Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 327–340. Printed in the USA.

Optimal design of flywheels using an injection island genetic algorithm

DAVID EBY,[1] R.C. AVERILL,[1] WILLIAM F. PUNCH III,[2] AND ERIK D. GOODMAN[2]
[1]Computational Mechanics Research Group and [2]Genetic Algorithms Research and Applications Group (GARAGe), College of Engineering, Michigan State University, East Lansing, Michigan 48824, U.S.A.

(RECEIVED October 30, 1998; ACCEPTED June 3, 1999)

Abstract

This paper presents an approach to optimal design of elastic flywheels using an Injection Island Genetic Algorithm (iiGA), summarizing a sequence of results reported in earlier publications. An iiGA in combination with a structural finite element code is used to search for shape variations and material placement to optimize the Specific Energy Density (SED, rotational energy per unit weight) of elastic flywheels while controlling the failure angular velocity. iiGAs 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 axi-symmetric 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 use an axi-symmetric three-dimensional finite element code to "fine-tune" the structures. A greatly simplified design space (containing two million possible solutions) was enumerated for comparison with various approaches that include: simple GAs, threshold accepting (TA), iiGAs and hybrid iiGAs. For all approaches compared for this simplified problem, all variations of the iiGA were found to be the most efficient. This paper will summarize results obtained studying a constrained optimization problem with a huge design space approached with parallel GAs that had various topological structures and several different types of iiGA, to compare efficiency. For this problem, all variations of the iiGA were found to be extremely efficient in terms of computational time required to final solution of similar fitness when compared to the parallel GAs.

Keywords: Optimization; Automated Design; Flywheel; Genetic Algorithm and FEM

1. INTRODUCTION

New optimization problems arise every day in engineering practice. Sometimes such 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.

Reprint requests to: Erik D. Goodman, Genetic Algorithms Research Group (GARAGe), College of Engineering, Michigan State University, East Lansing, MI 48824, U.S.A. Phone: (517) 355-6453; Fax: (517) 432-0704; E-mail: goodman@egr.msu.edu

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) is an appealing thought that has received a fair amount of attention from researchers. The concept of a flywheel is as old as the axe grinder's wheel, but could well hold the key to tomorrow's problems of efficient energy storage. 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 by rotation.

The flywheel is modeled as a series of concentric rings (see Fig. 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

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

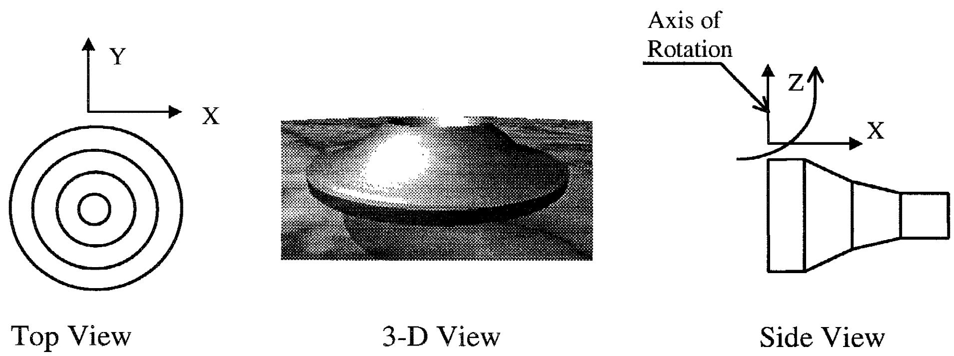


Fig. 1. Visual display of flywheel.

symmetry about the transverse normal direction and about the axis of rotation is used to increase computational efficiency.

For the flywheel problem treated here, the authors use *concurrently* a set of models of various levels of refinement, beginning with a simple axi-symmetric plane stress finite element model (with a "subfitness" function), which quickly finds "building blocks" to inject into a series of genetic algorithms (GA) populations using several more refined, axi-symmetric, three-dimensional finite element models. The overall fitness function for the genetic algorithm GALOPPS (Goodman, 1996) is the SED of the flywheel, which is defined as:

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

where ω is the angular velocity of the flywheel ("subfitness" 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.

This paper begins with a brief literature review in the general area of optimization methods as applied to flywheels. Simulated annealing, threshold accepting, and parallel genetic algorithms are reviewed, and Injection Island GAs (iiGAs) are described. Two different finite element models of flywheels are also summarized. Next, results are presented for a reduced design space that was enumerated to allow comparison of the efficiency of various GAs and iiGAs in finding a known global optimum. Next, an unconstrained flywheel optimization problem is defined, containing a huge number of possible designs. The flywheel optimization problem is then constrained and approached with a simple GA, parallel GAs with various topological structures, iiGAs and hybrid iiGAs (Eby et al., 1999).

1.1. 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, for example (Suzuki & Kikuchi, 1990, 1991; Soto & Diaz, 1993). 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 GA, is a multiple-point, directed stochastic search method that can be an effective optimization approach to a broad class of problems. The use

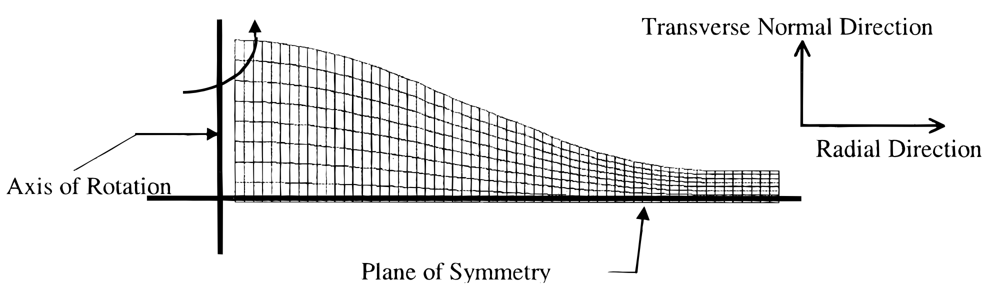


Fig. 2. Typical flywheel model.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

of GAs for optimal design requires that a large number of possible designs be analyzed, even though this number generally still represents only a minuscule fraction of the total design space. When each evaluation is computationally intensive, a traditional simple or parallel GA can thus be difficult to apply. iiGAs, described below, can help reduce the computational intensity associated with typical GAs 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 GAs is the main topic of this paper 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), and Furuya and Haftka (1995). Recently, GAs have been successfully applied in the optimization of laminated composite materials (Kosigo et al., 1993; Le Riche and Haftka, 1993; Punch et al., 1994, 1995; Todoroki et al., 1995). The authors of this paper have used an iiGA in the design of laminated composite structures (Goodman et al., 1998), and others have applied the iiGA to other engineering problems [for example, 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 GAs to shape optimization problems. Fabbri (1997) and Foster and Dulikravich (1997) used GAs to find optimal shapes based on various polynomials, while Haslinger and Jedelsky (1996) use the concept of fictitious domains to generate new shapes. Wolfersdorf et al. (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 Fig. 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 paper differs in many respects: Genta and Bassani *seeded* the initial population with flywheels that varied linearly in thickness from the inner to outer radii, while this paper allows for ring thickness to be *randomly* chosen in the initial population; Genta and Bassani searched for shapes using only a simple GA, while this paper will present various optimization approaches such as Threshold Accepting (TA), GAs, iiGAs, and hybrid techniques; Genta and Bassani based fitness on a *single objective* in each run while *multiple fitness* definitions were used *concurrently* in each iiGA run for this paper; Genta and Bassani measured 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 a GA 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 appropriate 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 paper will show how an efficient, simplified axi-symmetric 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 axi-symmetric three-dimensional finite element model requires excessively long search times, in comparison to an iiGA approach which employs both the axi-symmetric plane stress and three-dimensional finite element models.

An eventual goal of this effort is to develop tools for multicriterion 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.

1.2. Simulated annealing and threshold accepting

Simulated annealing (SA) methods begin with an initial solution that is often generated randomly, and try to perturb the solution to improve it (Ruthenbar, 1989). If the perturbation improves the value of the objective function, 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 earching 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 decrease the value of the objective function 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 higher ("worse") states 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),

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

the chance that a worse solution is accepted is small. This probability of accepting worse solutions is based on a Boltzmann distribution:

$$Pr[\text{Accept}] = e^{-\frac{\Delta E}{T}} \tag{3}$$

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

TA is a simplified version of SA. 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 (objective function value). This percentage decreases over the set of generations. This causes the TA in earlier generations to have a higher probability of accepting a worse solution, while later generations in the process are less likely to accept a worse solution.

1.3. Parallel genetic algorithms

Two problems associated with GAs are their need for many fitness evaluations and their propensity to converge prematurely to suboptimal solutions. An approach that ameliorates both of these problems is a coarse-grain or island parallel GA (PGA), in which the population is divided into subpopulations and recombination normally occurs only among individuals in the same subpopulation. This also produces a more realistic model of nature than a single large population. PGAs 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 GAs, which may pay a nontrivial computational cost for maintaining a structured population (demes, etc.) based on similarity comparisons (niching techniques, etc.), PGAs operate essentially as independent, smaller GA populations that 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, in the infrequent *migration* operation.

1.4. Injection island GAs

iiGAs represent our extension to the usual notion of parallel GAs (Lin, Punch, & Goodman, 1994). An iiGA allows heterogeneity of problem representation and/or objective (fitness) function definition among the various subpopulations. Migration of individuals between heterogeneous subpopulations requires that the user provide a function that remaps migrants from the donor subpopulation (using one representation) into the representation of the receiving subpopulation. An iiGA thus continually "injects" solutions (and hopefully, useful building blocks) from an inexpensive, low-resolution search into subpopulations searching more accurately (and expensively), helping to focus their search on promising regions. In such a case, migration from higher resolution subpopulations to lower resolution ones is not usually used, but is not barred, should it be desired. If the difference among subpopulations is in the types of models used, and not necessarily in model refinement or resolution, then "backward" diffusion or circular migration may prove useful. Of course, subpopulations may simply be using different models or fitness definitions of similar levels of complexity, but allowing the exchange of migrants to assist in more robust search (of Pareto surfaces, for example).

The remapping of the representation of migrants between subpopulations with different representations is most straightforward if the solution space of the donor subpopulation is a subset of the solution space of the recipient subpopulation—then, the migrating individual may be assigned exactly the same structure in the recipient as it had in the donor subpopulation. For many applications, the migrant will then be assigned the same fitness in the receiving subpopulation as it had before migrating (provided that the fitness function yields the same value for an equivalent structure in both representations). However, the migration can also be done by approximating the migrating individual's representation with a "similar" one in the representation of the receiving subpopulation. In that case, care must be taken that the "good" qualities of the migrant are not lost in this remapping. The fact that the fitness of the migrant then changes during migration is also problematic in a mechanical sense—it makes it harder to track real progress in the search, for example.

The injection occurs while all islands continue to search simultaneously, although it is also possible to (wholly or partially) re-initialize 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 developed by Goodman (1996). The software can be run on one or multiple PCs or 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 Fig. 3). The GA parameters—rates of crossover, mutation, migration, etc.—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 paper.

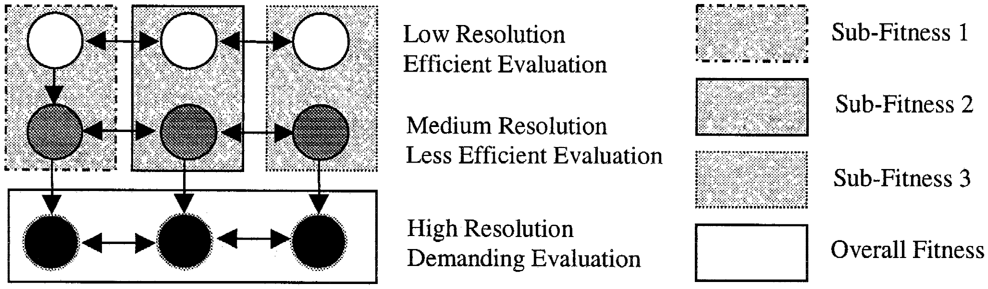


Fig. 3. An iiGA that searches with multiple fitness definitions at various levels of resolution with evaluations that vary in levels of complexity, accuracy, and computational efficiency.

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 Fig. 3). However, this is not a sufficient condition to guarantee effective search of the Pareto optimal solutions.

An iiGA takes 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 "subfitness" functions. An iiGA using islands of different resolutions has the following characteristics relative to other PGAs (which are often found to be an advantage for real-world engineering problems):

1. 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.
2. 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.
3. Islands connected in the hierarchy (islands with a parent–child relationship) share portions of the same search space because 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.
4. iiGAs embody a divide-and-conquer and partitioning strategy which has been successfully applied to many problems. In iiGAs, 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).
5. In iiGAs, nodes with smaller block size can find the solutions with higher resolution. Although Dynamic Parameter Encoding (DPE) (Schraudolph & Belew, 1991) and ARGOT (Schaefer, 1987) also deal with the resolution problem, using a zoom or inverse zoom operator, they are different from iiGAs. First, they are working at the phenotype level and only for real-valued parameters. iiGAs 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, iiGAs 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. FINITE ELEMENT MODELS OF FLYWHEELS

Two axi-symmetric 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

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

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. GLOBAL OPTIMUM FOR A SIMPLIFIED FLYWHEEL

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 iiGAs with various enhancements, a simplified flywheel problem was posed. A solid isotropic flywheel that contains six concentric rings (i.e., seven heights) with eight possible values for each height (see Fig. 9b) created a design space of 8^7 or about two million possible designs. Using a coarse (962 DOF), axisymmetric finite element model, it was possible to calculate the fitness based on the SED [Eq. (1)] of all of these designs, in about 50 h 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 was the first optimizer explored, and 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, 2200 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.

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 s 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 4a was

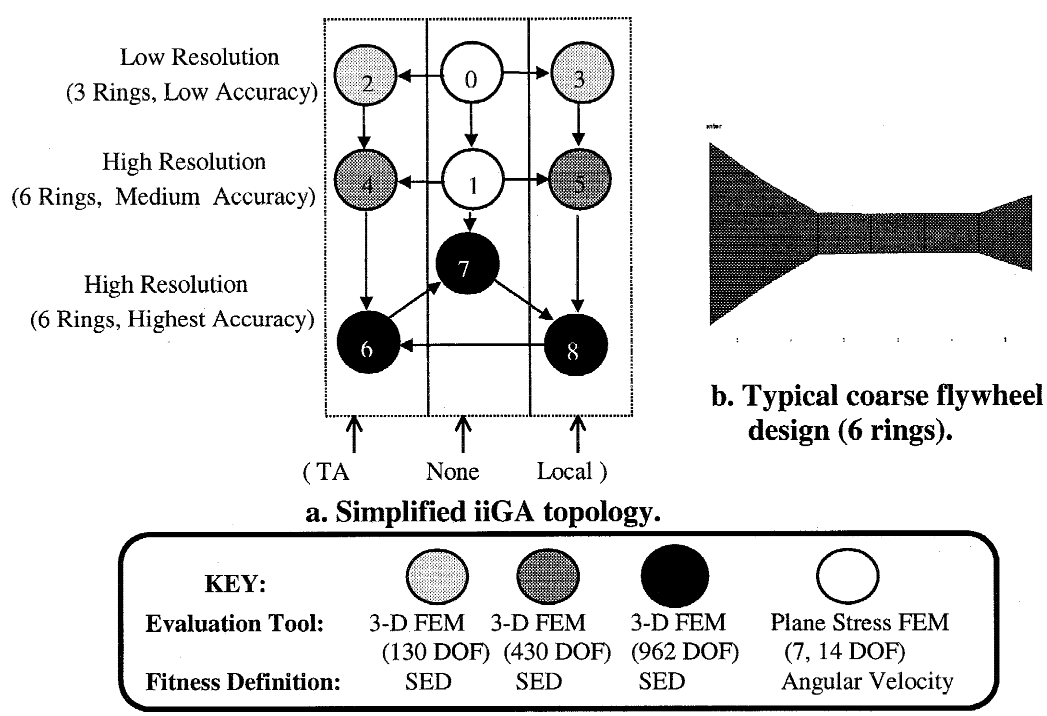


Fig. 4. Simplified injection island GA topology with coarse flywheel representation.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

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 4a. Islands 0 and 1 measured fitness with plane stress finite element code, basing fitness on the subfitness 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 six generations per cycle before migrating three individuals, evaluating fitness with the three-dimensional axi-symmetric finite element code basing fitness on SED (130 DOF). Islands 4 and 5 had a 65% crossover rate, population size of 200 and completed four generations before migrating individuals, measuring fitness with the three-dimensional axi-symmetric finite element code basing fitness on SED (430 DOF). Islands six through eight had a crossover rate of 60%, population size of 100, and received migrated individuals every two generations, measuring fitness with the three-dimensional axi-symmetric finite element code basing fitness on SED (962 DOF). Islands six through eight had 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 six through eight should fine tune potentially good designs (building blocks) received from the islands at a lower resolution. Figure 4a 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.

4. RESULTS OF GLOBAL OPTIMIZATION STUDY

Table 1 shows the results of the various methods. Each run lasted 6000 s 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 5 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 TA or local search methods. Without the local search or TA heuristics, the iiGA took an average of 768 s to find the global optimum. The hybrid iiGA that also used local search found the global optimum in 715 s (average) while the iiGA that incorporated the TA found the global solution in 674 s (average). Figures 6 and 7 display the fitness as a function of time for the iiGA (same topology as Fig. 4a) and hybrid iiGA (Fig. 4a, TA/None/Local), respectively. All figures that display fitness as a function

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 s
Hybrid iiGA with local search	Always found, 715 s
Hybrid iiGA with TA	Always found, 674 s
Hybrid iiGA with local search and TA	Always found, 417 s

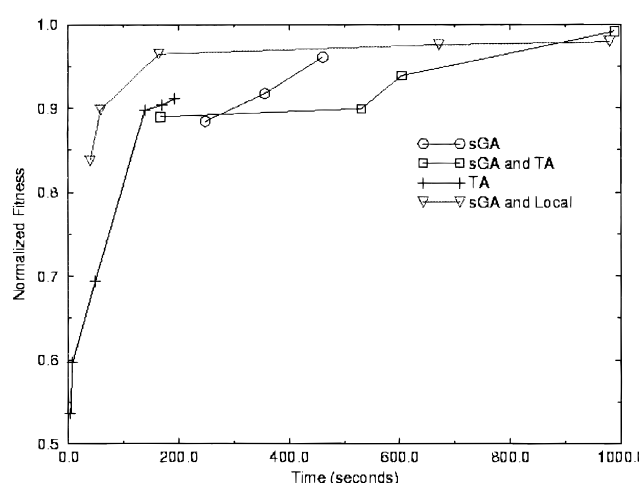


Fig. 5. Fitness as a function of time on a single processor for a typical run of a simple GA, GA with TA, and simple GA with local search method.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

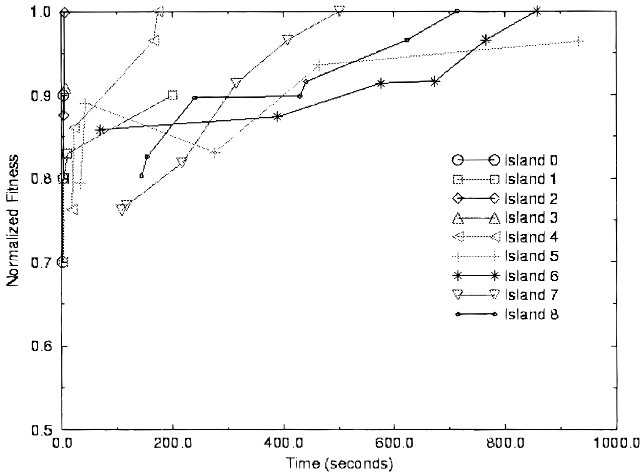


Fig. 6. Fitness as a function of time on a single processor for a typical iiGA run.

of time are reevaluated with the most accurate three-dimensional finite element model (962 DOF) to ensure 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 s (average), while the hybrid iiGA (Fig. 4a, TA/None/Local) found the global optimum in 417 s (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 5, 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 (Fig. 4a, TA/None/Local) benefiting from the combination of TA and local search heuristics. Figures 5–7 only display the first 1000 s because no better solutions were ever found thereafter.

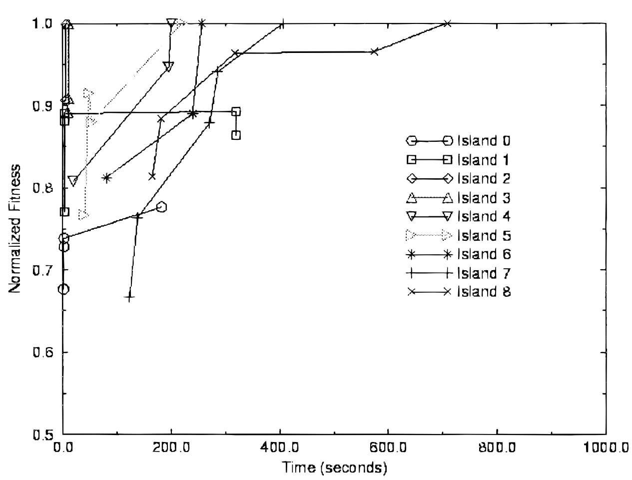


Fig. 7. Fitness as a function of time on a single processor for a typical hybrid iiGA that incorporated TA and Local search methods.

5. SEARCHING LARGER DESIGN SPACES USING iiGAs AND PGAs

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 design search space. Another goal would 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.

5.1. The unconstrained optimization problem

Because no previous numerical information was known about typical ranges of angular velocities and air gap growth, the unconstrained problem was first approached with a hybrid iiGA basing overall fitness on SED [Eq. (1)]. To make the GA search less computationally intensive and more robust,

Table 2. *Material properties*

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

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

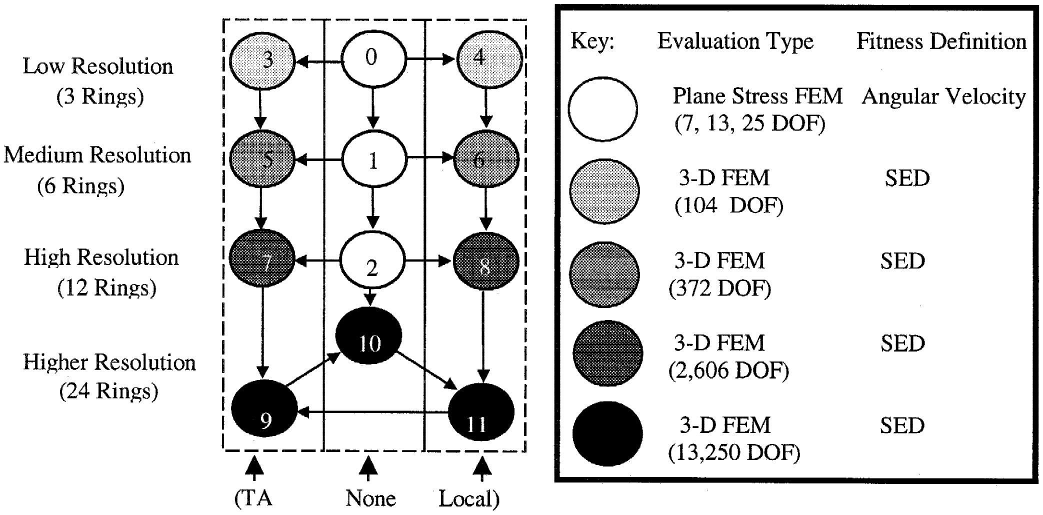


Fig. 8. Hybrid injection island GA topology.

a hybrid iiGA, as shown in Figure 8, 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 axi-symmetric 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 df. Islands 7 and 8 are high in geometric resolution (12 rings), having 2606 df. Islands 9 through 11 are the highest in geometric resolution (24 rings) with 13,250 df.

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 8. 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 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 in Figure 8 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 axi-symmetric three-dimensional finite element model to evaluate SED.

Figure 9 displays the "best ever" annular composite flywheel at all the levels of geometric resolution for the unconstrained optimization problem. Also, Figure 9 compares the three-dimensional to the plane stress axi-symmetric 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

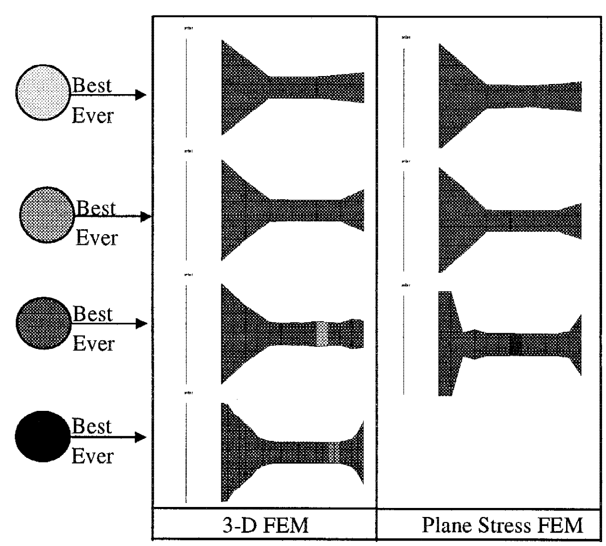


Fig. 9. Best flywheel discovered at each level of resolution with a comparison of three-dimensional and plane stress solutions. The plane stress solutions are exaggerated variations of the three-dimensional counterparts.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

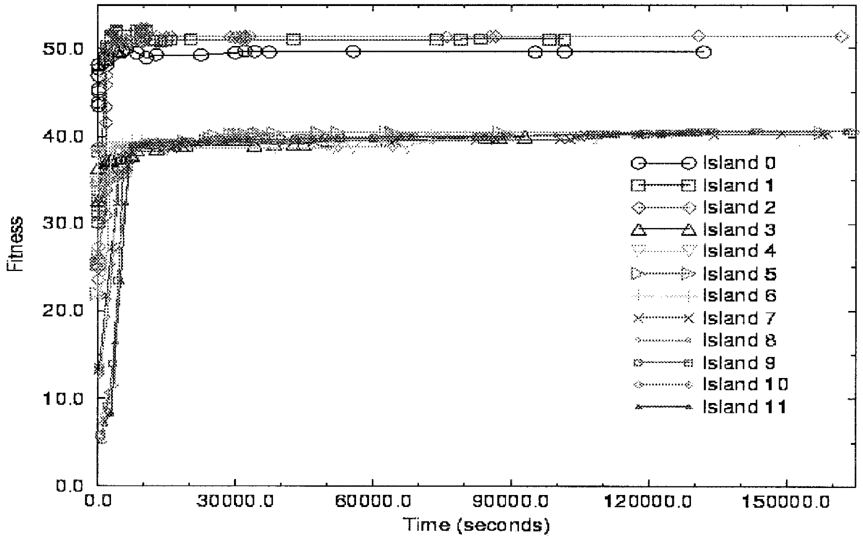


Fig. 10. "Raw" fitness of each island as a function of time. Islands 0–2 predict excessively optimistic fitness values for designs that violate the plane stress evaluation while all other islands have realistic fitness values.

they are the building blocks that helped rapidly form the final "finely tuned" flywheels.

Figure 10 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 df 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 11 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 df). Figure 11 displays an expected response; islands 0–8

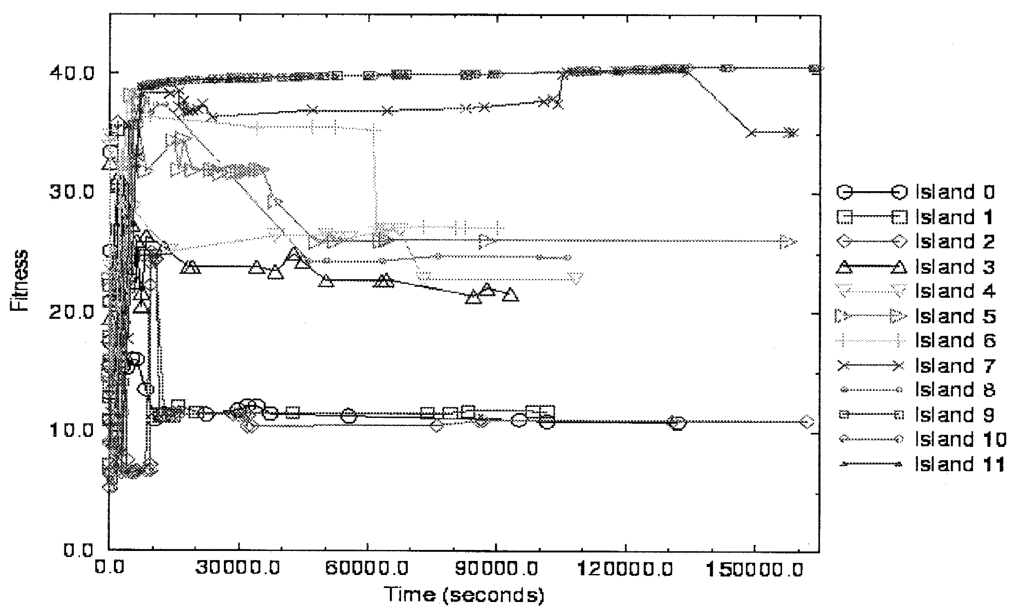
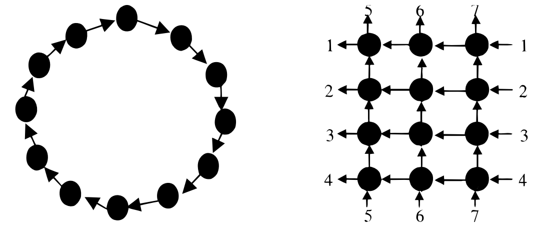


Fig. 11. Reevaluated fitness (with most accurate evaluation) of each island as a function of time. Islands 0–8 display "noise" that develops from modeling complex structural response with less accurate evaluations.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

Table 3. *Weighting coefficient values*

C_1	C_2	C_3
250	40	20



a. "Ring" PGA topology. b. "Matrix" PGA topology.

Fig. 12. "Ring" and "Matrix" PGA topologies. All evaluations performed by the highest level of finite element accuracy and resolution.

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 solutions that do or do 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, we 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 dominant 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 11, where the iiGA instantly finds good designs with the plane stress evaluation and then the designs progressively worsen as time progresses, when evaluated with most accurate finite element evaluation.

5.2. 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. There is no guarantee of 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 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). Alternatives to the penalty method, such as repair of chromosomes violating feasibility constraints, were not explored extensively, although the local search operator described later might have been used as part of such a scheme.

The constraint on maximum allowable angular velocity was enforced through the penalty method. The maximum values of SED, "air-gap" growth, and angular velocity from the unconstrained problem were used to normalize the fitness function. The fitness was defined by aggregating the normalized objectives (maximize SED while minimizing air gap growth) with the normalized constraint violation (maximum allowable angular velocity) in the following manner:

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

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, Eq. (4) slightly penalizes flywheels that have large air gap growths. For this problem, it was not difficult to determine appropriate weights for the various penalty terms, although that is sometimes an issue for constrained optimization problems.

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 (Fig. 12a), a PGA with a topological "matrix," or "toroid" structure (Fig. 12b, similar numbers connect the structured migration) and some variations of the heuristic searches

Table 4. *Average fitness (five independent runs) for various GA approaches*

	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

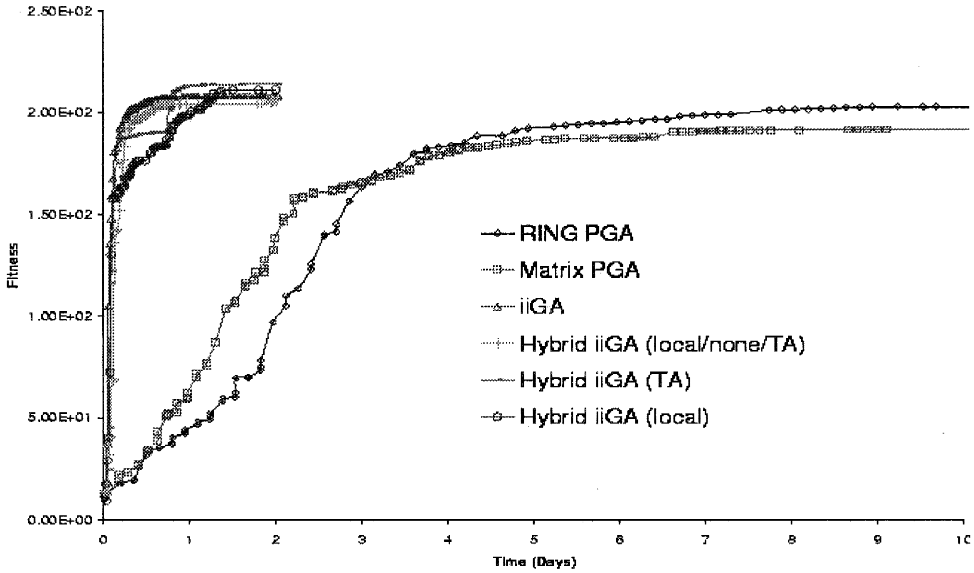


Fig. 13. Comparison of fitness as a function of time for typical single island for a "ring" PGA, "matrix" PGA, iiGA and various hybrid iiGAs ran on a single processor. PGA displays excessive computational efforts when compared to all forms of the iiGA.

found in the hybrid iiGA depicted in Figure 8. It would be detrimental to maximize the angular velocity in some islands while constraining the angular velocity in other islands: the posed constrained problem based fitness on Eq. (1) for all islands in the iiGA (Fig. 8). All PGAs measured fitness at the highest level of resolution (24 rings) with the most refined three-dimensional finite element model. All PGAs migrated the best solution every three 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.

Figure 13 compares the fitness as a function of time for a typical island for the ring PGA, matrix PGA, iiGA, and various hybrid iiGAs. The PGAs display excessive computational effort when compared to all forms of the iiGA.

Figure 14 displays typical annular flywheels found by the iiGA, all hybrid iiGAs, topological ring, and matrix PGAs. 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 [Eq. (1)] for the normalized fitness definition [Eq. (4)] due to the constraint placed on angular velocity.

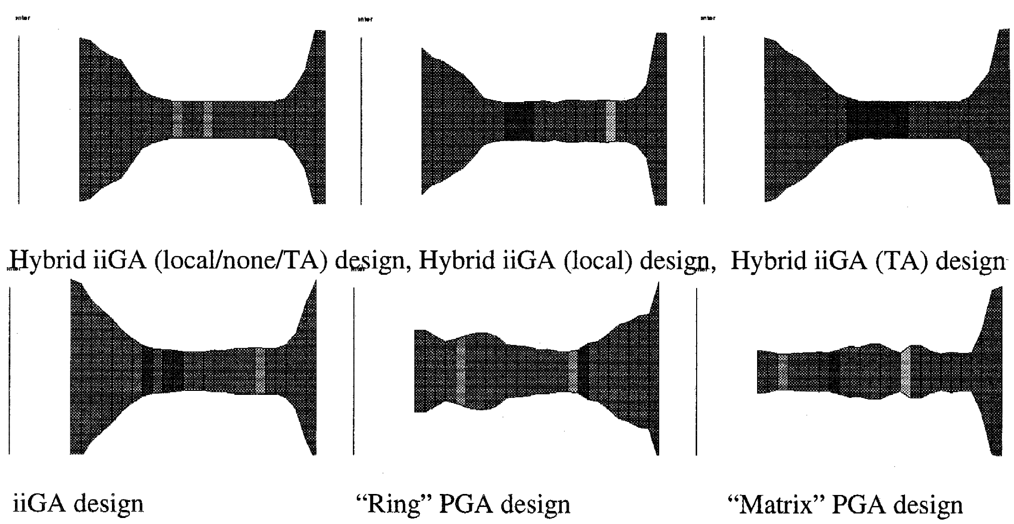


Fig. 14. Typical designs found by all GA techniques. All iiGA flywheel designs are of similar shape with some variations in material placement. PGA and iiGA designed flywheels have noticeably different shapes near the inner radius.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

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 13 have fitness values that are about 5% higher than the PGA designs, but the PGA designs required excessive computational effort.

6. DISCUSSION AND CONCLUSION

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 subfitness 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 s simple problem with s 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., & Jakiela, M. (1996). Genetic algorithm-based structural topology design with compliance and topology simplification considerations. *J. Mechan Design, 118*, 89–98.

Eby, D., Averill, R.C., Punch, W., & Goodman, E. (1999). The optimization of flywheels using an injection island genetic algorithm. In *Evolutionary Design by Computers*, (Bentley, P.J., Ed.), Morgan Kaufman, San Francisco, 1999, pp. 167–190.

Fabbri, G. (1997). A genetic algorithm for fin profile optimization. *Int. J. Heat and Mass Transfer 40*, 2165–2172.

Flynn, R., & Sherman, P. (1995). Multi-criteria optimization of aircraft panels: Determining viable genetic algorithm configurations. *Int. J. Intell. Syst., 10*, 987–999.

Foster, N., & Dulikravich, G. (1997). Three-dimensional aerodynamic shape optimization using genetic and gradient search algorithms. *J. Spacecraft Rockets 34*, 36–42.

Furuya, H., & Haftka, R. (1995). Placing actuators on space structures by genetic algorithms and effective indices. *Struc. Optimization 9*, 69–75.

Genta, G., & 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, user's guide.* Technical Report #960801, Genetic Algorithms Research and Applications Group (GARAGe), Michigan State University, East Lansing, MI.

Goodman, E., Averill, R., Punch, W., & Eby, D. (1998). Parallel genetic algorithms in the optimization of composite structures. In *Soft Computing in Engineering Design and Manufacturing*, (Chawdry, P.K., et al., Eds.), pp. 199–208. Springer Verlag, Berlin.

Hajela P., & Lee, E. (1997). Topological optimization of rotorcraft subfloor structures for crashworthiness considerations. *Comput. Struct., 64*, 65–76.

Haslinger, J., & Jedelsky, D. (1996). Genetic algorithms and fictitious domain based approaches in shape optimization. *Struct. Optimization, 12*, 257–264.

Keane, A. (1995). Passive vibration control via unusual geometries: The application of genetic algorithm optimization to structural design. *J. Sound Vibration 185*, 441–453.

Kosigo, N., Watson, L., Gurdal, Z., & Haftka, R. (1993). Genetic algorithms with local improvement for composite laminate design. *Structure & Controls Optimization, ASME 38*, 13–28.

Le Riche, T., & Haftka, R. (1993). Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm. *AIAA Journal 31*, 951–956.

Lin, S.-C., Punch, W., & Goodman, E. (1994). Coarse-grain parallel genetic algorithms: Categorization and new approach. *Proc. Sixth IEEE Conf. Parallel and Distributed Process*, 28–37.

Mares, C., & Surace, C. (1996). An application of genetic algorithms to identify damage in elastic structures. *J. Sound Vibration 195*, 195–215.

Nakaishi, Y., & Nakagiri, S. (1996). Optimization of frame topology using boundary cycle and genetic algorithm. *JSME Int. J., 39*, 279–285.

Parmee, I., & Vekeria, H. (1997). Co-operative evolutionary strategies for single component design. *Proc., Seventh Int. Conf. Genetic Algorithms*, pp. 529–536, Morgan Kaufmann, San Francisco, California.

Punch, W., Averill, R., Goodman, E., Lin, S.-C., & 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., & Yip, Y. (1994). Optimal design of laminated composite structures using coarse-grain parallel genetic algorithms. *Comput. Sys. Eng., 5*, 414–423.

Queipo, N., Devarakonda, R., & Humphrey, J. (1994). Genetic algorithms for thermosciences research: Application to the optimized cooling of electronic components. *Int. J. Heat Mass Transf., 37*, 893–908.

Rajan, S. (1995). Sizing, shape and topology design optimization of trusses using genetic algorithms. *J. Struct. Eng., 121*, 1480–1487.

Ruthenbar, R. (1989). Simulated annealing algorithms: An overview. *IEEE Circuits and Devices Magazine 5*, 19–26.

Sangren, E., Jensen, E., & Welton, J. (1990). Topological design of structural components using genetic optimization methods. *Sensitivity Analysis Optimization with Numerical Meth., 115*, 31–43.

Schaefer, C. (1987). The ARGOT strategy: Adaptive representation genetic optimized technique. *Proc. Second Int. Conf. Genetic Algorithms and their Applications*, (Grefenstette, J., Ed.), pp. 50–58. Lawrence Erlbaum Assoc., Cambridge, Massachusetts.

Schraudolph, N., & Belew, R. (1991). *Dynamic parameter encoding for genetic algorithms*. Technical Report No. LAUR 90-2795 (revised). Los Alamos National Laboratories, California.

Soto, C., & Diaz, A. (1993). Optimum layout and shape of plate structures using homogenization. *Topol. Design Struct.*, 407–420.

Suzuki, K., & Kikuchi, N. (1990). Shape and topology optimization by a homogenization method. *Sensitivity Analysis and Optimization with Numerical Meth., 115*, 15–30.

Suzuki, K., & Kikuchi, N. (1991). A homogenization method for shape and topology optimization. *Computational Methods App. Mechanics Eng., 93*, 291–318.

https://doi.org/10.1017/S0890060499135066 Published online by Cambridge University Press

Todoroki, A., Watanabe K., & Kobayashi, H. (1995). Application of genetic algorithms to stiffness optimization of laminated composite plates with stress-concentrated open holes. *JSME Int. J., 38(4)*, 458–464.
Wolfersdorf, J., Achermann, E., & Weigand, B. (1997). Shape optimization of cooling channels using genetic algorithms. *J. Heat Transfer 119*, 380–388.

David J. Eby is currently working towards the Ph.D. at Michigan State University in the Department of Materials Science and Mechanics. Research interests include optimization and computational mechanics. He specializes in nonlinear mechanics and application of genetic algorithms to structural optimization.

Ronald C. Averill is an Associate Professor in the Department of Materials Science and Mechanics at Michigan State University. His field of specialization is Computational Mechanics, especially in the areas of composite materials and structures, damage analysis, crash analysis, and design optimization. Dr. Averill received his Ph.D. in Engineering Mechanics in 1992 from Virginia Polytechnic Institute and State University, where he performed research in the progressive failure of laminated composite structures and micromechanics of composite materials. Dr. Averill has published over 35 technical papers in the areas of structural analysis and design, and has received numerous awards for his research efforts.

William F. Punch III is an Associate Professor of Computer Science and Engineering at Michigan State University. He is co-director of the Genetic Algorithms Research and Applications Group ("GARAGe"). His main interests are genetic algorithms and genetic programming, including theoretical issues (parallel GA/GP) and application issues (design, layout, scheduling, etc.). He also conducts research on web search, focused on intelligent search approaches based on pattern-recognition techniques and GA/GP search.

Erik D. Goodman is Professor of Mechanical Engineering, of Electrical and Computer Engineering, and of Materials Science and Mechanics, at Michigan State University. He directs the A.H. Case Center for Computer-Aided Engineering and Manufacturing, co-directs the Genetic Algorithms Research and Applications Group ("GARAGe"), and directs MSU's Manufacturing Research Consortium. He received the Ph.D. in Computer and Communication Sciences from the University of Michigan in 1972, where he began his research in genetic algorithms. His particular interest is in the development of parallel genetic algorithms for solution of problems in engineering design and manufacturing.