Ferdowsi University of Mashhad

Master of Science Thesis in
Electrical Engineering – Control Program

Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization

By:

Mohsen Davarynejad

June 2007

This document provides a full abstract of my M.Sc thesis.
The original document is in Farsi.

TABLE OF CONTENTS

	Page
1. Introduction	2
2. Adaptive Fuzzy Fitness Granulation (AFFG)	6
3. Benchmark Problems and Numerical Results	11
3.1 Traditional Optimization problems	11
3.2. Structural Design Optimization Problems	14
3-layer Composite Beam	15
Airplane Wing	15
2-D TrussFrame	16
Voltage and Pattern Design of Piezoelectric Actuator	19
Piezoelectric design for static shape control	20
Analysis of Results	23
4. Conclusion	25
References	26

Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization

Abstract - Nature may have been the original inspiration for evolutionary algorithms, but unlike artificially designed systems, nature has an abundance of resources and time. For man-made systems, computational complexity is a prohibitive factor in sufficiently large and complex problems of today. Much of this computational complexity is due to the fitness function evaluation that may either not exist or be computationally very expensive. But, an exact computation of fitness may not be really necessary as long as a proper rank is approximately preserved in the evolution's scheme of the survival of the fittest. Here, we aim to exploit this feature of evolution and to investigate the use of fitness granulation via an adaptive fuzzy similarity analysis in order to reduce the number of fitness evaluations. In the proposed algorithm in this thesis, an individual's fitness is only computed if it has insufficient similarity to a pool of fuzzy granules whose fitness has already been computed. If an individual is sufficiently similar to a known fuzzy granule, then that granule's fitness is used instead as a crude estimate. Otherwise, that individual is added to the pool as the core of a new fuzzy granule. Each granule's radius of influence is adaptive and will grow/shrink depending on the population fitness. The proposed technique is applied to two sets of problems. First is a set of several numerical benchmark problems with various optimization characteristics. Second is a set of four hardware design problems that are evaluated via finite element analysis. Performance of the proposed algorithm is compared with several other competing algorithms, i.e. a Fast Evolutionary Strategy (FES), a GA-NN, as well as a simple GA, in terms of both computational efficiency and accuracy. Statistical analysis reveals that the proposed method significantly decreases the number of fitness function evaluations while finding equally good or better solutions. Moreover, application to the hardware design problems reveals better structural designs more consistently with better computational efficiency.

Keywords: Evolutionary Computation, Fuzzy Granulation, Fitness Function Evaluations (FFE), Fitness Approximation (FA), Shape Optimization, Evolutionary Design.

1. INTRODUCTION

As the field of evolution-based algorithms matures and tackles more real-world applications, its limitations and challenges also become clearer. While nature is indeed the original inspiration as well as a presumably successful example of evolutionary algorithms, it is clear that natural and artificial evolutions are not really at par, at least not yet. This is because nature has an abundance of resources and time while man made systems are severely limited in both.

Fitness function evaluation is often the most prohibitive and limiting segment of artificial evolutionary algorithms, because an explicit fitness function may either be non-existent or its computation may be prohibitively costly. In both cases, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. In other words, an *approximated model* also referred to as *surrogate* or *meta model,* can be used as a substitute for the computationally more expensive but also more exact model. These approximation models greatly reduce the computational expense since the efforts involved in building/training a surrogate model is much lower than the more exact simulation models/codes.

In design of mechanical structures, for instance, each exact fitness evaluation requires the time consuming stage of finite element analysis (FEA) which, depending on the size of the problem, may require anywhere from several seconds to several days. In a conventional genetic algorithm with a fixed and modest population size of 100, 100 generations, and a very small scale structural problem that requires 10 seconds for each instance of fitness evaluation, this means about thirty hours of computing. One can see the inhibiting role of computational complexity for more non-trivial and large-scale problems.

Various methods in the literature have addressed this problem. In [1] and [2] a method of function approximation is proposed based on fitness inheritance. Sastry, et. al. [3] analyzed convergence time and population sizing of evolutionary algorithms with fitness inheritance. A similar approach has also been suggested in [4, 31] namely “Fast Evolutionary Strategy” (FES) where fitness of a child is the weighted sum of its parents. In that approach, a fitness and associated reliability value are assigned to each new individual that is only evaluated using the true fitness function if the reliability value is below a certain threshold. Additionally, Coello and his colleagues incorporated the concept of fitness inheritance into a multi-objective particle swarm optimizer to reduce the number of fitness evaluations [38]. In [39], they tested their approach on a set of well-known test suite of multi-objective optimization problems. They generally reported lower computation cost, while the quality of their results improved in higher dimensional spaces. However, as also shown in [5] as well as in this paper, the performance of parents may not be a good predictor of their children for sufficiently complex and multiobjective problems rendering fitness inheritance inappropriate under such circumstances.

The problem of fitness estimate also appears in sufficiently complex applications where it may be desirable to decompose a problem into several smaller/simpler problems that are more easily solvable such as in cooperative co-evolutionary schemes. But the new problem becomes estimating the fitness of these smaller problems from evaluation of the original problem at large. Individuals in these sub-populations encode only part of the problem and their fitness value always depends on others. To solve this problem, methods such as fitness assignment for estimating fitness values [10] and fitness estimation by association/friendship [28] have been developed.

Other common approaches are based on learning and interpolating from known fitness values of a small population. Specifically, widely used methods in design engineering include the response surface methodology that uses low-order polynomials and the least square estimations [11], as well as the Kriging model that is also called the Design and Analysis of Computer Experiments (DACE) model [12]. In Kriging model, a global polynomial approximation is combined with a local gaussian process, and the maximum likelihood method is used for parameter estimation. Zhou, et. al. in [33] presented a combination of global and local meta-models for solving computationally expensive problems. They showed that multiple meta-models can be combined to accelerate EA search. In [34] a class of new meta-models which utilize both known responses and response gradients for their training is proposed. The new gradient-assisted meta-models are extensions of standard multi-layer perceptrons and radial basis function networks.

While the above methods aim to improve the performance of evolutionary optimization approaches by approximating the fitness of each individual, other general methods employ other soft computing strategies. For instance in [35] and [36], fuzzy logic tools are used to adapt parameters that control the application of genetic algorithms. In these approaches, fuzzy rule bases are implicitly learned by means of an additional genetic algorithm that coevolves with the main one. The goal is to obtain control parameter values that offer more adaptation to the genetic operator to show an adequate performance.

In the last few years, artificial neural networks (ANN), including multi-layer perceptrons [13] and radial basis function networks [14] have also been employed to build approximate models for design optimization. Due to universal approximation property of ANN, ANN can be good estimators of fitness function if provided with sufficient structural complexity and richness of training data set [16]. As with any other numerically driven approximation method, the performance of the neural network is closely related to the quality of the training data.

Lack of sufficient training data is the main problem in using fitness approximation models and hence the failure to reach a model with sufficient approximation accuracy. Since evaluation of the original fitness function is very time-consuming and/or expensive, the approximate model may be of low fidelity and may even introduce false optima. Furthermore, if the training data does not cover all the domain range, large errors may occur due to extrapolation. Errors may also occur when the set of training points is not sufficiently dense and uniform. In such situations, a combination of methods may be more desirable. For example, Ong et al. [17] combined radial basis functions with transductive inference to generate local surrogate models. Gaussian Processing [18] is a statistical modeling technique which is also used for fitness function approximation. A comparison of neural networks and kriging for fitness approximation in evolutionary optimization can be found in [19]. Fitness approximation by Support Vector Regression (SVR) is introduced in [20] as well as applied in [29].

Alternatively, if individuals in a population can be clustered into several groups as in [6], then only the individual that represents its cluster can be evaluated. The fitness value of other individuals in the same cluster will be estimated from the representative individual based on a distance measure. This is termed fitness imitation in contrast to fitness inheritance in [7]. The idea of fitness imitation has been extended and more sophisticated estimation methods have been developed in [8] and [9].

While the above methods aim to reduce computational cost by approximating the fitness function, constructing a globally correct approximate model remains to be difficult because of the high dimensionality and limited number of training samples. Evolutionary algorithms using such approximate fitness functions may then converge to false optima. Therefore, it may be beneficial to selectively use the original fitness function together with the approximate model [21]. In conventional optimization, this is commonly known as model management [21] or evolution control in evolutionary computation [22]. For example, Khorsand and Akbarzadeh [15] recently investigated structural design by a hybrid of neural network and finite element analysis that only selectively used the neuro-estimation when either interpolation was expected (interpolation is generally expected to be more accurate) or the individual was not deemed to be highly fit (error in estimation may not be important). However, the prevalent problems with interpolation in rough surfaces remain. The assumption of smooth continuity may not be valid, and interpolation may hence yield values that are not even physically realizable. Furthermore, we may be blinded to the optimal solutions using interpolation as interpolation assumes a pattern of behavior that may not be valid around optimal peaks.

Fuzzy granulation of information is a vehicle for handling *information*, as well as a *lack* of it (uncertainty), at the level of coarseness that can still solve problems appropriately and efficiently. Zadeh proposed fuzzy information granulation in 1979 [23] as a technique by which a class of points (objects) are partitioned into granules, *with a granule being a clump of objects drawn together by indistinguishability*, *similarity*, or *functionality*. The fuzziness of granules and their attributes is characteristic of the ways by which human concept and reasoning is formed, organized and manipulated. The concept of a granule is more general than that of a cluster, potentially giving rise to various conceptual structures in various fields of science as well as mathematics.

In this thesis, the concept of fitness granulation is applied to exploit the natural tolerance of evolutionary algorithms in fitness function computations. Nature's "survival of the fittest" is not about exact measures of fitness; rather it is about rankings among competing peers [32]. By exploiting this natural tolerance for imprecision, optimization performance can be preserved by computing fitness only selectively and only to preserve this ranking among individuals in a given population. Also, fitness is not interpolated or estimated; rather, the similarity and indistinguishability among real solutions is exploited.

In the proposed algorithm, an adaptive pool of solutions (fuzzy granules) with an exactly computed fitness function is maintained. If a new individual is sufficiently similar to a known fuzzy granule [24], then that granule's fitness is used instead as a crude estimate. Otherwise, that individual is added to the pool as a new fuzzy granule. In this fashion, regardless of the competition's outcome, fitness of the new individual is always a physically realizable one, even if it is a "crude" estimate and not an exact measurement. The pool size as well as each granule's radius of influence is adaptive and will grow/shrink depending on the utility of each granule and the overall population fitness. To encourage fewer function evaluations, each granule's radius of influence is initially large and is gradually shrunk in latter stages of evolution. This encourages more exact fitness evaluations when competition is fierce among more similar and converging solutions. Furthermore, to prevent the pool from growing too large, once the pool reaches a certain maturity, granules that are not used are gradually replaced by new granules.

This summery of thesis is organized as follows. The proposed method of generating fuzzy granules is explained in Section 2 via an adaptive fuzzy similarity analysis for granule generation. Thus, in Section 3, two groups of optimization problems are investigated and simulated. The first group is a set of six conventional optimization benchmark problems [30] of various characteristics. The second group is four structural design problems [37], i.e. determination of six and two design parameters in an Airplane Wing and a 3-layer Composite Beam respectively to increase the wing's and beam's first natural frequency, design of static shape control of a 2-D truss frame structure and a Piezoelectric bimorph beam, each having 36 and 200 optimization parameters, respectively. Statistical analysis confirms that the proposed approach reduces the computational complexity of the design problems by over 50% while reaching similar or better fitness levels. It should be mentioned that the present approach does not require any initial training.

2. ADAPTIVE FUZZY FITNESS GRANULATION (AFFG)

The proposed fuzzy adaptive fitness granulation aims to minimize the number of exact fitness function evaluations by creating a pool of solutions (fuzzy granules) by which an approximate solution may be sufficiently applied to proceed with the evolution. If a human designer could be in the middle of an evolutionary cycle, trying to selectively minimize the number of fitness evaluations, the human designer would group and cluster rather than interpolate. In other words, if a given design is sufficiently similar to an existing design that is poor, it is discarded; and if it is similar to one that is good, it is kept. So, the question for the designer would be when to assign a new individual to an existing cluster and when to create a new cluster. With this approach, every cluster is assigned the fitness value of a representative individual. The designer would then know that there exists at least one physically realizable solution for that cluster.

Similarly, the proposed algorithm uses fuzzy similarity analysis to produce and update an adaptive competitive pool of dissimilar solutions/granules. When a new solution is introduced to this pool, granules compete by a measure of similarity to bond with the new solution and thereby to prolong their lives in the pool. In turn, the new individual simply assumes fitness of the *most similar* (winning) individual in this pool. If none of the granules are sufficiently similar to the new individual, i.e. their similarity is below a certain threshold, the new individual is instead added to the pool after its fitness is evaluated exactly by the known fitness function. Finally, granules that cannot bond with new individuals are gradually eliminated in order to avoid a continuously enlarging pool. The proposed algorithm is shown in Figure 1 and is discussed in detail below.

As is shown in Figure 1, a random parent population $P_0 = \{X_1^1, X_2^1, \ldots, X_j^1, \ldots, X_t^1\}$ is initially created, where $X_j^i = \{x_{j,1}^i, x_{j,2}^i, \ldots, x_{j,r}^i, \ldots, x_{j,m}^i\}$ is *j-th* individual in *i-th* generation, $x_{j,r}^i$ is the *r-th* parameter of X_j^i, t is population size, and m is the number of design variables. Also, $G = \{(C_k, \sigma_k, L_k) \mid C_k \in \Re^m, \sigma_k \in \Re, L_k \in \Re, k = 1,\ldots,l\}$ is a set of fuzzy granules that is initially empty, i.e. $l = 0$, where C_k is an *m*-dimensional vector of centers, σ_k is the width of membership functions of the *k*-th fuzzy granule, and L_k is the granule's life index. The phenotype of first chromosome i.e. $X_1^1 = \{x_{1,1}^1, x_{1,2}^1, \ldots, x_{1,r}^1, \ldots, x_{1,m}^1\}$ is chosen as the center $C_1 = \{c_{1,1}, c_{1,2}, \ldots, c_{1,r}, \ldots, c_{1,m}\} = X_1^1$ of first granule. The membership function $\mu_{r,k}$ therefore describes a Gaussian similarity neighborhood for each parameter *k* as follows,

$$\mu_{k,r}(x_{j,r}^i) = \exp(-(x_{j,r}^i - c_{k,r})^2 / (\sigma_{k,r})^2) \tag{1}$$

for $k = 1,\ 2,\ \ldots,\ l$ where l is the number of fuzzy granules.

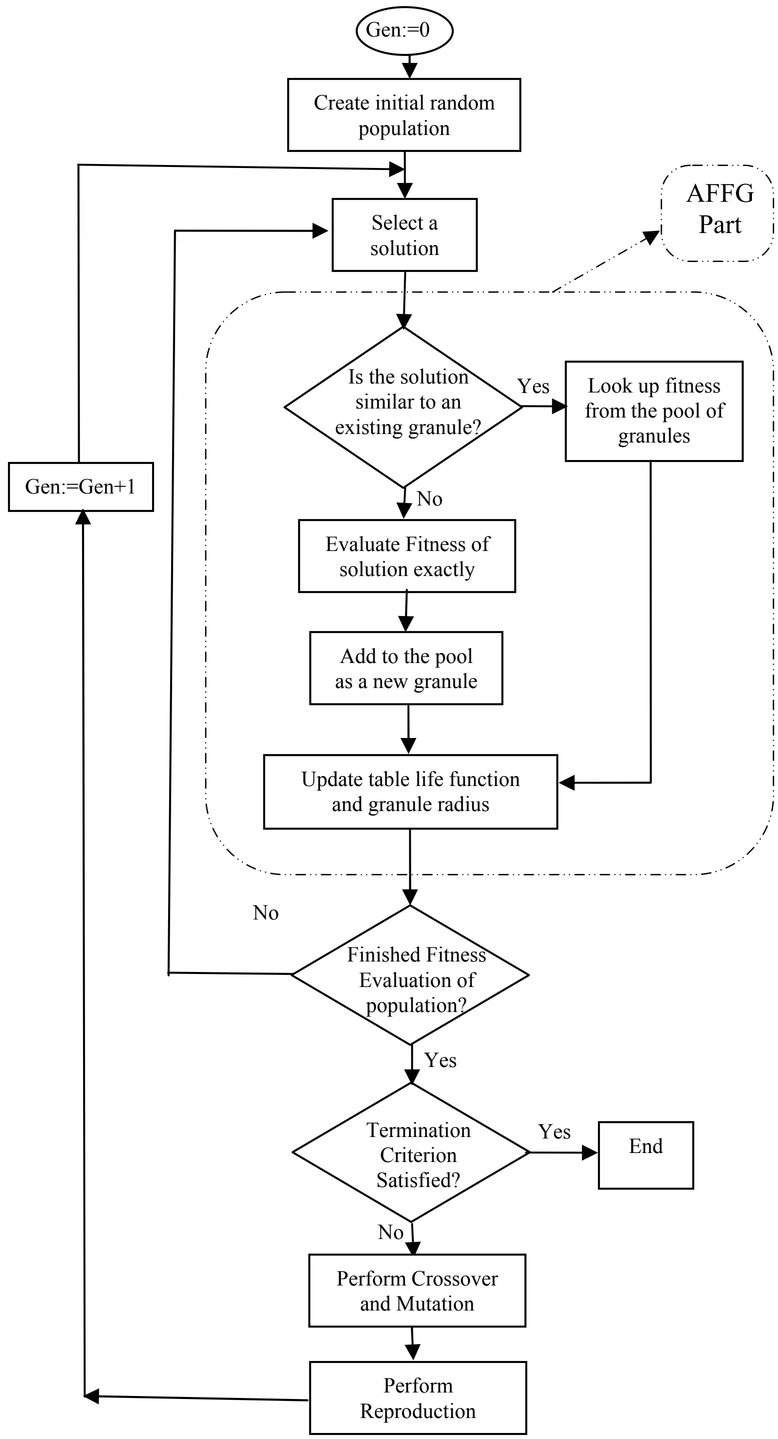

Figure 1. Flowchart of the Proposed AFFG Algorithm

Then, the average similarity of a new solution $X_j^i = \{x_{j,1}^i, x_{j,2}^i, \ldots, x_{j,r}^i, \ldots, x_{j,m}^i\}$ to each granule G_k can be computed by $\overline{\mu}_{j,k} = \sum_{r=1}^{m} \frac{\mu_{k,r}(x_{j,r}^i)}{m}$. Fitness of X_j^i is either calculated by computing the exact fitness function or estimated by associating it to one of the granules in the pool if there is a granule in the pool with higher similarity to X_j^i than a predefined threshold, as follows.

$$f\left(X_j^i\right) = \begin{cases} f(C_K) & \text{if} \quad \max\limits_{k \in \{1,\ 2,\ \ldots,\ l\}} \{\overline{\mu}_{j,k}\} > \theta^i \\ f\left(X_j^i\right) \ \text{computed by fitness function} & otherwise \end{cases}$$

where $K = \underset{k \in \{1,\ 2,\ \ldots,\ l\}}{index \max} \{\overline{\mu}_{j,k}\}$, $\theta^i = \alpha . \frac{Max\{f(X_1^{i-1}),\ f(X_2^{i-1}), \ldots f(X_t^{i-1})\}}{\bar{f}^{\,i-1}}$, $\bar{f}^{\,i} = \sum_{j=1}^{t} \frac{f\left(X_j^i\right)}{t}$, and $\alpha > 0$ is a constant of proportionality. Threshold θ^i increases as the best individual's fitness in generation i increases. Hence as the population matures and reaches higher fitness valuations while also converging more, the algorithm becomes more selective and uses exact fitness calculations more often. Therefore, with this technique we can utilize the previous computational efforts during previous generations. Alternatively, if $\max\limits_{k \in \{1,2,\ldots,l\}} \{\overline{\mu}_{j,k}\} < \theta^i$, X_j^i is chosen as a newly created granule.

σ_k is distance measurement parameter that controls the degree of similarity between two individuals. Since it is more important to have accurate estimation of the fitness function of the individuals that are highly fit, the granules shrink or enlarge in reverse proportion to their fitness as below.

$$\sigma_k = \gamma \frac{1}{\left(e^{F(C_k)}\right)^{\beta}} \tag{2}$$

where $\beta > 0$ is an emphasis operator, and γ is a constant of proportionality that is usually set at 1 unless otherwise indicated. The combined effect of granule enlargement/shrinkage in accordance to the granule fitness and the threshold increase in proportion to each population's fitness is that the algorithm initially accepts individuals with less similarity as similar individuals. In other words, since members of the initial populations generally have less fitness, σ_k larger and θ^i is smaller, fitness is assumed more often initially by estimation/association to the granules. As the evolution proceeds, fitness in both the pool of granules as well as current population is expected to increase. This prompts higher selectivity for granule associability and higher threshold for estimation. In other words, in later generations, the degree of similarity between two individuals must be larger than the earlier generations to be accepted as similar individuals. Equation (2) adapts the width of membership functions in order to have more exact fitness computed around individuals who perform very well, but fewer fitness computations around individuals who have poor performance. This procedure promotes both fast convergence rate as well as high accuracy because of lower computation cost in the early steps of evolution and accurate estimation of fitness function during later generations.

Finally, as the evolutionary algorithm proceeds, it is inevitable that new granules are increasingly generated and added to the pool. Depending on complexity of the problem, the size of this pool can become excessive and become a computational burden itself. To prevent such unnecessary computational effort, granules compete for their survival through a life index. In other words, it is better to remove granules that do not bond with new individuals, thereby producing a bias against individuals that have low fitness and were likely produced by a failed mutation attempt. Hence, L_k is initially set at 1 and subsequently updated as below,

$$L_k = \begin{cases} L_k + M & if \quad k = K \\ L_k & Otherwise \end{cases} \tag{3}$$

where M is the life reward of the granule and K is the index of the winning granule for each individual in generation i. At each table update, only N_G granules with highest L_k index are kept, and others are discarded. The following example is provided to illustrate the competitive granule pool update.

Example: Suppose there are three granules in the pool with four variables in the *i*-th generation (Table 1.a), and two new upcoming individuals for fitness estimation. Similarity threshold is computed as $\theta^i = 0.8$ from previous generation. Table 1.b shows the two individuals in the *i*-th generation. Tables 1.c and 1.d illustrate similarities between individuals in the current population and the granules. Since $\theta^i < \max_{\substack{k \in \{1,\ 2,\ \ldots,\ l\} \\ j=1}} \{\overline{\mu}_{j,k}\}$, the 1st individual is similar to second granule and so $F(X_1^i)$ can be approximated as $\hat{F}(X_1^i) = F(X_2)$. But $\theta^i > \max_{\substack{k \in \{1,\ 2,\ \ldots,\ l\} \\ j=2}} \{\overline{\mu}_{j,k}\}$ for the second individual is not similar to any of the existing granules and is added as a new granule to the pool. Finally, the first granule is deleted from the pool (stack of granules) as shown by the updated granule pool in Table 1.e.

Table 1. (a) The pool of granules and two new individuals in generation i, $\alpha = 0.9$, $\beta = 0.1$, $M = 5$, $N_G = 3$.

	$c_{k,1}$	$c_{k,2}$	$c_{k,3}$	$c_{k,4}$	$f(G_k)$	σ_k	L_k
G_1, 1st granule	1	1	1	1	6	0.5488	1
G_2, 2nd granule	1	2	2	1	12	0.3012	2
G_3, 3rd granule	2	1	1	2	18	0.1653	4

(b) Members of Population i, Population size = 2.

	$x^i_{j,1}$	$x^i_{j,2}$	$x^i_{j,3}$	$x^i_{j,4}$
X^i_1	1.1	1.9	1.9	1.1
X^i_2	2	2	2	2

(c) Degrees of similarity for first individual in i^{th} population.

	$\mu_{k,1}(x^i_{1,1})$	$\mu_{k,2}(x^i_{1,2})$	$\mu_{k,3}(x^i_{1,3})$	$\mu_{k,4}(x^i_{1,4})$	$\overline{\mu}_{1,k}$	$f(X^i_1)$
X^i_1 and G_1	0.9835	0.2606	0.2606	0.9835	0.6220	6
X^i_1 and G_2	0.9464	0.9464	0.9464	0.9464	0.9464	12
X^i_1 and G_3	$3.654*10^{-7}$	$3.654*10^{-7}$	$3.654*10^{-7}$	$3.654*10^{-7}$	$3.654*10^{-7}$	18

(d) Degrees of similarity for second individual in i-th population.

	$\mu_{k,1}(x^i_{2,1})$	$\mu_{k,2}(x^i_{2,2})$	$\mu_{k,3}(x^i_{2,3})$	$\mu_{k,4}(x^i_{2,4})$	$\overline{\mu}_{2,k}$	$f(X^i_2)$
X^i_2 and G_1	0.1901	0.1901	0.1901	0.1901	0.1901	6
X^i_2 and G_2	0.0040	1	1	0.0040	0.5020	12
X^i_2 and G_3	1	$1.129*10^{-8}$	$1.129*10^{-8}$	1	0.5000	18

(e) Updated pool of granules

	$c_{k,1}$	$c_{k,2}$	$c_{k,3}$	$c_{k,4}$	$f(G_k)$	σ_k	L_k
G_1	2	2	2	2	15	0.2231	5
G_2	1	2	2	1	12	0.3012	7
G_3	2	1	1	2	18	0.1653	4

3. BENCHMARK PROBLEMS AND NUMERICAL RESULTS

To illustrate the efficacy of the proposed granulation techniques, two classes of optimization problems are studied in the following two sections. First is a set of six traditional optimization benchmarks that are chosen for their various characteristics. Second is a set of four mechanical design problems that typically require finite element analysis for their fitness evaluation. This second set of problems also has a higher number of parameters, hence a more challenging optimization task from a fitness/computational perspective.

Furthermore, due to the stochastic nature of evolutionary optimization, each of the below simulations are repeated several times, and a paired t-test of significance is performed. The significance level α represents the maximum tolerable risk of incorrectly rejecting the null hypothesis H_0, indicating that population 1's mean is not significantly different from population 2's mean. The p-value or the observed significance level of a statistical test is the smallest value of α for which H_0 can be rejected. If the p-value is less than the pre-assigned significance level α, then the null hypothesis is rejected. Here, the significance level α was assigned, and the p-value was calculated for each of the following applications.

3.1 Traditional Optimization problems

De Jong [25] proposed that a suitable test environment must address the following characteristics: continuous vs. discontinuous, convex vs. non-convex, unimodal vs. multimodal, quadratic vs. non-quadratic, low-dimensionality vs. high-dimensionality and deterministic vs. stochastic. The selected test bed functions are listed in Table 2 along with their various distinguishing characteristics.

The GA routine utilizes random initial populations, binary-coded chromosomes, single-point crossover, mutation, fitness scaling, and an elitist stochastic universal sampling selection strategy. In these simulations, the crossover rate $P_{XOVER} = 1$, $P_{MUTATION} = 0.01$, population size is 20, and generation size is 100. Finally chromosome length varies depending on the number of variables in a given problem but each variable's length is 8 bits. The total number of generations as well as termination criterion is determined during several trial runs to ensure the convergence of the algorithm on all six benchmark problems. GA-AFFG uses all of the above evolutionary parameters as in GA to establish analysis only from the perspective of granulation.

Comparison results are illustrated in Table 3. The GA, FES, GA-NN and GA-AFFG are each run 15 times for each of the above 6 functions. For FES, a fitness and associated reliability value are assigned to each new individual that is truly evaluated if the reliability value is below a certain threshold T. The reliability value varies between 0 and 1 and depends on two factors: first is the reliability of parents, and second is how close parents and children are in the solution space. Also, as mentioned in [4], T = 0.7 is used for the threshold as it generally produces the best results. Also, the GA-NN general diagram is shown in Figure 2. The parameters of GA-NN are same as in GA. In GA-NN approach, a two layer neural network with 100 neurons in hidden layers and one output architecture is chosen and used for all optimization benchmarks except the first one that has 10 neurons in its hidden layer. Further details of GA-NN algorithm can be found in authors' earlier work [15] and is not included here for brevity.

Table 2. Proposed test bed functions for testing the performance of AFFG.

Function	Formulation and limits	Characteristics
De Jong's 1 F_1	$\sum_{i=1}^{m} x_i^2$, $i = 1:m, m = 3;$ $, -5.12 \le x_i \le 5.12;$	The simplest test function is De Jong's function 1. It is continuous, convex and unimodal
F_2	$\sum_{i=1}^{m} x_i^4 + Gauss(0,1)$, $i = 1:m, m = 30;$ $-1.28 \le x_i \le 1.28;$	It is continuous, unimodal but noisy.
Michalewicz F_3	$-\sum_{i=1}^{m} (\sin(x_i) \cdot (\sin(\frac{i \cdot x_i^2}{\pi}))^{(2 \cdot n)}),$ $i = 1:m;$ $, m = 10, n = 10;$ $, 0 \le x_i \le \pi;$	The Michalewicz function is a multimodal test function (m! local optima). The parameter n defines the "steepness" of the valleys or edges. Larger n leads to more difficult search. For very large n the function behaves like a needle in the haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum).
Rastrigins F_4	$10 \cdot m + \sum_{i=1}^{n} (x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i))$, $i = 1:m;$ $, m = 20;$ $, -5.12 \le x_i \le 5.12;$	Rastrigin's function is based on function 1 with the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the locations of the minima are regularly distributed.
Schwefel's F_5	$\sum_{i=1}^{m} (-x_i \cdot \sin(\sqrt{\lvert x_i \rvert}))$, $i = 1:m, m = 20;$ $, -500 \le x_i \le 500;$	Schwefel's function is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction.
Griewangk's F_6	$1 + \sum_{i=1}^{m} \frac{x_i^2}{4000} - \prod_{i=1}^{m} \cos(\frac{x_i}{\sqrt{i}})$, $i = 1:m, m = 20;$ $, -600 \le x_i \le 600;$	Griewangk's function is similar to Rastrigin's function. It has many widespread local minima. However, the locations of the minima are regularly distributed.

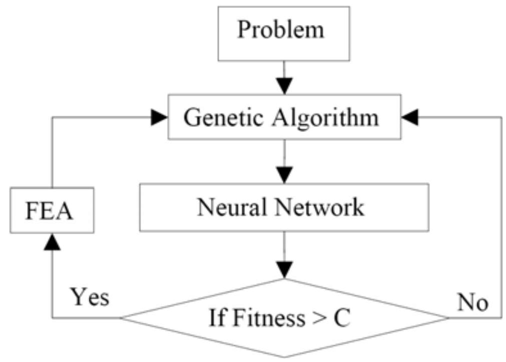

Figure 2. GA-NN Algorithm

Table 3. Comparison of GA, Fast Evolutionary Strategy (FES) & GA-AFFG. Opt means average of optimum solution in fifteen independent runs. GA is used as the benchmark for comparison in the t-tests. If p-value ≥ α, then there is not a statistically significant difference between the two methods.

$\alpha = 0.9$, $M = 5$, $N_G = 200$, $T = 0.7$

Function	β	γ	Opt. Method	GA	FES	GA-NN	GA-AFFG
F1	.1	1	No. FFE	2000	941.93	198.4	234.3333
			OPT (max)	78.5888	78.465	78.433	78.4796
			p -value	-	0.2854	0.229	0.2996
F2	0.04	1.7	No. FFE	2000	1220.67	631.258	735.6
			OPT (max)	72.6658	71.965	71.189	72.1092
			p -value	-	0.190	0.102	0.2505
F3	0.4	1.85	No. FFE	2000	898.026	730.178	609.733
			OPT (max)	7.856	7.697	7.692	7.7181
			p -value	-	0.237	0.0567	0.1734
F4	.004	0.15	No. FFE	2000	2000	711.269	842.26
			OPT (min)	62.5674	68.2049	69.112	65.761
			p -value	-	0.1733	0.1200	0.4568
F5	0.0008	300	No. FFE	2000	1472.467	999.6	945.333
			OPT (max)	6210.2466	5966.226	5885.121	6031.670
			p -value	-	0.0764	0.0628	0.2610
F6	0.00012	190	No. FFE	2000	1127.446	899.1333	759.667
			OPT (max)	1643.7785	1618.394	1571.322	1631.158
			p -value	-	0.0929	0.1050	0.3539

While the (two) evolutionary schemes (GA and GA-AFFG) reach statistically similar performance in terms of optimal fitness, the proposed technique reduces the number of function evaluations by over 50% for all benchmark problems. Moreover the fitness inheritance approach has comparable performance when size of dimension is small, but its performance deteriorates as problem complexity increases.

Table 4 represents the mean and variance of number of exact fitness function evaluations of the above six numerical optimization problems. A paired t-test of significance is also performed to study the significance of lower computation cost.

Table 4: Mean and Var represent the mean and variance of the number of fitness evaluations for fifteen independent runs. A paired t-test of significance is also performed.

Function	GA	FES			GA-NN			GA-AFFG		
	Mean	Mean	Var	p -value	Mean	Var	p -value	Mean	Var	p -value
F1	2000	941.93	37738	3×10^{-12}	198.4	2371	3.8×10^{-29}	234.3333	3091	7.2×10^{-22}
F2	2000	1220.667	93301	5.7×10^{-7}	631.2584	24952	2.5×10^{-14}	735.6	14498	1.9×10^{-15}
F3	2000	898	58059	5.5×10^{-11}	730.1	38384	4.8×10^{-13}	609.7333	28140	1.6×10^{-14}
F4	2000	2000	0	NA*	711.2	970	2.9×10^{-24}	842	13686	1.4×10^{-15}
F5	2000	1472	30391	1×10^{-6}	999.6	20448	1.6×10^{-13}	945.3333	4431.23	2×10^{-18}
F6	2000	1127.4	54434	8×10^{-10}	899.13	3841.4	4×10^{-19}	759	17273	1.7×10^{-15}

* Not Available

3.2. Structural Design Optimization Problems

Structural optimization can be a good application of AFFG since fitness evaluation by conventional finite element analysis is computationally costly. Such algorithms may require several days to complete for even trivial problems. Four structural design problems, with ascending order of number of optimization variables, are investigate here, namely: design of a 3-layer composite beam (two optimization variables), an airplane wing (6 variables), a 2-D Truss frame (36 variables) to increase their rigidity, .i.e. raising their first natural frequency, as well as voltage/pattern design of Piezoelectric actuators (200 variables).

The GA routines utilize random initial populations, binary-coded chromosomes, single-point crossover for the first three problems and 15-point crossover for the piezoelectric actuator design problem, mutation, fitness scaling, and an elitist stochastic universal sampling selection strategy. Similar to above, crossover rate, $P_{XOVER} = 1$, $P_{MUTATION} = 0.01$, population size is 20. However, due to the number of parameters and complexity of the structural problems, number of generations is set at 50 and 600 for the first three problems and the piezoelectric actuator design problems, respectively. These settings were determined during several trial runs to reflect the best performing set of parameters for GA. Finally chromosome length varies depending on the number of variables in a given problem but each variable is still allocated 8 bits. Also, as in earlier, FES is used for comparison with threshold values T = 0.7 for all simulations as well as GA-NN with a two layer architecture of neural network.

3-layer Composite Beam

A 3-layer composite beam has been considered to illustrate the efficiency of the proposed methodology in material optimization problems. In this example, the Young's modulus is EX=210 GPa, EY=25 GPa, EZ=25 GPa, GXY=GYZ=GXZ= 30 GPa, Poisson's ratio v = 0.2 and density $\rho = 2100$ kg/m^3. Composite lay up are design variables that change in the region [0-180]. The objective here is to raise first natural frequency by appropriately choosing 2 composite layers' angles. A 2-100-1 NN architecture is consequently chosen and used for the optimization runs. The proposed algorithm AFFG and other methods are compared in Table 6. Results indicate that while there is not a significant statistical difference between the three algorithms in terms of solution fitness, i.e. rigidity of the beam, the improved time of the proposed method is much higher than GA-NN. In particular, the proposed AFFG algorithm finds better solutions on the average with less computational time as compared with the GA-NN. Also, while FES seems to have found better solutions, the proposed GA-AFFG has used fewer than half as many evaluations.

Airplane Wing

An airplane wing has been considered to illustrate the efficiency of the proposed methodology in B-spline boundary shape optimization. The wing is of uniform configuration along its length, and its cross-sectional area is defined to be a straight line and a spline, as shown in Figure 3. It is held fixed to the body on one end and hangs freely at the other.

Material properties are: Young's modulus = 261.820 GPa, density $\rho = 11031$ kg/m^3, Poisson's ratio v = 0.3. The objective here is also to maximize the wing's first natural frequency by appropriately choosing three key points of the spline, as shown in Figure 3a. A 6-100-1 architecture is chosen for NN as fitness approximator.

The optimized shape by simple GA is shown in Figure 3b and by GA-AFFG is shown in Figure 3c. Table 7 illustrates that while the GA-NN finds inferior solutions as compared with GA, usage of NN significantly reduces computational time. Application of AFFG shows the proposed algorithm improves search quality while remaining computationally less intensive. Specifically, the average 10-run performance of AFFG solutions is higher than all competing algorithms including GA, FES and GA-NN. Furthermore, while the t-test confirms that the propose algorithm solutions are at least as good as those produced by GA, the proposed algorithm is over %82 faster.

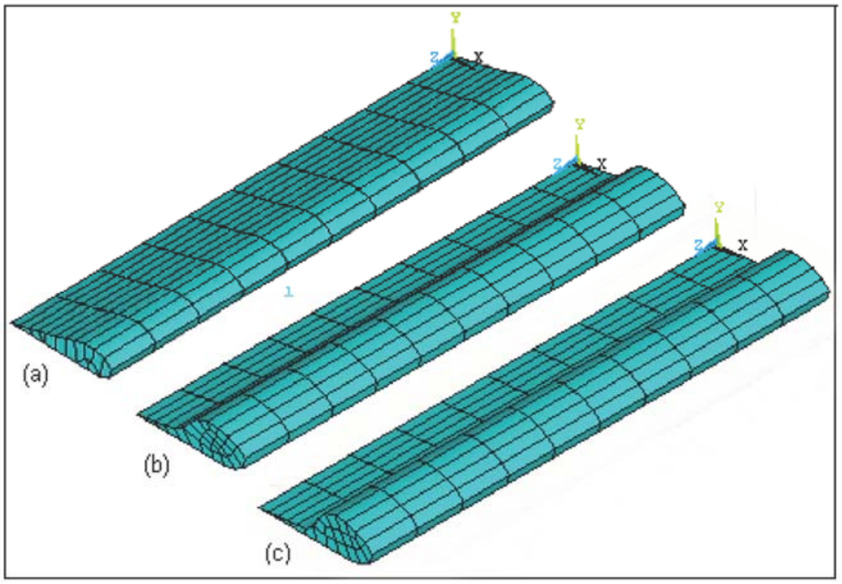

Figure 3. Airplane Wing: (a) Initial shape, (b) GA optimized shape, (c) GA-AFFG

2-D Truss Frame

A typical truss designed by an engineer is depicted in Figure 4(a). In this benchmark, isotropic material properties are assumed (Young's modulus E =210 GPa, Poisson's ratio v = 0.3 and density ρ = 7800 kg/m^3). The optimized shapes by GA and the new proposed method AFFG are depicted in Figures 4(b) and 4(c), respectively. The objective (fitness) here is to raise the structure's first natural frequency by appropriately choosing the 18 key point locations (our design variables) as depicted in Figure 4(a).

Search begins with an initial population. The maximum fitness in the initial population is nearly 9.32. Over several generations, the fitness gradually evolves to a higher value of 11.902. Figure 5.a and 5.b show the graph of best, average and worst fitness vs. generation for one instance of GA run. This performance curve shows the learning activity or adaptation associated with the algorithm. The total number of generations is 50. For a population size of twenty, this requires 1000 (50*20) fitness evaluations for GA while the proposed GA-AFFG required only 570.4 fitness evaluations. Figure 6 shows the graph of number of FEA vs. generation during one run [37].

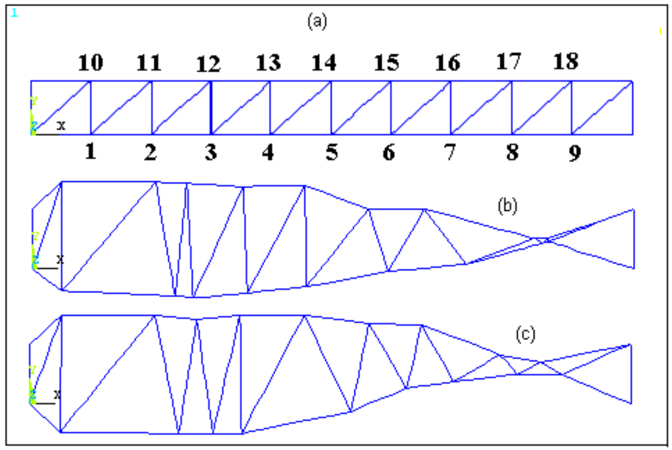

Figure 4. 2-D Truss Frame: (a) Initial configuration,(b) GA optimized shape,(c) GA-AFFG optimized shape

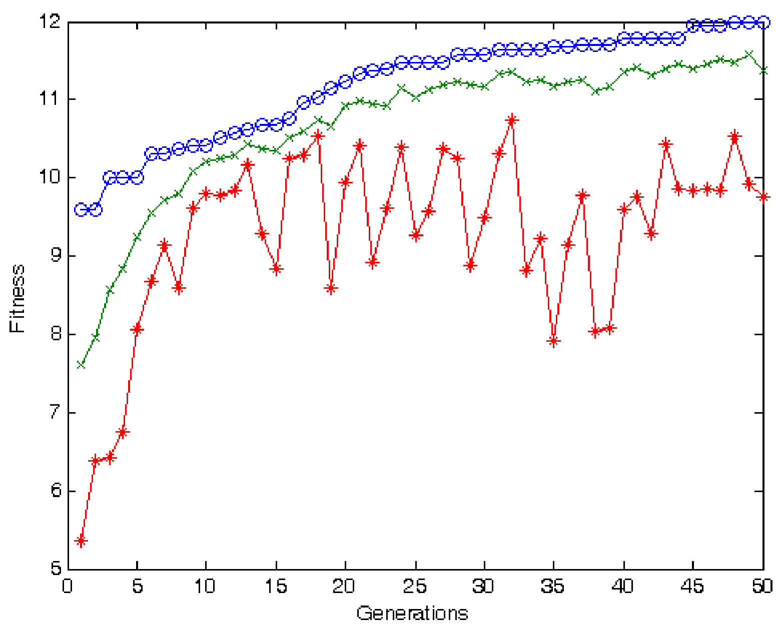

Figure 5.a Generation vs. Fitness, for 2-D Truss frame Simple GA: best (circle), average (cross) and worst (asterisk) individuals at each generation.

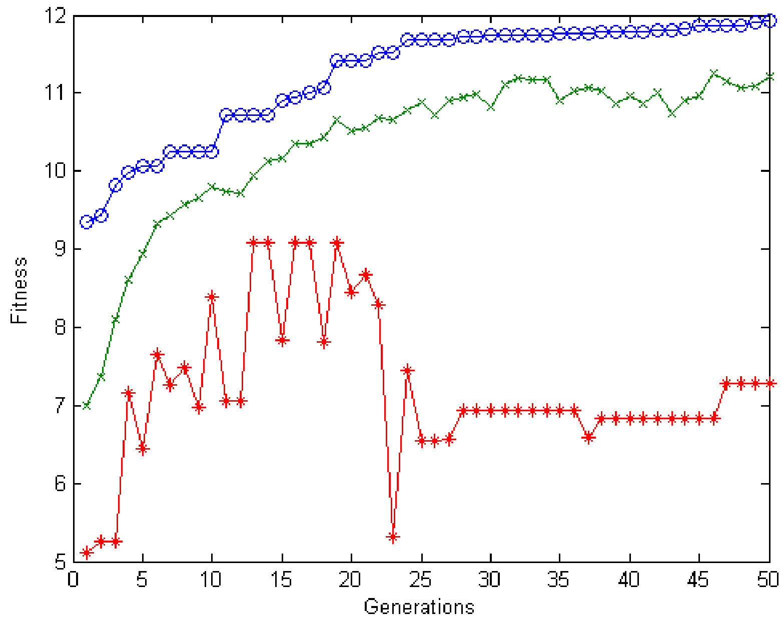

Figure 5.b Plot of Generation vs. Fitness for 2-D Truss frame using GA-AFFG: best (circle), average (cross) and worst (asterisk) individuals at each generation.

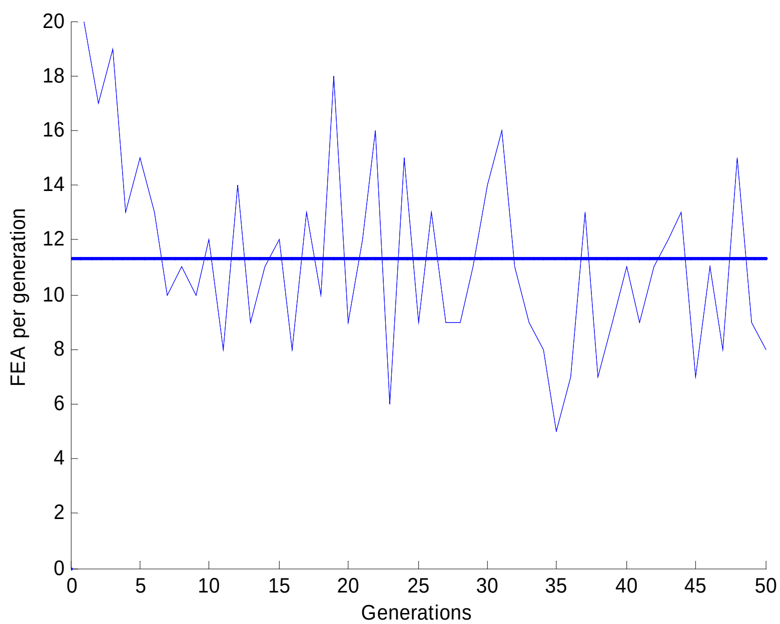

Figure 6. Generation vs. number of FEA for 2-D Truss frame in a single run using GA-AFFG.

Voltage and Pattern Design of Piezoelectric Actuator

Piezoelectric materials exhibit both direct and converse piezoelectric effects. The direct effect (electric field generation as a response to mechanical strains) is used in piezoelectric sensors; the converse effect (mechanical strain is produced as a result of an electric field) is used in piezoelectric actuators. Piezoelectric materials are reliable and efficient first of all in sensor applications but thermal and moisture variations influence the accuracy of measurements. Piezoceramics contain a large number of crystallites sintered together and polarized by an external electrical field. Piezoelectric application can be categorized as: ultrasound applications such as in medical and flow control; sensors such as in strain gauges and pressure transducers; actuators such as in vibration/noise control of adaptive structures; and energy harvesting.

In this study, we consider piezoelectric material PS5-N (from Philips Components) as characterized by its electro-mechanical properties in Table 5. The Solid 5 is used to model both plate and piezoelectric patches [27]. After performing mesh sensitivity, finite element mesh is built as depicted in Figure. 8. Furthermore, we consider here a cantilevered plate that is clamped at its left edge and is not subjected to a mechanical load as in Figure 7. The plate has a length of 154 mm; width of 48 mm and thickness of 0.5 mm. The piezoelectric actuators are attached to the top surfaces of this plate, each having a thickness of 0.3 mm. Electric voltage is only applied to the piezoelectric actuator patches, which are chosen as a design domain where the piezoelectric material can be removed as shown in Figure. 8. The desired pre-defined surface [26] is defined as below:

$$d_{i,j}^{d} = \gamma(x, y) = \left(1.91x^2 + 0.88xy + 0.19x\right)\times 10^{-4}$$

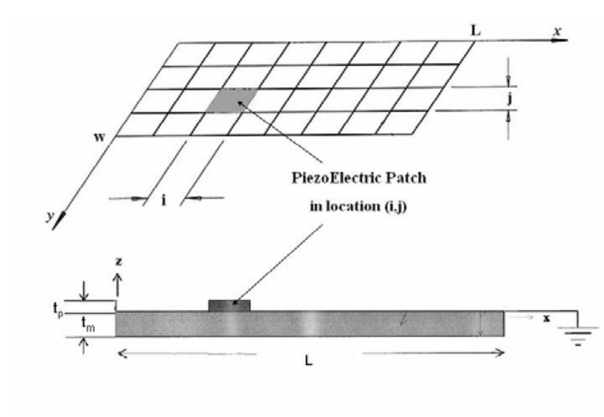

Figure 7. Geometrical model of piezoelectric patch

Table 5. Material properties for the PX5-N piezoelectric material [26].

$C_{11}^{E}\left(\mathrm{N\,m^{-2}}\right)$	13.11×10^{10}	$d_{15}\left(\mathrm{m\,V^{-1}}\right)$	515×10^{-12}
$C_{12}^{E}\left(\mathrm{N\,m^{-2}}\right)$	7.984×10^{10}	$d_{31}\left(\mathrm{m\,V^{-1}}\right)$	-215×10^{-12}
$C_{13}^{E}\left(\mathrm{N\,m^{-2}}\right)$	8.439×10^{10}	$d_{33}\left(\mathrm{m\,V^{-1}}\right)$	500×10^{-12}
$C_{33}^{E}\left(\mathrm{N\,m^{-2}}\right)$	12.31×10^{10}	$\varepsilon_{11}^{t}/\varepsilon_{0}$	1800
$C_{44}^{E}\left(\mathrm{N\,m^{-2}}\right)$	2.564×10^{10}	$\varepsilon_{33}^{t}/\varepsilon_{0}$	2100
$C_{66}^{E}\left(\mathrm{N\,m^{-2}}\right)$	2.564×10^{10}	$\rho\left(\mathrm{kg\,m^{-3}}\right)$	7800

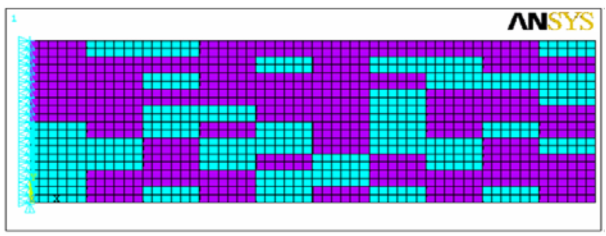

Figure 8. Finite Element Model built by ANSYS.

Since there are a total of 100 piezoelectric patches, there are a total of 200 design variables. Half of these design variables belong to actuation voltage of piezoelectric patches which varies between -10 and 20 V and the rest of the design variables indicates whether any voltage should be applied to the corresponding piezoelectric patch. This means that piezoelectric pattern vector P is binary. When (i,j)th ($i = 1, \ldots, 25; j=1,\ldots,4$) piezoelectric pattern variable is 0, piezoelectric patch is not built so there is not any actuation voltage and vice versa. After assignment of design variables by GA or GA-AFFG, they will be used by either FEA in ANSYS or AFFG to determine fitness.

Piezoelectric design for static shape control

The shape control problem considered in this thesis focuses on voltage and piezoelectric actuator pattern design by finding optimum values of applied voltages and actuator. By considering the pattern parameter vector of Piezoelectric actuators P and the applied voltage vector V as design variables, the quasi-static shape control problem can be generally defined to determine design variables $S=[P,V]$ that minimizes,

$$\text{Minimize } f(S) = \sum_{j=1}^{N_x} \sum_{i=1}^{N_y} \frac{\left| d_{i,j}^{d} - d_{i,j}^{f} \right|}{\left| \max\left(d_{i,j}^{d}\right) \right|} \Bigg/ \left(N_x + N_y\right) \tag{4}$$

where N is the number of patches, $d_{i,j}^{d}$ and $d_{i,j}^{f}$ are the desired and actual transverse displacements or z in Cartesian plane. Here, the pattern variables P in vector S are chosen to be the distribution of active piezoelectric actuator material, the voltage variables in vector V are the electrical potentials applied across the thickness direction of each actuator, and N_x and N_y equal 25 and 4, respectively. Plates are considered to be in the (*x*, *y*) Cartesian plane pattern as is illustrated in Figures 7 and 8. Since displacement is small here, there is no need to consider stress or strain constraint variables for the shape control problem. Figures 9(a), 9(b) and 10 show the graph of best, average and worst fitness vs. generation and number of FEA vs. generation respectively.

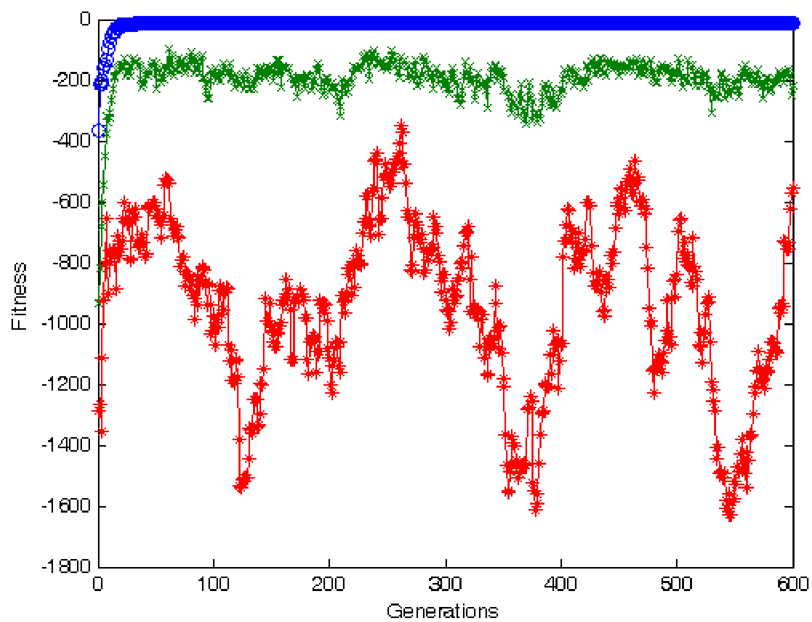

Figure 9.a Generation vs. Fitness for Piezoelectric actuator using simple GA for a single run: best (circle), average (cross) and worst (asterisk) of individuals at each generation.

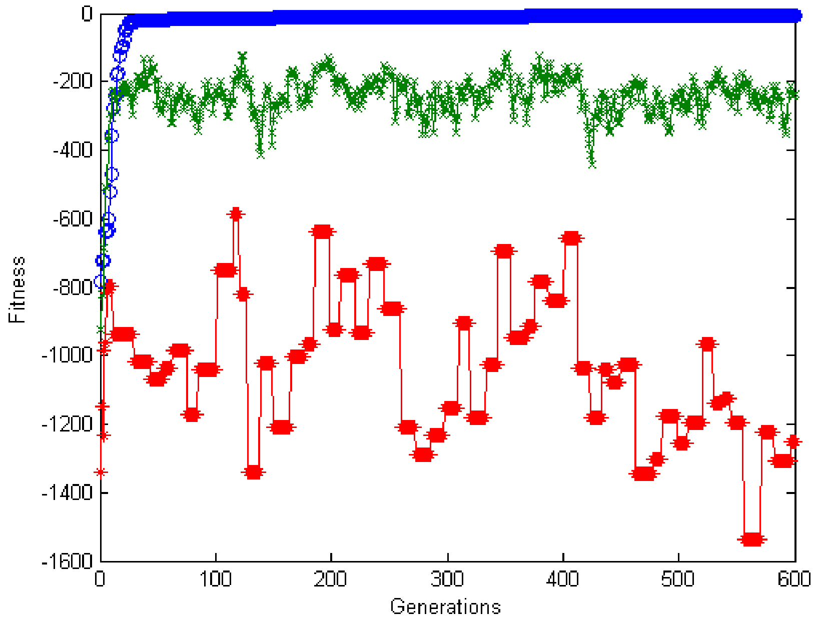

Figure 9.b Generation vs. Fitness for Piezoelectric actuator using proposed GA-AFFG for single run: best (circle), average (cross) and worst (asterisk) of individuals at each generation.

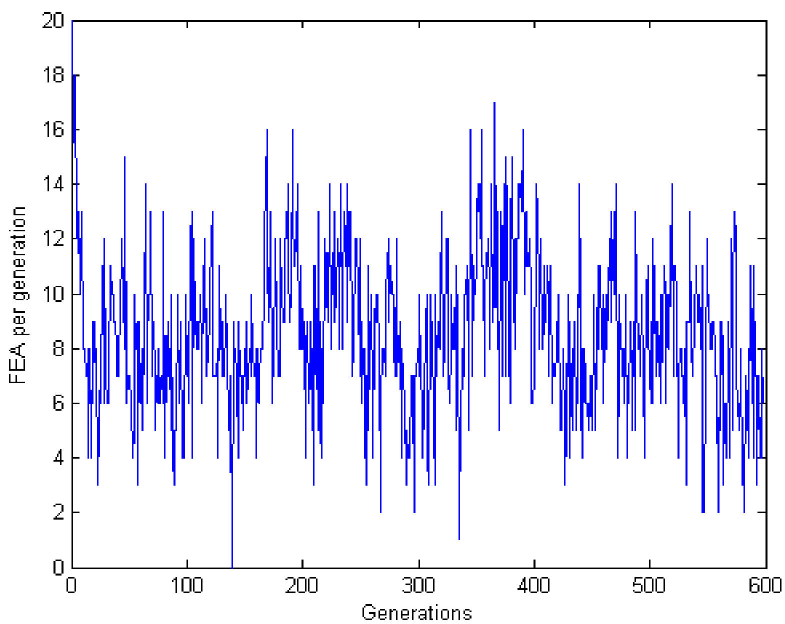

Figure 10. Generation vs. number of FEA, for Piezoelectric actuator, using GA-AFFG for a single run.

Analysis of Results

Tables 6 - 9 illustrate the performance of the proposed GA-AFFG method in comparison with GA, FES and GA-NN for the four structural problems. Due to the random nature of the results, the first three design simulations are repeated 10 times and statistical analysis is performed. However, Piezoelectic actuator design could not be repeated as many times due to the time consuming nature of FEA in this problem.

Table 6. Performance of the optimization methods (Ten run average) for 3-layer composite beam, $\alpha = 0.9$, $\beta = 0.1$, $\gamma = 30$, $M = 5$, $N_G = 250$, T = 0.7.

	Design variables	FEA Evaluations	Number of training data	Improved Time	Optimum frequency (s^{-1})	p-value
GA	2	1000	-	-	19.3722	-
FES	2	228.1	-	%77.19	19.369	0.6737
GA-NN[15]	2	155.9	100	%74.41	19.3551	0.1164
GA-AFFG	2	97.5	-	%90.25	19.3681	0.1944

Table 7. Performance of the optimization methods (Ten run average) for Airplane wing, $\alpha = 0.9$, $\beta = 0.5$, $\gamma = 1$, $M = 5$, $N_G = 250$, T = 0.7.

	Design variables	FEA Evaluations	Number of training data	Improved Time	Optimum frequency (s^{-1})	p-value
GA	6	1000	-	-	6.0006	-
FES	6	481.6	-	%51.84	5.9801	0.7623
GA-NN[15]	6	172.1	100	%72.79	5.9386	0.4057
GA-AFFG	6	173.5	-	%82.65	6.0527	0.1286

Table 8. Performance of the optimization methods (Ten run average) for 2D-Truss, $\alpha = 0.9$, $\beta = 0.11$, $\gamma = 3.05$, $M = 5$, $N_G = 550$, T = 0.7.

	Design Variables	FEA Evaluations	Number of training data	Improved Time	Optimum frequency (s^{-1})	p-value
GA	36	1000	-	-	12.1052	-
FES	36	1000	-	%0	11.8726	0.0081
GA-NN [15]	36	293	100	%60.66	11.8697	0.0203
GA-AFFG	36	570.4	-	%42.96	12.1160	0.7262

Tables 6 - 8 illustrate a comparison of the GA, FES and GA-NN [15] with GA-AFFG algorithms in terms of computational efficiency and performance for 3-layer composite beam, Airplane wing and 2-D

Truss design problems respectively. The second column in these tables makes a comparison of the three algorithms in terms of number of FEA evaluations as compared to GA, while the fourth column makes a comparison in terms of performance. Results indicate that GA-AFFG finds statistically equivalent solutions by using more than 90%, 82% and 42% fewer finite element evaluations. The GA-NN also reduces the number of FEA significantly, but average performance is inferior when compared with GA-AFFG due to NN's approximation error. It must be noted that the GA-NN's improved time includes the number of initial training data.

For the piezoelectric actuator design problem, Table 9 illustrates a comparison of the GA, FES and GA-NN [15] with GA-AFFG algorithms in terms of computational efficiency and performance. The second column in Table 9 makes a comparison of the three algorithms in terms of the number of FEA evaluations as compared with GA, while the fourth column makes a comparison in terms of quality of optimal solutions. Results indicate that GA-AFFG finds equivalent solutions by using 57% fewer finite element evaluations as compared to GA. Also, when compared with GA-NN, the proposed algorithm finds better solutions while being more computationally efficient. The main difference, here, is NN's need for pre-training. Trying various sizes of initial training set and considering the 200 parameters, the NN required at least 5000 training data pairs for comparable performance.

Overall, when compared with GA, the two sets of application indicate that FES, GA-NN and GA-AFFG improve computational efficiency of their problem by reducing the number of exact fitness function evaluations. However, the neuro-approximation as well as the fitness inheritance fails with growing size of input-output space. Consequently, the utility of AFFG becomes more significant in larger and more complex design problems. Furthermore, statistical analysis confirms that fitness inheritance is more comparable in terms of performance when size of search space is smaller (Tables 6 and 7), but its performance deteriorates as problem complexity increases (Tables 8 and 9).

Table 9. *Piezoelectric Actuator performance of the optimization methods.* $\alpha = 0.9$, $\beta = 0.11$, $\gamma = 3.05$, $M = 5$, $N_G = 550$, T = 0.7.

	Design Variables	FEA Evaluations	Number of training data	Improved Time	Minimum error pre-defined surface
GA	200	12000	-	-	%7.313
FES	200	12000	-	%0	%12.82
GA-NN	200	2617	5000	%36.52	%8.093
GA-AFFG	200	5066	Not needed	%57.64	%7.141

A comparison of the number of exact fitness function evaluations in terms of mean and variance that presents the improved computational time is presented in Table 4 for the above first three mechanical optimization problems. A paired t-test of significance is also performed to study the significance of lower computation cost. Since the forth optimization problem (Piezoelectic actuator design) could not be repeated due to the its FEA time consuming nature, t-test could not be performed.

Table 10: Mean and Var represent the mean and variance of number of real fitness calculation for ten runs respectively. A paired t-test of significance is also performed. F7, F8 and F9 are 3-layer Composite Beam, Airplane Wing and 2-D Truss Frame mechanical optimization problems, respectively.

Function	GA	FES			GA-NN			GA-AFFG		
	Mean	Mean	Var	p -value	Mean	Var	p -value	Mean	Var	p -value
F7	1000	228.1	4601.2	4.9×10^{-11}	155.9	511.9	1.1×10^{-15}	97.5	406.7	2.2×10^{-16}
F8	1000	481.6	38648	1.6×10^{-5}	172.1	6392.1	1.1×10^{-10}	173.5	1600.3	2.3×10^{-13}
F9	1000	1000	0	NA*	293	2394.2	5.8×10^{-12}	570.4	18477	3.6×10^{-6}

* Not Available

4. Conclusion

Evolutionary cycles are ruled by competitive games of survival and not absolute measures of fitness. By exploiting this robustness of evolution against uncertainties in fitness function evaluations, the proposed adaptive fuzzy fitness granulation provides a method to selectively reduce the number of fitness function evaluations by considering the similarity/indistinguishability of an individual to a pool of fuzzy information granules. Since the approach does not use approximation or on-line training, it is not caught in the pitfalls of such techniques such as false peaks, large approximation error due to extrapolation, and time consuming online training.

Numerically, the above approach is tested on two sets of simulations. The first set of simulations is a number of standard optimization benchmarks chosen for their various features such as multimodality and nonlinearity. The second set of problems is four mechanical hardware designs that are highly computationally intensive. Results indicate that the proposed method could lead to improvement in computation time while maintaining performance by its accurate but selective evaluations of actual fitness functions. Statistical analysis confirms that the proposed method demonstrates an ability to reduce computation without sacrificing performance. Furthermore, this improvement is more significant when the problem grows larger.

REFERENCES

[1] X. Zhang, B. Julstrom, and W. Cheng, “Design of vector quantization codebooks using a genetic algorithm”, IEEE International Conference on Evolutionary Computation, pp. 525–529, 1997.

[2] J.-H. Chen, D. Goldberg, S.-Y. Ho, and K. Sastry, “Fitness inheritance in multiobjective optimization”, Proceedings of genetic and Evolutionary Computation Conference, pp. 319–326, 2002.

[3] K. Sastry, D. Goldberg, and M. Pelikan, “Don’t evaluate, inherit”, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 551–558, 2001.

[4] M. Salami and T. Hendtlass, “A fast evaluation strategy for evolutionary algorithms”, Applied Soft Computing, Vol. 2, NO. 3, pp. 156–173, 2003.

[5] E. Ducheyne, B. De Baets, and R. deWulf, “Is fitness inheritance useful for real-world applications?”, in Evolutionary Multi-Criterion Optimization, ser. LNCS 2631. Berlin, Germany: Springer-Verlag, pp. 31–42, 2003.

[6] H.-S. Kim and S.-B. Cho, “An efficient genetic algorithms with less fitness evaluation by clustering”, In Proceedings of IEEE Congress on Evolutionary Computation, pp. 887–894, 2001.

[7] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” Soft Computing., Vol. 9, No. 1, pp. 3–12, 2005.

[8] M. Bhattacharya and G. Lu, “A dynamic approximate fitness-based hybrid EA for optimization problems”, Proceedings of IEEE Congress on Evolutionary Computation, pp. 1879–1886, 2003.

[9] Y. Jin and B. Sendhoff, , “Reducing fitness evaluations using clustering techniques and neural network ensembles”, in Genetic and Evolutionary Computation, ser. LNCS 3102. Berlin, Germany: Springer-Verlag, pp. 688–699, 2004.

[10] B. Whitehead, “Genetic evolution of radial basis function coverage using orthogonal niches”, IEEE Transactions on Neural Networks, Vol. 7, NO. 6, pp. 1525–1528, 1996.

[11] R. Myers and D. Montgomery. Response Surface Methodology. John Wiley & Sons, Inc., New York, 1995.

[12] J. Sacks,W. J.Welch, T. J. Michell, and H. P.Wynn. “Design and analysis of computer experiments”, Statistical Science, Vol. 4, NO. 4, pp. 409–435, 1989.

[13] Y.-S. Hong, H.Lee, and M.-J. Tahk. "Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks", Engineering Optimization, Vol. 35, NO. 1, pp. 91–102, 2003.

[14] Kok Sung Won; Ray, T.; Kang Tai "A framework for optimization using approximate functions", Evolutionary Computation, 2003. CEC '03. The 2003 Congress on Vol. 3, pp.1520 – 1527, 2003.

[15] Amir R. Khorsand, Mohammad R. Akbarzadeh, “Multi-objective meta level soft computing-based evolutionary structural design”, Journal of the Franklin Institute, Vol. 9, NO. 5, pp. 595–612, August 2007.

[16] Gaspar-Cunha, A., Vieira, A., “A Multi-Objective Evolutionary Algorithm Using Neural Networks To Approximate Fitness Evaluations”, International Journal of Computers, Systems, and Signals**,** 2004.

[17] Y. S. Ong, P. B. Nair, and A. J. Keane. “Evolutionary optimization of computationally expensive problems via surrogate modeling”, AIAA Journal, Vol. 40, NO. 4, pp. 687–696, 2003.

[18] H. Ulmer, F. Streichert, and A. Zell. “Evolution strategies assisted by gaussian processes with improved pre-selection criterion”, In Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, Canberra, Australia, pp. 692–699. IEEE Press, 2003.

[19] L. Willmes, T. B¨ack, Y. Jin, and B. Sendhoff. “Comparing neural networks and kriging for fitness approximation in evolutionary optimization”, In Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, Canberra, Australia, pp. 663–670. IEEE Press, 2003.

[20] H. Ulmer and F. Streichert and A. Zell, “Evolution strategies with controlled model assistance”, Congress on Evolutionary Computation, pp. 1569-1576, 2004

[21] J. Dennis and V. Torczon, “Managing approximate models in optimization”, In N. Alexandrov and M. Hussain, editors, Multidisciplinary design optimization: State-of-the-art, pp. 330-347, SIAM, 1997.

[22] Yaochu Jin, Markus Olhofer, Bernhard Sendhoff, “On Evolutionary Optimization with Approximate Fitness Functions”, GECCO 2000, pp. 786-793, 2000.

[23] L. A. Zadeh, “Fuzzy Sets and Information Granularity”, in Advances in Fuzzy Set Theory and Applications, M. Gupta, R. Ragade and R. Yager (eds.), pp. 3-18, Amsterdam: North-Holland Publishing Co., 1979

[24] L. A. Zadeh, “Towards a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic”, Fuzzy Sets and Systems, Vol. 90, NO. 2, p.111-127, Sept. 1997.

[25] K. A. De Jong, “An Analysis of the Behavior of a Class of Genetic Adaptive Systems”, Ph.D. dissertation, University of Michigan, 1975.

[26] S.Silva, R.Ribeiro, J.Rodrigues, M.Vaz and J.Monteiro, “The application of genetic algorithms for shape control with piezoelectric patches—an experimental comparison”, Smart Materials and Structure Journal, No. 13, pp. 220–226, 2004.

[27] ANSYS User’s Manuals.

[28] M.-R. Akbarzadeh-T., I. Mosavat, and S. Abbasi, “Friendship Modeling for Cooperative Co-Evolutionary Fuzzy Systems: A Hybrid GA-GP Algorithm”, Proceedings of the 22nd International Conference of North American Fuzzy Information Processing Society, pp.61-66, Chicago, Illinois, 2003.

[29] Maumita bhattacharya, “Meta Model Based EA for Complex Optimization”, International Journal of Computational Intelligence, Vol. 4, NO. 1, 2007.

[30] M. Davarynejad, M.-R. Akbarzadeh-T, N. Pariz, “A Novel General Framework for Evolutionary Optimization: Adaptive Fuzzy Fitness Granulation”, Proceedings of the 2007 IEEE International Conference on Evolutionary Computing, pp. 951-956, Singapore, September 25-28, 2007.

[31] Mehrdad Salami, Tim Hendtlass, “The Fast Evaluation Strategy for Evolvable Hardware”, Genetic Programming and Evolvable Machines, Vol. 6, pp. 139-162, 2005.

[32] M. Husken, Y. Jin, B. Sendhoff, “Structure optimization of neural networks for evolutionary design optimization”, Soft Computing, Vol. 9, NO. 1, pp. 21–28, 2005.

[33] Z. Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane and K. Y. Lum, “Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization”, IEEE Transactions on Systems, Man and Cybernetics, Part C: Reviews and Applications, Vol. 37, NO. 1, pp. 66-76, Jan 2007.

[34] K.C. Giannakoglou, D.I. Papadimitriou, I.C. Kampolis, "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels", Computer Methods in Applied Mechanics and Engineering, Vol. 195, pp. 6312-6329, Sept 2006.

[35] F. Herrera, M. Lozano. "Adaptive Genetic Algorithms Based on Coevolution with Fuzzy Behaviours", IEEE Transactions on Evolutionary Computation, Vol. 5, NO. 2, pp. 149-165, 2001.

[36] F. Herrera, M. Lozano. "Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy and Future Directions", Soft Computing Vol. 7, NO. 8, pp. 545-562, 2003.

[37] M. Davarynejad, M.-R. Akbarzadeh-T, N. Pariz, A-R khorsand, "Adaptive Fuzzy Fitness Granulation in Structural Optimization Problems", Proceedings of the 2007 IEEE Multi-conference on Systems and Control (MSC), pp. 172-177, Singapore, October 1- 3, 2007.

[38] Margarita Reyes Sierra and C. A. Coello Coello, "Fitness Inheritance in Multi-Objective Particle Swarm Optimization", In IEEE Swarm Intelligence Symposium, pp. 116-123, 2005.

[39] Margarita Reyes Sierra and C. A. Coello Coello, "A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization", In the proceedings of the Congress on Evolutionary Computation, pp. 65-72, 2005.