Adaptive Fuzzy Fitness Granulation in Structural Optimization Problems

M. Davarynejad, M.-R. Akbarzadeh-T, IEEE Senior Member, N. Pariz, A-R khorsand

Abstract— **Computational complexity is a prohibitive factor in evolutionary optimization of sufficiently large and/or complex problems. Much of this computational complexity is due to the fitness function evaluation that may either not exist or be computationally very expensive. Here, we investigate the use of fitness granulation via an adaptive fuzzy similarity analysis as applied to two different hardware design problems that are evaluated using finite element analysis. The first design problem is a relatively simpler 2-D Truss Frame Design with 36 parameters while the second problem is Piezoelectric voltage and pattern arrangement design for static shape control in which 200 parameters are optimized. In comparison with standard application of evolutionary algorithms, statistical analysis reveals that the proposed method significantly decreases the number of fitness function evaluations while finding equally good or better solutions. Additionally, this more improvement is indicated with higher problem complexity.**

I. Introduction

As the field of evolution-based algorithms matures and tackles more real-world applications, its limitations and challenges also become clearer. Fitness function evaluation is often the most prohibitive and limiting segment of artificial evolutionary algorithms. In this case, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. In design of mechanical structures, for instance, each exact fitness evaluation requires the time consuming stage of finite element analysis which may require anywhere from several seconds to several days. In a conventional genetic algorithm with a fixed and modest population size of 100 and 100 number of generations, and a very small scale structural problem that requires only 10 seconds, this means about thirty hours of computing.

To alleviate this problem, various methods have been proposed to date. Fitness inheritance is a popular subclass of fitness function approximation methods where fitness is simply inherited [1, 2]. A similar approach to fitness inheritance has also been suggested in [3] where the fitness of a child individual is the weighted sum of its parents. This simple strategy can fail in sufficiently complex and multi objective problems [4].

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 rising problem is estimating 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 [9] and fitness estimation by association/friendship [10] have been developed.

Other common approaches are based on learning and interpolation from known fitness values of a small population. Specifically, one widely used method in design engineering include the response surface methodology that uses low-order polynomials and the least square estimations [11], and 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.

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 become good estimators of fitness function if provided with sufficient complexity of their neuro-structure and richness of training data points [16]. As with any other 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 of using fitness approximation models and 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

M.-R.Akbarzadeh-T. is with the Electrical Engineering Department, Ferdowsi University of Mashhad. He is also currently a Visiting Scholar at BISC, UC Berkeley (e-mail: akbarzadeh@ieee.org). Author would also like to acknowledge the generous support of the Islamic Development Bank under the 2005-2006 IDB Merit Scholarship for High Technology.

M. Davarynejad, M.Sc. student at Electrical Engineering Department, Ferdowsi University of Mashhad (e-mail: davarynejad@kiaeee.org)

N. Pariz is with the Department of Electrical Engineering, Ferdowsi University of Mashhad, 9177948944-1111, Iran (e-mail: n-pariz@um.ac.ir)

A.-R. Khorsand is currently with the Department of Mechanical & Manufacturing engineering, University of Manitoba, Winnipeg, Canada (e-mail: khorsand@ieee.org)

1-4244-0441-X/07/$20.00 ©2007 IEEE.

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.

Alternatively, if individuals in a population can be clustered into several groups as in [5], 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 [6]. The idea of fitness imitation has been extended and more sophisticated estimation methods have been developed in [7], [8].

Because of the high dimensionality and limited number of training samples, constructing a globally correct approximate model remains to be difficult. Evolutionary algorithms using such approximate fitness functions may converge to false optimums. Therefore, it can be beneficial to selectively use the original fitness function together with the approximate model [6]. In conventional optimization, this is commonly known as model management [20] or evolution control in evolutionary computation [21]. For example, Khorsand and Akbarzadeh [15] recently investigated structural design by a hybrid of neural network and finite element analysis. 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 peeks.

In this paper, fitness is not interpolated or estimated; rather the uncertainty in the similarity among real solutions is exploited. In the proposed algorithm as explained in detail by authors in [19, 22], 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, 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, granules that are not used are gradually eliminated.

This fuzzy granulation scheme is applied here to solving two structural optimization problems as follows. The optimization method is briefly reviewed in Section 2. The two hardware design problems that are evaluated via finite element analysis are investigated and simulated in Section 3. The first structural design problem is static shape control of a 2-D truss frame structure and the second is Piezoelectric actuate, 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.

II. Adaptive Fuzzy Fitness Granulation (AFFG) [19]

A. The Main Idea

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. The 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 win the new solution and thereby to prolong their lives in the pool. In turn, the new individual simply assumes fitness of the winning (most similar) 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 win new individuals are gradually eliminated in order to avoid a continuously enlarging pool. The proposed algorithm is shown in Figure 1.

As is shown in Figure 1, a random parent population $P_0 = \left\{X_1^1, X_2^1, \ldots, X_j^1, \ldots, X_t^1\right\}$ is initially created, where $X_j^i = \left\{x_{j,1}^i, x_{j,2}^i, \ldots, x_{j,r}^i, \ldots, x_{j,m}^i\right\}$ 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 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. $X_1^1 = \left\{x_{1,1}^1, x_{1,2}^1, \ldots, x_{1,r}^1, \ldots, x_{1,m}^1\right\}$ is chosen as the center

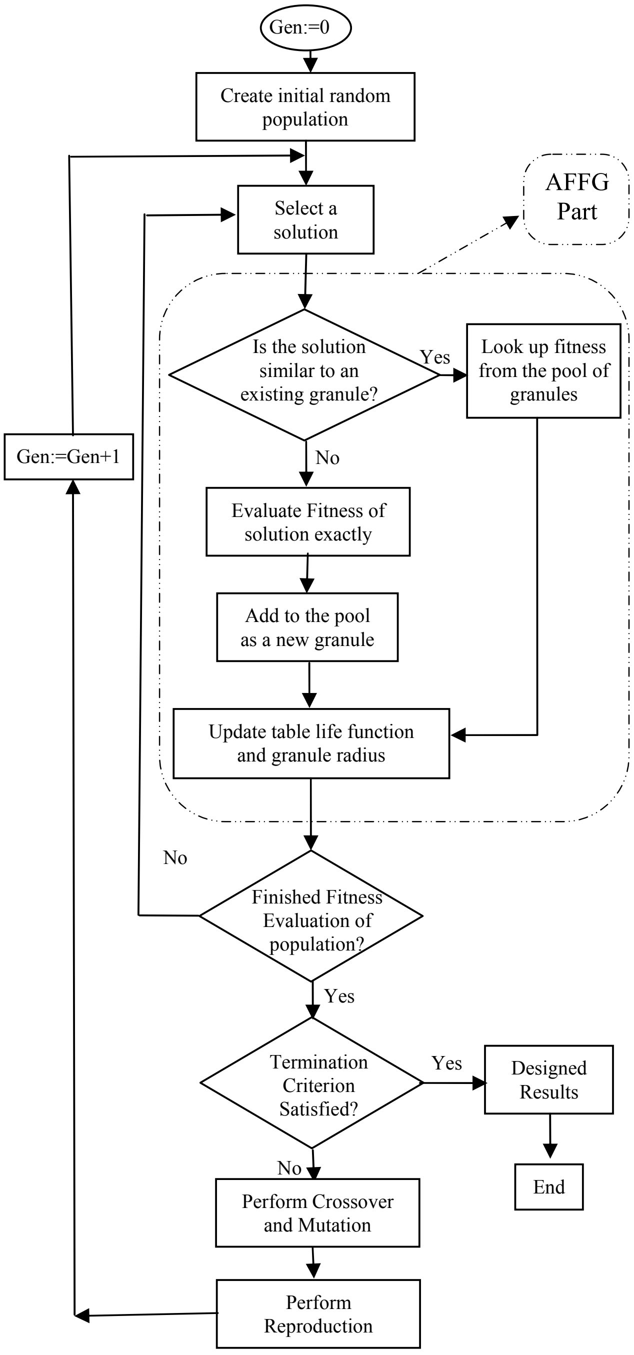
Gen:=0
Create initial random population
Select a solution
AFFG Part
Is the solution similar to an existing granule?
Yes
Look up fitness from the pool of granules
No
Evaluate Fitness of solution exactly
Add to the pool as a new granule
Update table life function and granule radius
Gen:=Gen+1
No
Finished Fitness Evaluation of population?
Yes
Termination Criterion Satisfied?
Yes
Designed Results
End
No
Perform Crossover and Mutation
Perform Reproduction

Figure 1. Flowchart of the Purposed AFFG Algorithm

$C_1 = \{c_{1,1}, c_{1,2}, ..., c_{1,r}, ..., c_{1,m}\} = X_1^1$ of first granule. The membership function $\mu_{r,k}$ therefore describes a Gaussian similarity neighborhood for each parameter k as follows,

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

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

Then, the average similarity of a new solution $X_j^i = \left\{x_{j,1}^i, x_{j,2}^i, ..., x_{j,r}^i, ..., x_{j,m}^i\right\}$ 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 exact fitness function computing 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.

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

where $\theta^i = \alpha.\frac{Max\left\{f(X_1^{i-1}), f(X_2^{i-1}), ...f(X_t^{i-1})\right\}}{\bar{f}^{i-1}}$,

$K = index \max\limits_{k\in\{1,\ 2,\ ...,\ l\}} \left\{\overline{\mu}_{j,k}\right\}$, $\bar{f}^i = \sum_{j=1}^{t} \frac{f\left(X_j^i\right)}{t}$, 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 $\max\limits_{k\in\{1,2,...,l\}} \left\{\overline{\mu}_{j,k}\right\} < \theta^i$, X_j^i is chosen as a newly created granule.

B. Adaptation in the Width of Membership Functions

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.

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

Where $\beta > 0$ is an emphasis operator and γ is constant of proportion. 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 individual. Since, in general, members of the initial populations also have smaller fitness, threshold is also smaller. Therefore, fitness is computed by more often 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.

C. Adaptation in the Length of Granule Pool

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, a "*forgetting factor*" is introduced in order to appropriately decrease the size of the pool. In other word, it is better to remove granules that do not win 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 N and subsequently updated as below,

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

Where M is the life reward of the granule and K is the index of the winning granule for each individual in generation i.

In [22] an example is provided to illustrate the competitive granule pool update law. Additionally, in order to prove the performance of the proposed method, six traditional optimization problems are simulated in [22].

III. Numerical Results in Structural Design Optimization Problems

To illustrate the efficacy of the proposed granulation techniques, we have chosen two mechanical design problems that typically require finite element analysis for their fitness evaluation namely: design of a 2-D Truss frame for maximum rigidity and voltage/pattern design of Piezoelectric actuators. These problems have a higher number of param-eters, hence a more challenging optimization task. Structural optimization benchmarks are considered here because fitness evaluation through conventional finite element analysis is time consuming and usually requires several days to finish even in trivial problems.

Due to the stochastic nature of evolutionary optimization, the first simulation is repeated several times, and a paired t-test of significance is performed.

The GA routines utilized random initial populations, binary-coded chromosomes, mutation, fitness scaling, and an elitist stochastic universal sampling selection strategy. Also we have used single-point crossover for 2-D truss Frame and 15-point crossover for piezoelectric actuator. The probabilities of crossover ($P_{CROSSOVER} = 1$) and mutation ($P_{MUTATION} = 0.01$), and the population size (Number of individuals = 20) in each generation, and population size of 50 for 2-DTruss and population size of 600 for piezoelectric actuator. Finally individual length varies depending on the number of variables in a given problem but each variable is allocated 8 bits.

A. 2-D Truss Frame

A typical truss designed by an engineer is depicted in Figure 2(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/m3). The optimized shapes by GA and the new proposed method AFFG are depicted in Figures 2(b) and Figure 2(c) respectively. The objective 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 2(a).

Search begins with an initial population. The maximum fitness in initial population is nearly 9.32. Over several generations, the fitness gradually evolves to a higher value of 11.902. For a population size of twenty, this requires 1000 (50 X 20) simulations for GA. The GA-AFFG strategy used here is required only 570.4 simulations.

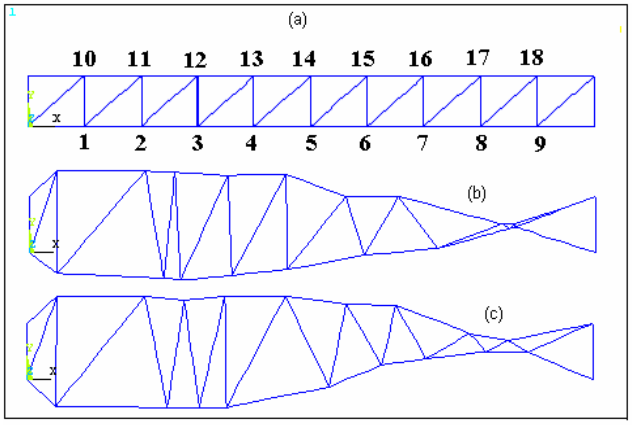

Figure 2. (a) Initial configuration, (b) GA optimized shape, (c) GA-AFFG optimized shape.

B. 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.

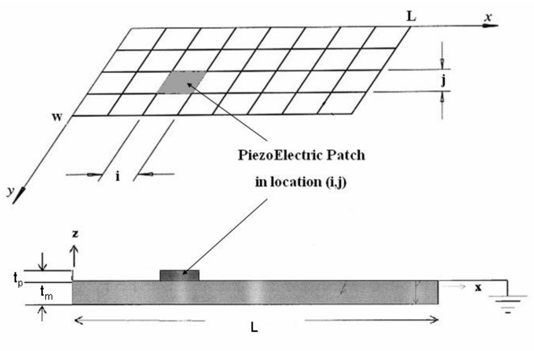

Figure 3. Geometrical model of piezoelectric patch

C. Piezoelectric Design for Static Shape Control

The shape control problem considered in this paper 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,

$$\text{Minimize } f(S)=\sum_{i=1}^{N}\frac{\left|d_i^d-d_i^f\right|}{\left|\max\left(d_i^d\right)\right|}\Bigg/N \qquad (3)$$

Where in the context of general optimization formulation, S is the design variable vector; f(S) is the objective function; N is number of patches, d_i^d and d_i^f are the desired and actual transverse displacements or z in Cartesian plane. In this paper, 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 equals 100. Plates are considered to be in x, y Cartesian plane pattern as is illustrated in Figure 3.

D. Model Details

Consider a cantilevered plate that is clamped at its left edge and is not subjected to a mechanical load as in Figure 3. 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. Only the piezoelectric actuator patches are applied electric voltage, which are chosen as a design domain where the piezoelectric material can be removed. Desired pre-defined surface [23] is defined as below.

$$d_i^d=\gamma(x,y)=\left(1.91x^2+0.88xy+0.19x\right)\times10^{-4}$$

For more details on piezoelectric electro-mechanical properties the readers are referred to [23].

Since there are 100 piezoelectric patches, there are a total of 200 design variable here. 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 voltage should be applied to the corresponding piezoelectric patch. This means that piezoelectric pattern vector P is binary. When i-th (i = 1, …, 100) 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 calculate or estimate Equation (3).

IV. Statistical Analysis of Results

Tables I and II illustrates the performance of the proposed GA-AFFG method in comparison with GA and GA-NN for the two structural problems. Due to random nature of results, 2D-Truss design simulation is repeated 10 times and statistical analysis is performed. However, the Piezoelectic actuator design could not be repeated due to the time consuming FEA of this problem.

Table I illustrates the comparison of the GA and GA-NN [15] with GA-AFFG algorithms in terms of computational efficiency and performance for the 2-D Truss design problem. The second column in Table I makes a comparison of the three algorithms in terms of number of FEA evaluations as compared with GA, while the fourth column makes a comparison in terms of performance. Results indicate that GA-AFFG finds statistically equivalent solutions by using 43% fewer finite element evaluations. The GA-NN seems to also reduce number of FEA significantly, but average performance is inferior when compared with GA-AFFG due to NN's approximation error.

Table II illustrates the comparison of the GA and GA-NN [19] with GA-AFFG algorithms in terms of computational efficiency and performance for the Piezoelectric actuator

TABLE I. 2D– TRUSS PERFORMANCE OF THE OPTIMIZATION METHODS (TEN RUN AVERAGE), $\alpha=0.9$, $\beta=0.08$, $\gamma=1$, $M=5$.

	Design variables	FEA evolutions	Number of training data	Improved Time	Optimum frequency (s^{-1})	α	p-value
GA	36	1000	-	-	12.1052	-	-
GA-NN [15]	36	293	100	%60.66	11.8697	0.01	0.0103
GA-AFFG	36	570.4	Not needed	%42.96	12.1160	0.01	0.7262

TABLE II. PIEZOELECTRIC ACTUATOR PERFORMANCE OF THE OPTIMIZATION METHODS, $\alpha=0.9$, $\beta=0.11$, $\gamma=3.05$, $M=5$

	Design variables	FEA evolutions	Number of training data	Improved Time	Minimum error pre-defined surface
GA	200	12000	-	-	%7.313
GA-NN [19]	200	2617	5000	%36.52	% 8.093
GA-AFFG	200	5066	Not needed	%57.64	%7.141

design problem. The second column in Table II makes a comparison of the three algorithms in terms of number of FEA evaluations as compared with GA, while the fourth column makes a comparison in terms of performance. Results indicate that GA-AFFG finds statistically equivalent solutions by using 57% fewer finite element evaluations as compared to GA. It must be noted that for GA-NN, improved time includes the number of training data.

The two sets of simulations indicate that both GA-NN and GA-AFFG improve computational efficiency of their problem by reducing number of exact fitness function evaluations. However, the neuro-approximation fails with growing size of input-output space. Consequently, the utility of AFFG becomes more significant in large design problems.

V. Conclusion

By exploiting evolution's robustness against uncertainties in fitness function evaluations, the proposed adaptive fuzzy fitness granulation provides a method to selectively reduce number of fitness function evaluations without approximating or on-line training. The proposed algorithm detects the similarity between solutions to either create new fuzzy granules or to use results of earlier computations in order to avoid unnecessary computation of fitness, even among members of same generation. This technique overcomes many of the drawbacks of prior methods like: initial training, approximation error, and time consuming online learning.

Two mechanical design problems that typically require finite element analysis for their fitness evaluation and are usually requires several days to finish are simulated. The simulations show that the proposed method could lead to improvement in computation time while keeping performance by its accurate estimates of fitness function for only about 50 percent of individuals in each generation. Statistical analysis confirms that the proposed method demonstrates an ability to reduce computation without sacrificing perform-ance.

VI. Acknowledgements

The first author wishes to thank Dr. M. Khademi for his support.

References

[1] X. Zhang, B. Julstrom, and W. Cheng, "Design of vector quantization codebooks using a genetic algorithm", in Proc. Int. Conf. Evol. Comput., 1997, pp. 525–529.

[2] J.-H. Chen, D. Goldberg, S.-Y. Ho, and K. Sastry, "Fitness inheritance in multiobjective optimization", in Proc. Genetic Evol. Comput. Conf., 2002, pp. 319–326.

[3] M. Salami and T. Hendtlass, "A fast evaluation strategy for evolutionary algorithms", Appl. Soft Comput., vol. 2, pp. 156–173, 2003.

[4] 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, 2003, pp. 31–42.

[5] H.-S. Kim and S.-B. Cho, "An efficient genetic algorithms with less fitness evaluation by clustering," in Proc. Congr. Evol. Comput., 2001, pp. 887–894.

[6] Yaochu. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," Soft Computing., vol. 9, no. 1, pp. 3–12, 2005.

[7] M. Bhattacharya and G. Lu, "A dynamic approximate fitness-based hybrid EA for optimization problems," in Proc. Congr. Evol. Comput., 2003, pp. 1879–1886.

[8] 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, 2004, pp. 688–699.

[9] B. Whitehead, "Genetic evolution of radial basis function coverage using orthogonal niches", IEEE Trans. Neural Netw., vol. 7, no. 6, pp. 1525–1528, 1996.

[10] 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.

[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, 4:409–435, 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, 35(1):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 Volume 3, 8-12 Dec. 2003 Page(s):1520 - 1527 Vol.3

[15] Amir R. Khorsand, Mohammad R. Akbarzadeh, "Multi-objective meta level soft computing-based evolutionary structural design", Journal of the Franklin Institute, Article in press, 2006.

[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, 40(4):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, pages 692–699. IEEE Press, 2003.

[19] M. Davarynejad, "Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization", M.Sc. Thesis. Ferdowsi University of Mashhad/Iran, Department of Electrical Engineering, 2007.

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

[21] Yaochu Jin, Markus Olhofer, Bernhard Sendhoff, "On Evolutionary Optimization with Approximate Fitness Functions". GECCO 2000: 786-793.

[22] M. Davarynejad, M.-R. Akbarzadeh-T, N. Pariz, "A Novel General Framework for Evolutionary Optimization: Adaptive Fuzzy Fitness Granulation", Accepted for Publication at the 2007 IEEE International Conference on Evolutionary Computing, September 25-28, 2007, Singapore.

[23] 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, 2004, pp. 220–226.