approach produces at best linear speedup.

2.4 Injection Island GA's

In Lin *et al.* [1994], we extended the notion of a coarse-grain (or island-) parallel GA to the iiGA (injection island GA), allowing each subpopulation to search at 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 often 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 to 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 developed by Goodman [1996]. The software can be run on one or multiple workstations (a single processor was used for all runs reported here). Islands with different levels of resolution evaluate fitness using either a simplified analysis that is computationally cheaper or a refined, computationally expensive analysis (see Figure 3). 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. This will be demonstrated later in the chapter.

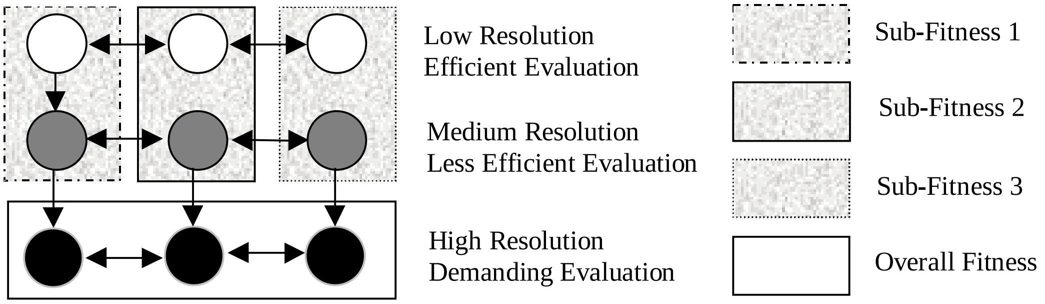


Figure 3. An iiGA that searches with multiple fitness definitions at various levels of resolution with evaluations that vary in levels of complexity, accuracy and computational efficiency.

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 (see Figure 3). iiGA's take advantage of the low communications bandwidth 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) [Schraudolph, 1991] and ARGOT [Schaefer, 1987] 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.

3.0 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.

4.0 Searching for the Constant Stress Profile

A simple approach was first taken using a sGA to search for the well known constant stress profile [Ugural, 1995] for a solid isotropic flywheel measuring fitness with a plane stress finite element model. The fitness function was defined as SED (amount of rotational energy stored per unit mass). The constant stress profile requires a natural boundary condition placed at the end of the disk (this can be accomplished for example by placing a metal "band" around the end of the flywheel). sGA runs had a population size of 300 and a 75% crossover rate. All GA runs used elitism (guaranteed survival of best individual) and one-point crossover with a 1% mutation rate.

Figure 4 shows the best solution in the 400th generation. The simplified plane stress evaluation ignores in-plane shearing and transverse normal stresses and will not capture the variation of radial and tangential stresses through the thickness of the flywheel. Thus, the GA places a large volume of mass near the end of the flywheel to increase the mass moment of inertia (Equation 2) in the fitness (equation 1). This large volume of mass at the end of the flywheel causes gross variations in ring thickness near the end of the flywheel which give rise to substantial through-thickness variations in stress. Because this is a violation of the plane stress assumption, the design is an *artifact* of the simplified plane stress evaluation. To help overcome this, the fitness was based on a "sub-fitness" function defined by the failure angular velocity of the flywheel. SED is dependent upon the square of the failure angular velocity (equation 1), so the search space defined by the failure angular velocity could contain potentially good designs that would be found in the search space based on SED. Also, basing fitness on angular velocity (not explicitly SED) eliminates the sGA's drive to find flywheels designed as in Figure 4. Figure 5 displays the evolution of the constant stress flywheel basing fitness on angular velocity while Figure 6 compares the theoretical constant stress profile to the constant stress profile found by the sGA. This demonstrates that the search space defined by the "sub-fitness" function

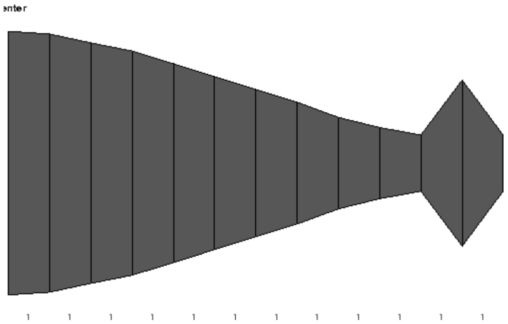
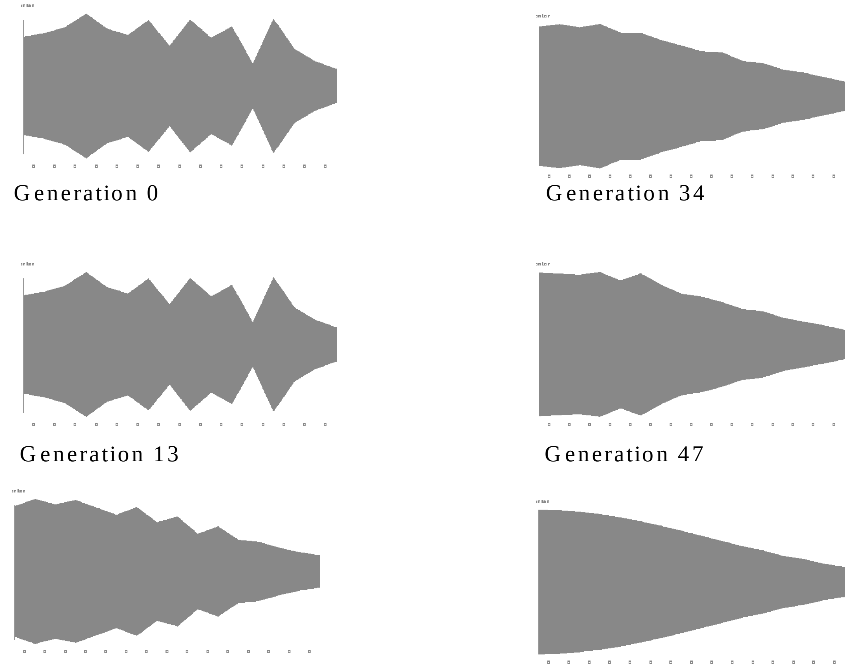


Figure 4. Typical example of a solution that is an artifact of a simplified plane stress evaluation. The sGA places a large volume of mass near the end the flywheel to increase the mass moment of inertia in the definition of the fitness (SED).



Generation 0

Generation 34

Generation 13

Generation 47

Generation 21

Generation 400

Figure 5. The evolution of the constant stress flywheel. Fitness was measured with a plane stress finite element evaluation while maximizing angular velocity.

(failure angular velocity) provided not only potentially good designs, but fruitful designs. The sGA spent less than 5 minutes (on a SUN SPARC Ultra) searching for the constant stress profile.

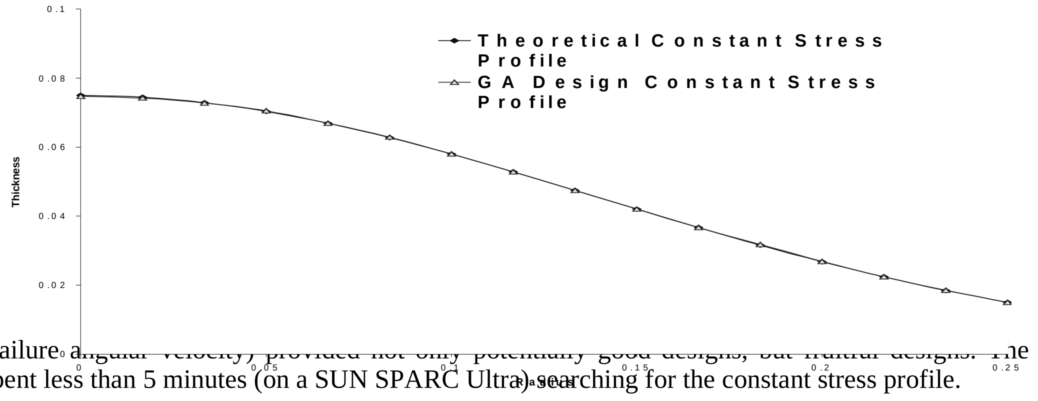


Figure 6. Comparison of theoretical to the GA-designed constant stress profile.

Next, the natural boundary condition at the outer radius of the flywheel was made homogeneous to design a flywheel that does not require a prescribed force at the outer radius of the flywheel. Figure 7 displays a GA designed flywheel profile that has a 55% increase in SED when compared to the theoretical constant stress flywheel.

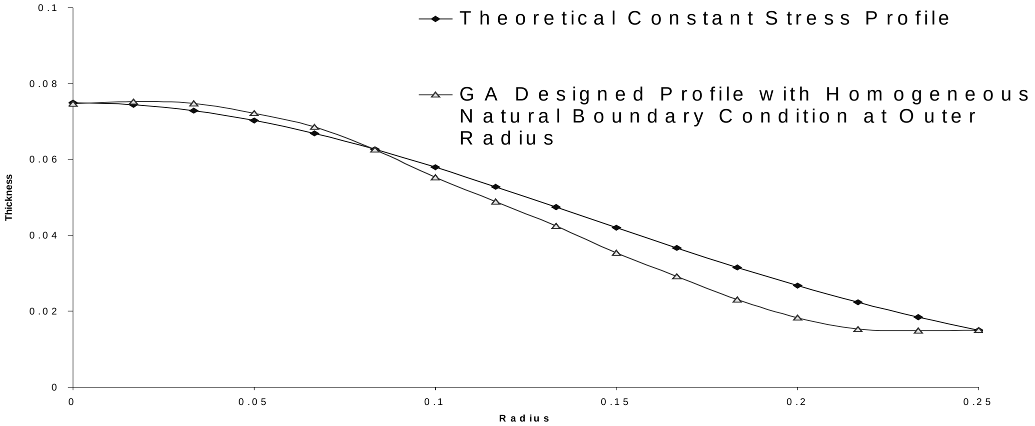


Figure 7. Comparison of theoretical constant stress profile to the GA designed profile (homogeneous natural boundary condition). A 55% increase in SED was achieved in the GA-designed flywheel.

Finally, an optimal profile for an annular isotropic flywheel was sought with the sGA measuring fitness with the plane stress finite element evaluation while basing fitness on a.) SED and b.) the "sub-fitness" function (failure angular velocity) in two independent runs. Figure 8a and 8b display the best flywheel found in the 400th generation basing fitness on SED and angular velocity, respectively. The sGA has discovered designs that have steep gradients in ring thickness, which violate the condition of plane stress. The flywheel profile in Figure 8a and 8b are artifacts of the plane stress analysis. These artifacts (Figure 4, 8a and 8b) are motivations to utilize a more refined analysis. A more refined analysis will be more robust and also allow the fitness to be based on SED (rather than a "sub-fitness" function).

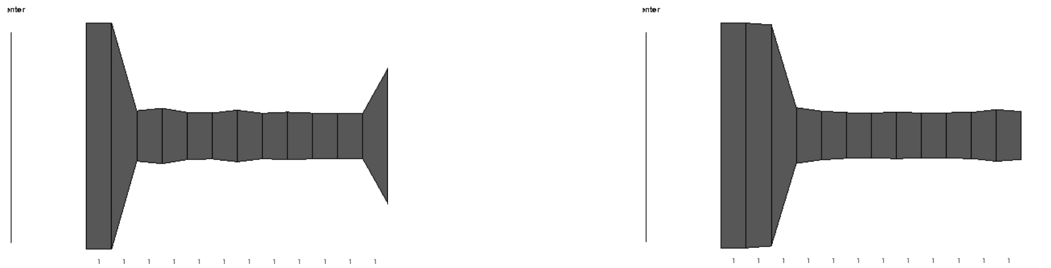


Figure 8a. Fitness based on SED. **Figure 8b. Fitness based on angular velocity.**

Figure 8. The best solutions found by two separate sGA runs in the 400th generation. These illustrate typical examples of artifacts (steep gradients in ring thickness) which violate the assumed condition of plane stress.

5.0 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 solid isotropic flywheel that contains 6 concentric rings (i.e., 7 heights) with 8 possible values for each height (see Figure 9b) created a design space of 8^7 or about 2 million possible designs. Using a coarse (962 DOF), axisymmetric 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 thickness 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. 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.

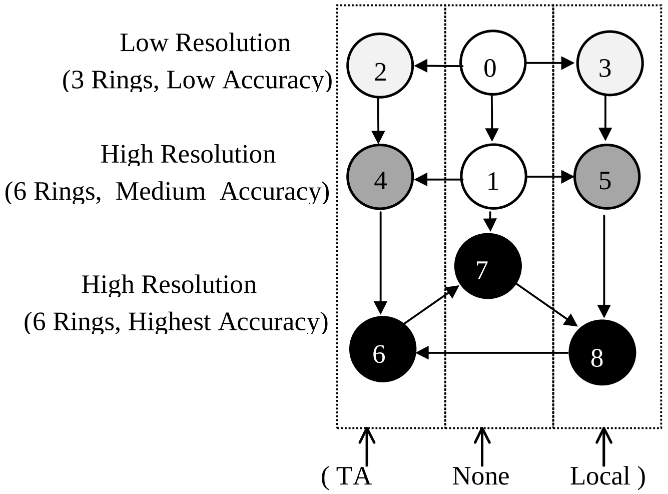


Figure 9a. Simplified iiGA topology.

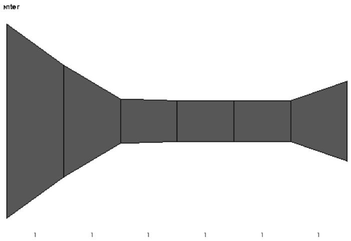


Figure 9b. Typical coarse flywheel design (6 rings).

KEY:	░	▒	●	○
Evaluation Tool:	3-D FEM (130 DOF)	3-D FEM (430 DOF)	3-D FEM (962 DOF)	Plane Stress FEM (7, 14 DOF)
Fitness Definition:	SED	SED	SED	Angular Velocity

Figure 9. Simplified injection island GA topology with coarse flywheel representation.

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 (0.001 seconds per evaluation) is up to 1000 times faster than the most refined three-dimensional evaluation of stress (for this analysis). To make the iiGA search less computationally intensive and more robust, the iiGA shown in Figure 9a 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 9a. 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. 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 9a 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. The authors believe that the process is not very sensitive to the particular parameters (such as genetic operator rates and number of migrants) chosen, and did not find it necessary to tune the parameters-they were set *a priori* based on the intuitions described above. Of course, the number of generations per cycle per island could increase overall run time if this parameter is significantly increased in islands that measure fitness with a computationally expensive analysis.

6.0 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 10 displays the fitness 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, which leads 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 11 and 12 display the fitness as a function of time for the iiGA (same topology as Figure 9a) and hybrid iiGA (Figure 9a, TA/None/local), respectively. All figures that display fitness as a function of time are reevaluated with the most accurate three-dimensional finite element model (962 DOF) to insure that all solutions are compared with the same “measuring stick” (the plane stress analysis will predict an overly optimistic fitness when compared to the more refined analysis). The iiGA alone found the global solution in 768 seconds (average), while the hybrid iiGA (Figure 9a, 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 10, 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 10 shows the iiGA quickly finding “building blocks” at low levels of resolution that are injected into islands of higher resolution. Figure 11 displays the hybrid iiGA (Figure 9a, TA/none/Local) benefiting from the combination of TA and local search heuristics. Figures 10-12 only display the first 1000 seconds because no better solutions were ever found thereafter.

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

Table 1. Comparison of Results of Various Optimization Approaches.

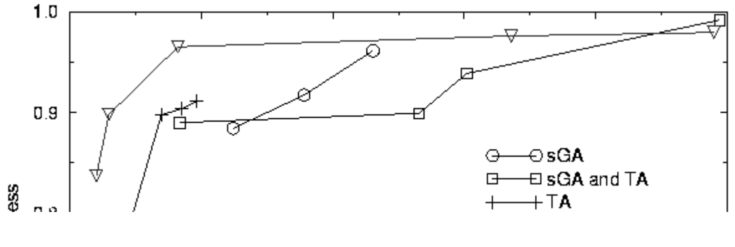


Figure 10. Fitness as a function of time on a single processor for a typical run of a simple GA, GA with TA, and simple GA with local search method.

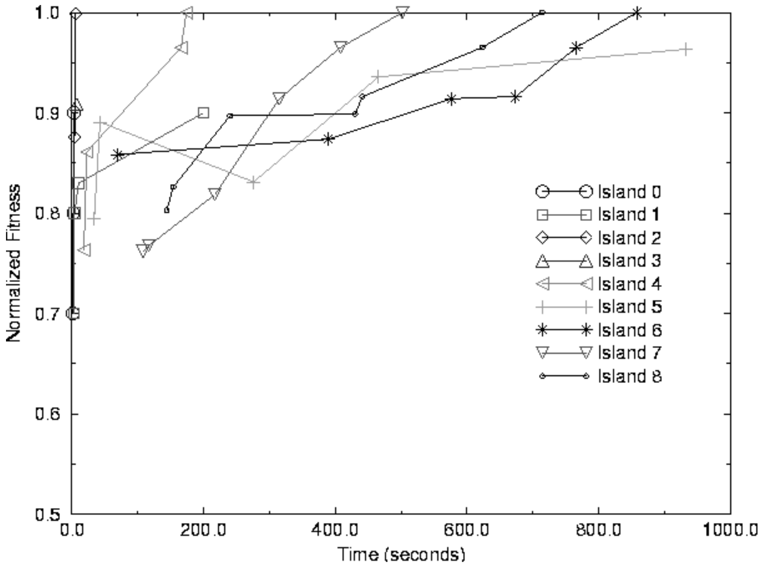


Figure 11. Fitness as a function of time on a single processor for a typical iiGA run.

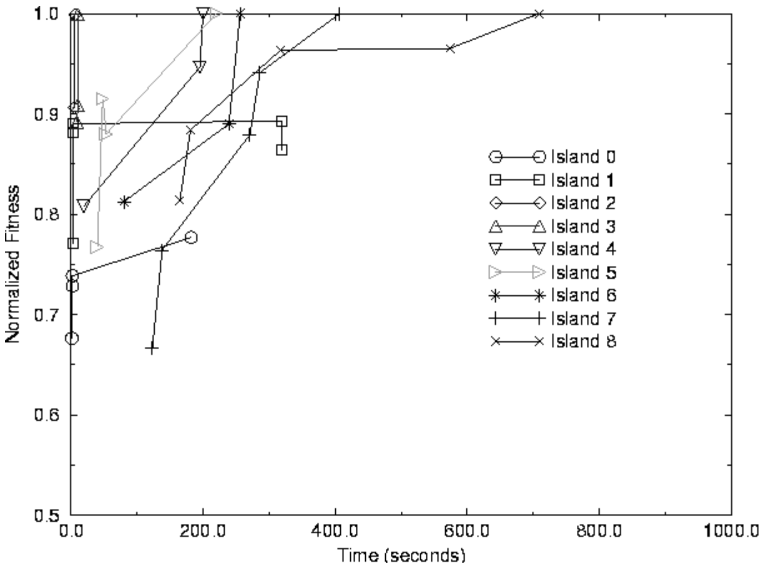


Figure 12. Fitness as a function of time on a single processor for a typical hybrid iiGA that incorporated TA and local search methods.

7.0 Sear

In this section, a much harder flywheel optimization problem is defined in order to compare results from PGAs (that have various topological structures), iiGAs and hybrid iiGAs. Two main changes were made to increase the problem difficulty: various constraints were added and a much larger search space was defined.

Often it is desirable to have an upper bound on the maximum allowable angular velocity of the flywheel in the design search space. Another possible goal might be to reduce “air gap” growth in annular flywheels (displacement of the inner radius due to forces induced from rotation). Constraints on a maximum allowable angular velocity and “air gap” growth will be developed by first considering the unconstrained version of the optimization problem with a hybrid iiGA.

A much larger search domain was created to increase the problem difficulty. A 24-ring flywheel with 1024 heights per thickness with 32 material choices created a huge design space. Table 2 lists all isotropic material properties, materials 1-3 have their Young’s modulus, density and strength recombined, representing 3^3 (27) materials with materials 4-8 representing the final five materials

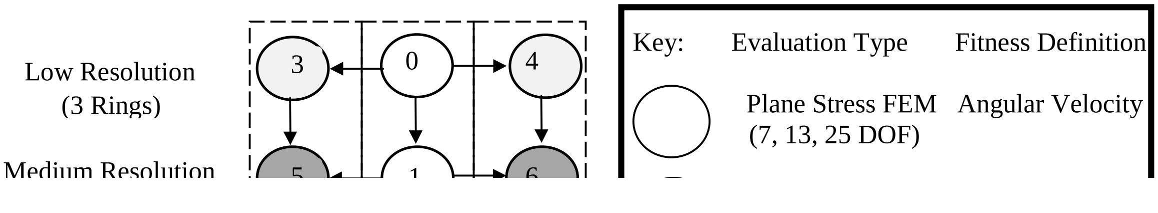
Material	Young’s Modulus (GPa)	Density (kg/m^3)	Strength (MPa)	Poisson’s Ratio
1*	10	1.5	100	0.25
2*	75	3.0	250	0.25
3*	200	9.0	400	0.25
4	140	1.5	1500	0.25
5	50	1.5	1600	0.25
6	15	1.5	250	0.25
7	45	1.5	150	0.25
8	3	1.5	85	0.25

Table 2. Material Properties.

7.1 The Unconstrained Optimization Problem

Since no previous numerical information is known about typical ranges of angular velocities and “air gap” growth, the unconstrained problem will be first approached with a hybrid iiGA basing overall fitness on SED. Again, to make the GA search less computationally intensive and more robust, a hybrid iiGA as shown in Figure 13 was designed. Islands that use similar special search heuristics (local, TA or none) are grouped together. 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 104 degrees of freedom. Islands 5 and 6 are medium in geometric resolution (6 rings), containing 372 degrees of freedom. Islands 7 and 8 are high in geometric resolution (12 rings), having 2,606 degrees of freedom. Islands 9 through 11 are the highest in geometric resolution (24 rings) with 13,250 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 12 generations per cycle before migrating the island’s best individual in accordance with Figure 13. Islands 3 and 4 had a crossover rate of 70%, population size of 200, and completed 8 generations per cycle before migrating the island’s best individual. Islands 5 and 6 had a 65% crossover rate, population size of 150 and completed 4 generations before migrating the island’s best individual. Islands 7 and 8 had a crossover rate of 60%, population size of 120 and the island’s best individual after evaluating 4 generations. Islands 9 through 11 had a crossover rate of 60%, population size of 86 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 The iiGA in Figure 13 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.



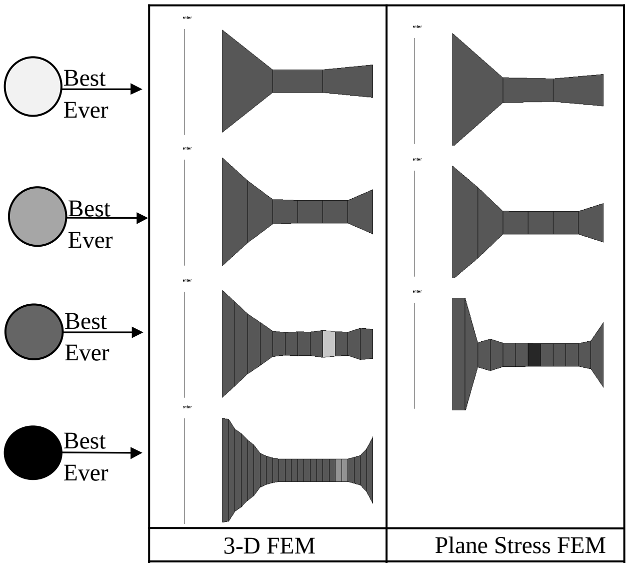


Figure 14. Best flywheel discovered at each level of resolution with a comparison of three-dimensional and plane stress solutions. The plane stress solutions are exaggerated variations of the three-dimensional counterparts.

Figure 14 displays the "best ever" annular composite flywheel at all the levels of geometric resolution for the unconstrained optimization problem. Also, Figure 14 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 rapidly form the final "finely tuned" flywheels.

Figure 15 displays the "raw" fitness of each island as a function of time for the unconstrained problem. The "raw" fitness is the actual SED measured by each island's specific finite element evaluation. Islands 0-2 measure "raw" fitness with an approximate but efficient evaluation based on angular velocity. The plane stress evaluation predicts fitness accurately for flywheels that have small gradients in ring thickness, but predicts excessively optimistic fitness values for designs that violate the plane stress assumption. Islands 3-8 evaluate fitness with a reduced number of degrees-of-freedom when compared to the refined evaluation in islands 9-11. Therefore we can expect discrepancies in the fitness values for islands 3-8 when reevaluating the designs with the most refined three-dimensional finite element model.

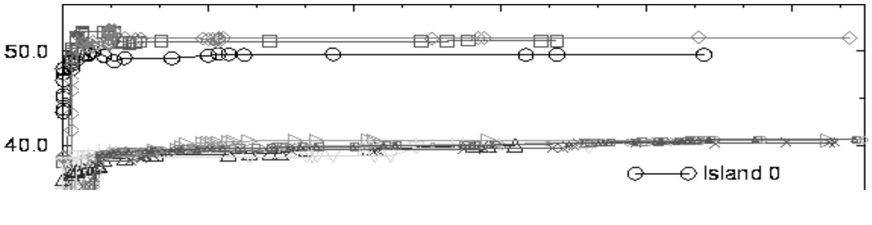


Figure 15. "Raw" fitness of each island as a function of time. Islands 0-2 predict excessively optimistic fitness values for designs that violate the plane stress evaluation while all other islands have realistic fitness values.

Figure 16 displays the fitness of annular multi-material flywheels as a function of time (reevaluated at the highest level of accuracy with the three-dimensional finite element model containing 13,250 degrees-of freedom). Figure 16 displays an expected response; islands 0-8 initially find good solutions but begin to find worse solutions as time progresses. These solutions contain building blocks that are used to help evolve islands at higher levels of resolution through injection and therefore cannot be discarded even though they have a low fitness when evaluated with the most refined finite element model. We can expect, but cannot discard what appears to be "noise" in the search. "Noise" occurs when the iiGA cannot decipher the differences between a solution that does or does not violate an assumption of the fitness evaluation (for example a plane stress finite element evaluation). If a high fitness is associated with solutions that violate the fitness evaluation, expect the iiGA to sooner or later exploit the evaluation's "Achilles heel" to improve the existing solutions in the population. This "noise" is typically more dominate near the end of a long run, where the design space is less "exciting" and more sensitive to slight variations in fitness because there is little more to gain in the designer's intended fitness definition. This effect can be seen in islands 0-8 in Figure 16, where the iiGA instantly finds good designs with the plane stress evaluation and then the designs progressively worsen as time progresses.

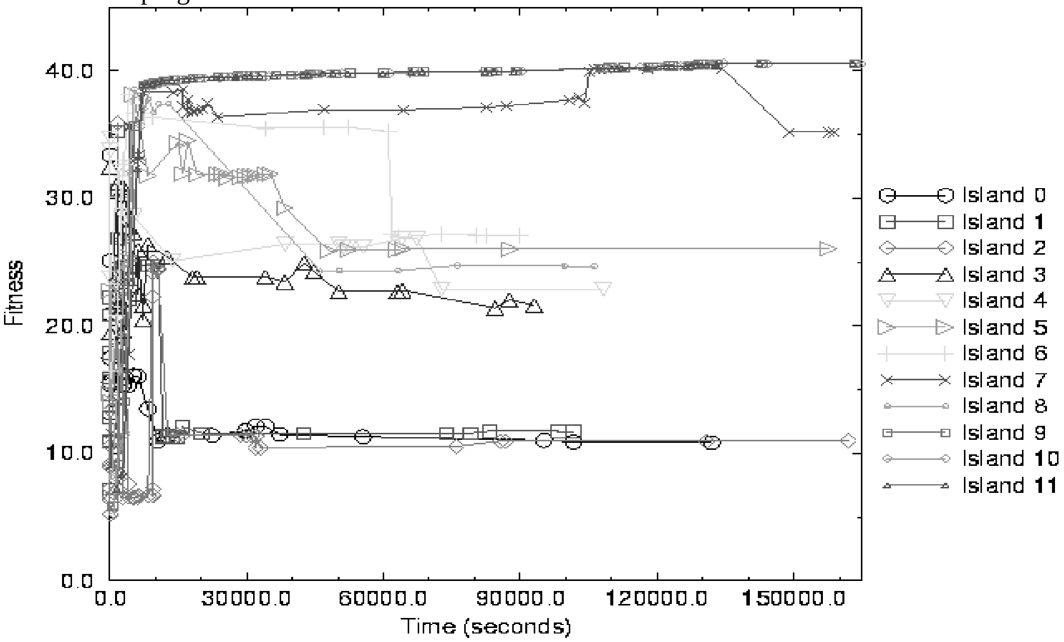


Figure 16. Reevaluated fitness (with most accurate evaluation) of each island as a function of time. Islands 0-8 display "noise" that develops from modeling complex structural response with less accurate evaluations.

7.2 The Constrained Optimization Problem

This section compares a constrained problem (with a huge search space) using PGAs (with various topological structures), iiGAs and hybrid iiGAs. The constrained optimization problem can be defined from numerical information based on the best design's maximum SED, angular velocity and "air gap" growth from the unconstrained problem. In no way is there a guarantee on discovering the global unconstrained solution with the hybrid iiGA, but rather the information gained from the unconstrained optimization problem is understood to be relative (possibly near global) and used as an estimate on constraint parameters in order to define a more difficult optimization problem. Constraints were enforced by the penalty method to ensure that designs not contained in the feasible set were still considered (but penalized).

The maximum values of SED, "air-gap" growth and angular velocity from the unconstrained problem were used to normalize the fitness function in the following manner:

$$\text{Fitness}_{\text{norm}} = C_1 \frac{SED}{SED_{\max}} - C_2 \frac{airgap}{airgap_{\max}} - C_3 \frac{\omega}{\omega_{\max}} \qquad 4)$$

C_1, C_2 and C_3 are weighting coefficients and are given in Table 3. The constraint C_3 was set to zero when the angular velocity of the design was below the maximum allowable angular velocity (which was chosen to be 75% of the angular velocity found in the best solution of the unconstrained problem). Also, equation 4 slightly penalizes flywheels that have large "air gap" growths.

C1	C2	C3
250	40	20

Table 3. Weighting Coefficient Values.

Table 4 contains average (found over five independent runs) fitness values with computation times for various GA runs that include: a PGA with a topological "ring" structure (Figure 17a), a PGA with a topological "matrix" or "toroid" structure (Figure 17b, similar numbers connect the structured migration) and some variations of the heuristic searches found in the hybrid iiGA depicted in Figure 12. Due to constraints on angular velocity, all islands in the iiGA (Figure 12) based fitness on equation 5. All PGA's measured fitness at the highest level of resolution (24 rings) with the most refined three-dimensional finite element model. All PGA's migrated the best solution every 3 generations and used a 65% crossover rate with 1% mutation with the same total number of individuals as the iiGA dispersed equally amongst 12 islands.

	Ring PGA	Matrix PGA	iiGA (none)	iiGA (TA)	iiGA (local)	iiGA (local/none/TA)
Fitness (Average over 5 runs)	200.8	194.4	206.1	212.3	199.1	205.4
Time (Days)	10	10	2	2	2	2

Table 4. Average Fitness (Five Independent Runs) for Various GA Approaches.

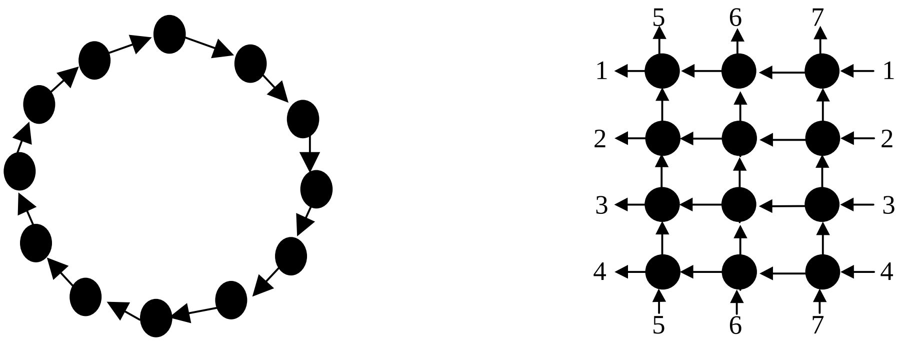


Figure 17a. "Ring" PGA topology.

Figure 17b "Matrix" PGA topology.

Figure 17. "Ring" and "Matrix" PGA topologies. All evaluations performed by the highest level of finite element accuracy and resolution.

Figure 18 compares the fitness as a function of time for a typical island for the "ring" PGA, "matrix" PGA, iiGA and various hybrid iiGA's. The PGA's display excessive computational effort when compared to all forms of the iiGA.

Figure 19 displays typical annular flywheels found by the iiGA, all hybrid iiGA’s, topological “ring” and “matrix” PGA’s. All designs are in the feasible set (satisfied the constraints). All designs display an increase in thickness at the end of the radius, which helps increase the mass moment of inertia in the SED term (Equation 1) for the normalized fitness definition (Equation 4) due to the constraint placed on angular velocity. All iiGA designs are similar in shape but have slight variations in material placement. The PGA designs are not as refined as the iiGA designs. The iiGA designs in Figure 18 have fitness values that are about 5% higher than the PGA designs, but the PGA’s designs required excessive computational effort.

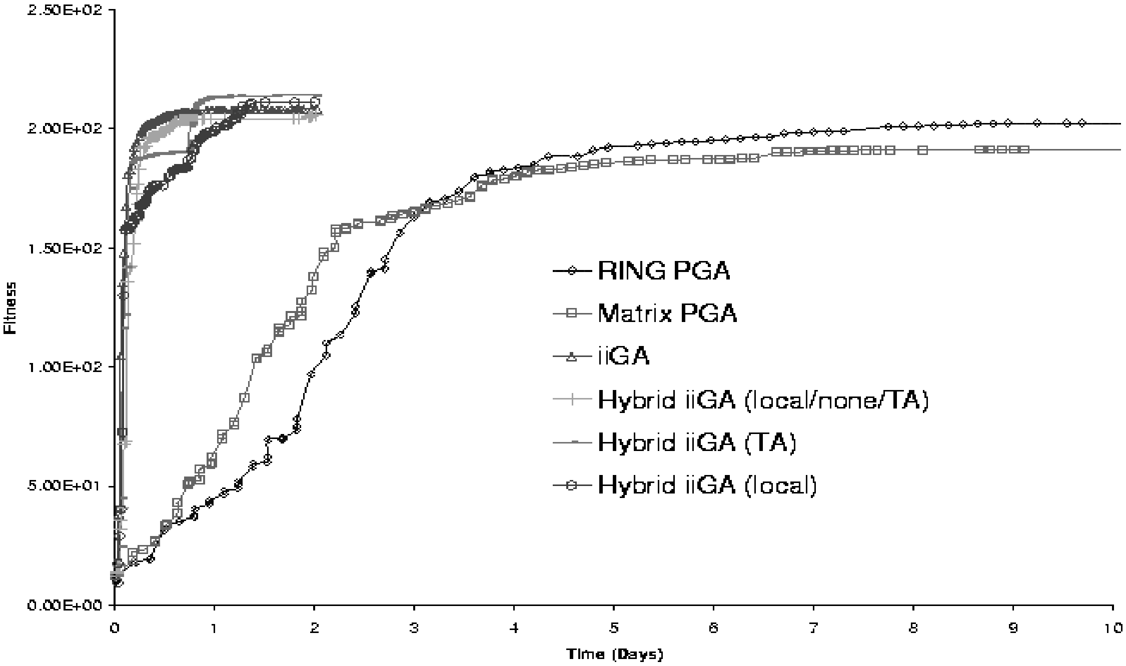
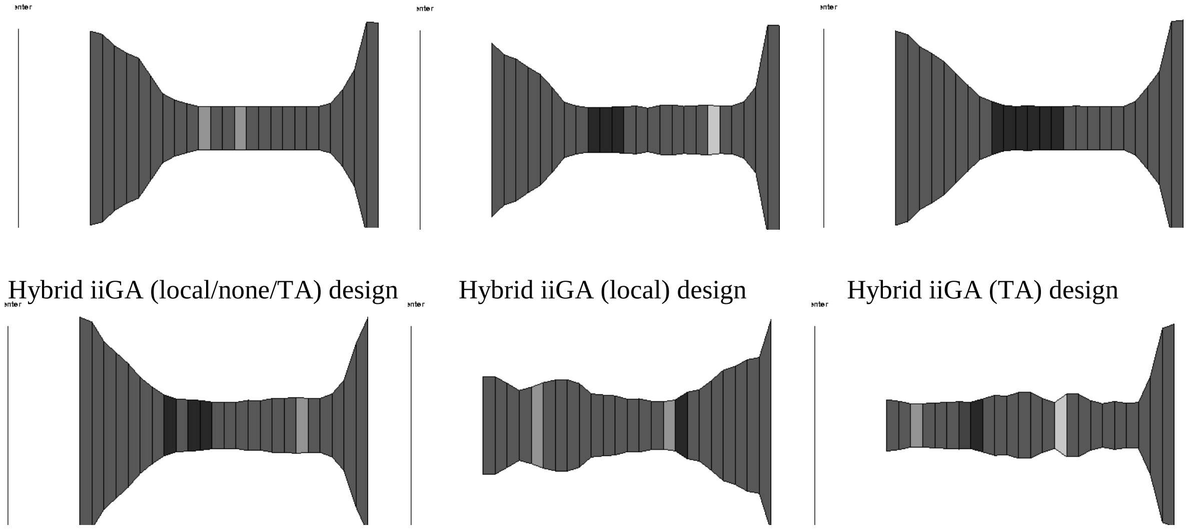


Figure 18. Comparison of fitness as a function of time for typical single island for a “ring” PGA, “matrix” PGA, iiGA and various hybrid iiGA’s ran on a single processor. PGA displays excessive computational efforts when compared to all forms of the iiGA.



Hybrid iiGA (local/none/TA) design Hybrid iiGA (local) design Hybrid iiGA (TA) design

iiGA design “Ring” PGA design “Matrix” PGA design

Figure 19. Typical designs found by all GA techniques. All iiGA flywheel designs are of similar shape with some variations in material placement. PGA and iiGA designed flywheels have noticeably different shapes near the inner radius.

8.0 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 demonstrated with the results for the simple 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. This was also demonstrated with the considerably more difficult constrained optimization problem where all topological versions of the PGA required excessive computational effort when compared to all versions of the iiGA. 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

Chapman, C. and Jakiela, M., 1996, "Genetic Algorithm-Based Structural Topology Design with Compliance and Topology Simplification Considerations," *Journal of Mechanical Design*, 118:89-98.

Fabbri, G., 1997, "A Genetic Algorithm for Fin Profile Optimization," *Int. J. of Heat and Mass Transfer*, 40:2165-2172.

Flynn, R., and Sherman, P., 1995, "Multi-Criteria Optimization of Aircraft Panels: Determining Viable Genetic Algorithm Configurations," *International J of Intelligent Systems*, 10: 987-999.

Foster, N., and Dulikravich, G., 1997, "Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms," *Journal of Spacecraft and Rockets*, 34:pp. 36-42.

Furuya, H., and Haftka, R., 1995, "Placing Actuators on Space Structures by Genetic Algorithms and Effective Indices," *Structural Optimization*, 9: 69-75.

Genta, G. and Bassani, D., 1995, "Use of Genetic Algorithms for the Design of Rotors," *Meccanica*, 30: 707-717.

Goodman, E., 1996, "GALOPPS, The Genetic Algorithm Optimized for Portability and Parallelism System, Release 3.2, User's Guide," Technical Report, Genetic Algorithms Research and Applications Group (GARAGe), Michigan State University, East Lansing, July, 1996, 100 pp.

Goodman, E., Averill, R., Punch, W. and Eby, D., 1997, "Parallel Genetic Algorithms in the Optimization of Composite Structures," in *Soft Computing in Engineering Design and Manufacture*, P. K. Chawdry, R. Roy and R. K. Pant, eds., Springer Verlag, 1998.

Hajela P., and Lee, E., 1997, "Topological Optimization of Rotorcraft Subfloor Structures for Crashworthiness Considerations," *Computers and Structures*, 64: 65-76.

Haslinger, J. and Jedelsky, D., 1996, "Genetic Algorithms and Fictitious Domain Based Approaches in Shape Optimization," *Structural Optimization*, 12:257-264.

Keane, A., 1995, "Passive Vibration Control via Unusual Geometries: The Application of Genetic Algorithm Optimization to Structural Design," *Journal of Sound and Vibration*, 185: 441-453.

Kosigo, N., Watson, L., Gurdal, Z. and Haftka, R., 1993, "Genetic Algorithms with Local Improvement for Composite Laminate Design," *Struct. & Controls Opt.*, ASME, Aero. Div., 38:13-28.

Le Riche, T. and Haftka, R., 1993, “Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm,” *AIAA J* , 31:951-956.

Lin, S.-C., Goodman, E. D., and Punch, W. F., “Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach,” *IEEE Conf. on Parallel & Distributed Processing*, Nov., 1994.

Mares, C. and Surace, C., 1996, “An Application of Genetic Algorithms to Identify Damage in Elastic Structures” *Journal of Sound and Vibration*, 195:195-215.

Nakaishi, Y. and Nakagiri, S., 1996, “Optimization of Frame Topology Using Boundary Cycle and Genetic Algorithm,” *JSME International Journal*, 39: 279-285.

Parmee, I. and Vekeria, H., 1997, “Co-Operative Evolutionary Strategies for Single Component Design,” *Proc. 7th Int. Conf. Genetic Alg.*, T. Baeck, ed., Morgan Kaufmann, San Francisco, 529-536.

Punch, W., Averill, R., Goodman, E., Lin, S.-C. and Ding, Y., 1995, “Design Using Genetic Algorithms - Some Results for Laminated Composite Structures,” *IEEE Expert*, 10:42-49.

Punch, W., Averill, R., Goodman, E., Lin, S.-C., Ding, Y. and Yip, Y., 1994, “Optimal Design of Laminated Composite Structures using Coarse-Grain Parallel Genetic Algorithms,” *Computing Systems in Eng.*, 5:414-423.

Queipo, N., Devarakonda, R. and Humphrey, J., 1994, “Genetic Algorithms for Thermosciences Research: Application to the Optimized Cooling of Electronic Components,” *Int. J. of Heat and Mass Transfer*, 37:893-908.

Rajan, S., 1995, “Sizing, Shape and Topology Design Optimization of Trusses using Genetic Algorithms,*” Journal of Structural Engineering*, 121:1480-1487.

Ruthenbar, R., 1989, “Simulated Annealing Algorithms: An Overview,” *IEEE Circuits and Devices Magazine*, 5: 19-26.

Schaefer, C., 1987 “The ARGOT Strategy: Adaptive Representation Genetic Optimized Technique”, Proc. 2nd Intl. Conf. on Genetic Alg., J. Grefenstette, ed., Lawrence Erlbaum, Cambridge, MA, 50-58.

Schraudolph, N. and Belew, R., 1991 “Dynamic Parameter Encoding for Genetic Algorithms,” Technical Report LAUR 90-2795 (revised), Los Alamos National Laboratories.

Sangren, E., Jensen, E., and Welton, J., 1990, “Topological Design of Structural Components using Genetic Optimization Methods,” *Sensitivity Analysis and Optimiz. with Num. Methods,* 115:31-43.

Soto, C. and Diaz, A., 1993, “Optimum Layout and Shape of Plate Structures using Homogenization,” *Topology Design of Structures*, pp. 407-420.

Suzuki, K. and Kikuchi, N., 1990, “Shape and Topology Optimization by a Homogenization Method,” *Sensitivity Analysis and Optimization with Numerical Methods*, 115:15-30.

Suzuki, K. and Kikuchi, N., 1991, “A Homogenization Method for Shape and Topology Optimization,” *Computational Methods in Applied Mechanics in Engineering*, 93:291-318.

Todoroki, A., Watanabe K. and Kobayashi, H., 1995, “ Application of Genetic Algorithms to Stiffness Optimization of Laminated Composite Plates with Stress-Concentrated Open Holes,” *JSME Internat. Journal*, 38(4), 458-464.

Ugural, A. and Fenster, S., *Advanced Strength and Applied Elasticity, 3rd Edition*, PTR Prentice Hall, 1995.

Wolfersdorf, J., Achermann, E. and Weigand, B., May 1997, “Shape Optimization of Cooling Channels Using Genetic Algorithms,” *J. of Heat Transfer*, 119: 380-388.