Auto-Tuning Fuzzy Granulation for Evolutionary Optimization

M. Davarynejad, M.-R. Akbarzadeh-T, *IEEE Senior Member*, Carlos A. Coello Coello, *IEEE Senior Member*

***Abstract*—Much of the computational complexity involved in employing evolutionary algorithms as optimization tools is due to the fitness function evaluation that may either not exist or be computationally very expensive. With the approach proposed in this paper, the expensive fitness evaluation step is replaced by an approximate model. An intelligent guided technique via an adaptive fuzzy similarity analysis for fitness granulation is used to decide on the use of expensive function evaluations and dynamically adapt the predicted model. In order to avoid tuning parameters in this approach, a fuzzy supervisor known as auto-tuning algorithm is employed with three inputs. The proposed method is then applied to 3 traditional optimization benchmarks with 4 different dimensions each. The effect of the number of granules on the convergence rate is also studied. When comparing the proposed approach with the standard application of evolutionary algorithms, the statistical analysis confirms that the proposed approach demonstrates an ability to reduce the computational complexity of the design problem without sacrificing performance. Furthermore, the auto-tuning of the fuzzy supervisory removes the need for exact parameter determination.**

I. INTRODUCTION

Evolution-based algorithms have long been accepted as promising global optimizers and have shown to be very powerful in solving many real-world and complex optimization tasks such as design optimization, see e.g., [1]. Unfortunately, repeated fitness function evaluation for such complex problems is often the most prohibitive and limiting segment of artificial evolutionary algorithms. Finding optimal solution for complex high dimensional, multimodal problems often requires very expensive fitness function evaluations. In real world problems such as structural optimization problems, one single function evaluation may require several hours or even several days of complete simulation. Typical optimization methods can not deal with such a type of problem. In this case, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. It is apparent that an amalgamation of approximate models may be one of the most promising approaches to convincingly use EAs to solve complex real life problems, especially where: (i) fitness computation time of a single solution is extremely high, (ii) precise models for fitness computation are missing, (iii) the fitness function is uncertain or noisy as a result of simulation/measurement errors or changes in the design variables and/or the environmental parameters, etc. To alleviate this problem, a variety of techniques for constructing the approximation models—often also referred to as metamodels or surrogates—for computationally expensive optimization problems have been investigated [2, 3].

A popular subclass of fitness function approximation methods is fitness inheritance where fitness is simply inherited [4]. A similar approach named “Fast Evolutionary Strategy” (FES) has also been suggested in [9] for fitness approximation where the fitness of a child is the weighted sum of its parents. In [10], it has been shown that this simple strategy can fail in sufficiently complex and multi- objective problems.

Other common approaches based on learning and interpolation from known fitness values of a small population, (e.g. low-order polynomials and the least square estimations [5], artificial neural networks (ANN), including multi-layer perceptrons (MLP) [6], radial basis functions (RBF) [7], support vector machines (SVM) [8, 15] regression models [11], etc.) have also been employed.

Because of the limited number of training samples and high dimensionality encountered in engineering design optimization, constructing a globally valid approximate model remains to be difficult. Evolutionary algorithms using such approximate fitness functions may converge to local optima. Therefore, it can be beneficial to selectively use the original fitness function together with the approximate model [13]. For example, Khorsand and Akbarzadeh [14] recently investigated structural optimization by a hybrid of neural network and finite element analysis, where expensive function evaluation is partially replaced by its approximate model. Also, Jin and Sendhoff [17] applied k-nearest-neighbour method to group the individuals of a population into a number of clusters. For each cluster, only the individual that is closest to the cluster center will be evaluated using the expensive original fitness function.

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 1977 as a technique by which a class of points (objects) are

Mohsen Davarynejad is M.Sc. student at Department of Electrical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, (email: davarynejad@kiaeee.org). He is also working at Cognitive Computing Laboratory of the stated department.

M.-R.Akbarzadeh-T. (*IEEE* Senior *Member*) is with the Department of Electrical Engineering, Ferdowsi University of Mashhad. (e-mail: akbarzadeh@ieee.org).

Carlos A. Coello Coello (*IEEE* Senior *Member*) is with CINVESTAV-IPN (Evolutionary Computation Group), Departmento de Computacion, Av. IPN No. 2508, Col. San Pedro Zacatanco, Mexico D.F. 07300, MEXICO. (email: ccoello@cs.cinvestav.mx).

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 paper, 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. 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. The basic ideas behind the proposed framework are to:

- Avoid initial training.
- Use guided approximation to speed up the search process.
- Exploit the design variable space by means of Fuzzy Similarity Analysis (FSA) as a type of fuzzy approximation model to avoid premature convergence.
- Gradually set up an independent model of initial training data to compensate the Lack of sufficient training data and to reach a model with sufficient approximation accuracy.
- Dynamic updating the approximate model with or without negligible overhead cost.
- Avoid the use of models in unrepresented design variable regions in the training set.

In the proposed algorithm, as explained in detail by the authors in [12, 16], 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 later 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 as a type of fuzzy approximation model to solve 3 traditional optimization benchmarks with 4 different dimensions each.

The paper is organized as follows: Section 2 presents a brief overview of the proposed method. For future details, of the present approach, readers are referred to [16] where the proposed method is described in more detail and also provides an example. In Section 3, an auto-tuning strategy for determining the width of the membership functions (MFs) is presented; which removes the need for exact parameter determination, without obvious influence on convergence speed. Some supporting simulation results and discussion thereof are presented in Section 4. Finally, some conclusions are drawn in Section 5.

II. The AFFG Framework

The AFFG framework [Figure 1] includes a global model of genetic algorithm (GA), hybridized with fuzzy granulation (FG) tool. Expensive fitness evaluation of individuals as required in traditional evolutionary algorithm is partially replaced by an approximation model. Explicit control strategies are used for evolution control, leading to considerable speedup without compromising heavily on solution accuracy.

While approximation is not a new idea in accelerating iterative optimisation process, AFFG focuses on controlled speedup to avoid detrimental effects of approximation. The following section presents the main algorithmic structure of AFFG.

A. The Main Idea

The proposed adaptive fuzzy 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 (FSA) 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.

B. Basic Algorithm Structure

Step One: Create a random parent population $P_0 = \{X_1^1, X_2^1, \dots, X_j^1, \dots, X_t^1\}$, where $X_j^i = \{x_{j,1}^i, x_{j,2}^i, \dots, x_{j,r}^i, \dots, x_{j,m}^i\}$ is the j-th individual in the i-th generation, $x_{j,r}^i$ is the r-th parameter of X_j^i, *m* is the number of design variables and *t* is the population size.

Step Two: Set $G = \{\ \}$, where $G = \{(C_k, \sigma_k, L_k) \mid C_k \in \Re^m, \sigma_k \in \Re, L_k \in \Re, k = 1, ..., l\}$ and 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 (WMFs) of the k-th fuzzy granule, and L_k is the granule's life index.

Step Three: The phenotype of the 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 $C_1 = \{c_{1,1}, c_{1,2}, ..., c_{1,r}, ..., c_{1,m}\} = X_1^1$ of the first granule.

Step Four: The membership function $\mu_{r,k}$ describes a Gaussian similarity neighborhood for each parameter *k* as follows,

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

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

Remark: σ_k is a distance measurement parameter that controls the degree of similarity between two individuals. In [12], σ_k is defined based on equation (2). Based on this definition, the granules shrink or enlarge in reverse proportion to their fitness as below.

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

Where $\beta > 0$ is an emphasis operator and γ is a proportionality constant. The problem arising here is how to determine β and γ as design parameters. The fact is that these two parameters are problem dependent and it is necessary to perform some trials to adjust these parameters. These trials are based on a simple rule: how much do we want to accelerate the optimization procedure? High speed needs to have enlargement in the granule spread and, in consequence, it produces less accuracy in the fitness approximation and vice versa. To overcome this drawback, a fuzzy controller with three inputs is introduced in Section III.

Step Five: Compute 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 Using $\overline{\mu}_{j,k} = \sum_{r=1}^{m} \frac{\mu_{k,r}(x_{j,r}^i)}{m}$.

Step Six: 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.

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

where $f(C_K)$ is the fitness function of the fuzzy granule, $f(X_j^i)$ is the real fitness calculation of the individual,

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

$$K = \underset{k \in \{1,\ 2,\ ...,\ l\}}{index \max} \{\overline{\mu}_{j,k}\}, \quad \bar{f}^{\,i} = \sum_{j=1}^{t} \frac{f(X_j^i)}{t}, \quad \text{and} \quad \alpha > 0$$

is a proportionality constant. Threshold θ^i increases as the best individual's fitness in generation i increases. Hence, as the population matures and reaches higher fitness values, 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 $\max\limits_{k \in \{1,2,...,l\}} \{\overline{\mu}_{j,k}\} < \theta^i$, X_j^i is chosen as a newly created granule.

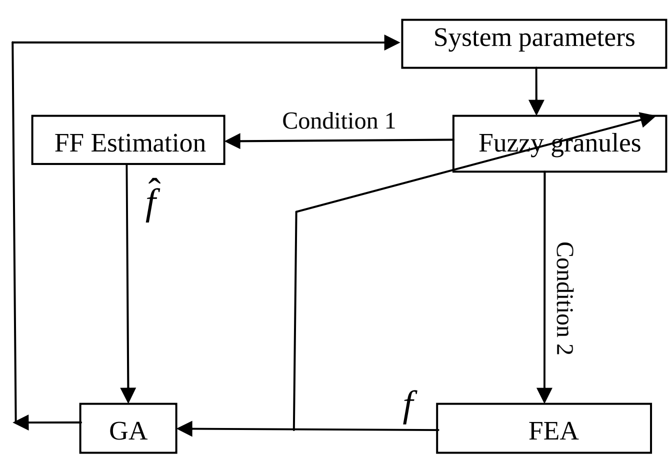
Step Seven: If the population size is not completed, repeat Steps Five to Seven.

Step Eight: Select parents using suitable selection operator and apply genetic operators, namely recombination and mutation, to create the new generation.

Step Nine: When termination/evolution control criteria are not met, then update σ_k using "Equation 2" and repeat Steps Five to Nine.

C. *How to Control the Length of Granule Pool?*

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



Condition 1: *Solution is similar to one of granules.*
Condition 2: *Solution is not similar to each of granules.*

Figure 1. The architecture of the proposed algorithm

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 words, 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*. At each table update, only N_G granules with the highest L_k index are kept, and the others are discarded. In [12] an example is provided to illustrate the competitive granule pool update law.

III. How To Determine The Width of the Membership Functions

As noted in [12], it is critically important to have accurate estimation of the fitness function of the individuals in the last generations and by the proposed method, this can be accomplished by controlling the width of the produced MFs. At the early steps of evolution, by choosing the Width of the Membership Functions (WMFs) relatively large, the algorithm accepts individuals with less degree of similarity as similar individuals. Therefore, fitness is computed more often by estimation/association to the granules. As the algorithm matures and reaches higher fitness valuations while also converging more, the width decreases and the similarity between individuals must be increased to be accepted as similar individuals. This prompts higher selectivity for granule associability and higher threshold for estimation. In other words, in the last generations, the degree of similarity between two individuals must be larger than the first generations to be accepted as similar individuals. This procedure ensures a fast convergence rate because of fast computation in the first steps and accurate estimation of fitness function in the last generations.

In [12], the widths of the produced MFs are determined using equation 2. Based on this equation, the combined effect of granule enlargement/shrinkage is in accordance to the granule fitness and it needs to adjust 2 parameters, namely β and γ. These parameters are problem dependent and it seems critical to set up a procedure in order to avoid this difficulty.

To achieve this desideratum, a fuzzy supervisor with three inputs has been employed. During AFFA search, the fuzzy logic controller observes the Number of Design Variables (NDV), Maximum Range of Design Variables (MRDV) and percentage of completed trials and specifies the WMFs. The first input is NDV. Range of input variables (RIV) is the second input; large NDV and MRDV needs big width in MFs and vice versa. Percent Completed Generations (PCG)

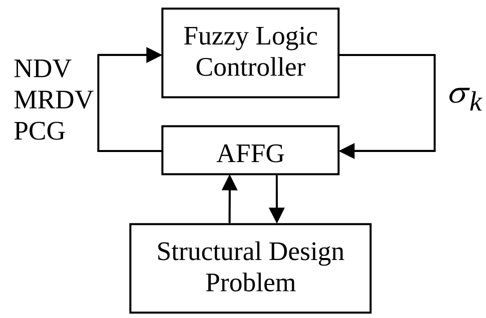


Fig. 2 Flowchart of the Purposed Fuzzy Controller

TABLE I
Fuzzy Rules of the First Controller

		NDV		
		Zero	Small	Big
MRDV	Zero	0	0.125	0.25
	Small	0.375	0.5	0.625
	Big	0.75	0.875	1

is the 3rd input, and is a number in the range [0, 1], where 1 signifies exhaustion of all allowed trials, and, consequently, a maturity of the search given a fixed amount of resources. The combined effect of granule enlargement/shrinkage in accordance to the PCG is to have fast computation in the first steps and accurate estimation of fitness function in the last generations.

The architecture for adaptive fuzzy control of the WMFs appears in Figure 2. Gaussian MFs are used for specification of the knowledge base of the fuzzy logic controller.

The knowledge base for control of the WMFs based on the above architecture has a large number of rules and the extraction of these rules is very difficult. Consequently the new architecture based on figure 3 is proposed in which the controller is separated in two controllers to diminish the complexity of rules. The first controller has two inputs (with three MFs in each, Zero(0, 0.3), Small(0.5, 0.3), Big(1.0,

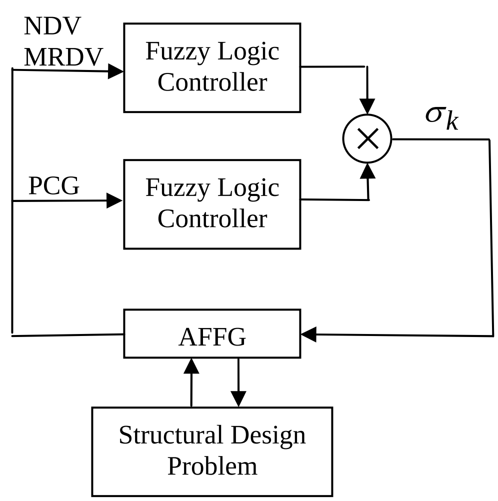


Fig. 3 Modified Flowchart of the Purposed Fuzzy Controller

0.3), the first number is the center and the second one is the spread) and the second controller has only one input which looks like a gain. The knowledge-base for the first controllers is shown in Table 1. The Gaussian MFs with equal width in each (0.3) are used for output. The second controller has just one Gaussian MF in which 0 and 1.25 are its center and spread respectively. The fuzzy controller which uses singleton fuzzifier, product inference engine and center average defuzzifier adjusts the σ_k after each generation.

IV. Empirical Results

To illustrate the efficacy of the proposed granulation techniques, we have chosen a set of 3 traditional optimization benchmarks (Table 2) namely: Griewank, Rastrigin and Ackley. These benchmark functions are scalable and are commonly used to assess the performance of optimization algorithms. For the first and the second one, the global minimum is $f(x)=0$ at $\{x_i\}^n=0$, and Ackley has a global minimum of $f(x)=0$ at $\{x_i\}^n=1$, for comparison purposes, four sets of input dimensions are considered; namely n = 5, 10, 20, 30.

The empirical study consisted in comparing the GA performance, as a function optimizer, AFFG (the parameters are used as Table III) and the proposed granulation techniques with fuzzy supervision (AFFG-FS).

Since the GA was used as a function optimizer, we chose roulette wheel with elitism as the selection method, in order to keep track of the best solution found.

The GA was implemented with 1-point crossover. The population size was set to 20 with the elite size of 2. The mutation and crossover rate used was 0.01 and 1.0, respectively. Ten runs of each experiment were executed.

A comparison was made with respect to the "Fast Evolutionary Strategy" (FES) in which a fitness and associated reliability values are assigned to each new individual. The fitness is truly evaluated if the reliability value is below a certain threshold. 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. Three different levels for T namely 0.5, 0.7, 0.9 are being used here as proposed in [9].

The report is given for the 5-D (dimension), 10-D, 20-D and 30-D scenarios. For AFFG, and AFFG-FS the number of individuals in granule pool is varied between 20, 20, 40 and 80 respectively. The reported results were obtained by achieving the same level of fitness evaluation for both the canonical GA and the proposed methods namely 500 for 5-D, 1000 for 10-D, 2000 for 20-D and 3000 for 3-D.

The average convergence trends of the standard GA, AFFG, AFFG-FG and FEA are summarized in Figures 4-6. All results presented were averaged over 10 runs. As shown in the Figures, the search performance of the AFFG and AFFG-FS are superior to the standard GA even with a small number of individuals in the granule pool. The Figures show that fitness inheritance has comparable performance when the number of dimension is small, but its performance deteriorates as the problem complexity increases.

We also studied the effect of varying the number of granules N_G on the convergence behavior of AFFG and AFFG-FS. Comparison can be made by the results obtained in Figure 7. It can be shown that AFFG and AFFG-FS are not significantly sensitive to N_G. However, further increase of N_G, slows down the rate of convergence duo to the imposed computational complexity.

TABLE II

List of Test benchmark Functions

Function	Formula
Griewangk	$1+\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$, $i=1:n$; $-600\le x_i\le 600$;
Rastrigin	$f(x)=10\cdot n+\sum_{i=1}^{n}(x_i^2-10\cdot\cos(2\cdot\pi\cdot x_i))$ $i=1:n$; $-5.12\le x_i\le 5.12$;
Akley	$f(x)=20+e-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i}$ $i=1:n$; $-32.768\le x_i\le 32.768$;

TABLE III

Parameters used for AFFG

Function	β	γ
Griewangk	0.00012	190
Rastrigin	0.004	0.15
Akely	0.02	0.25

V. Concluding Remarks

An intelligent guided technique via an adaptive fuzzy similarity analysis for fitness granulation is used to decide on the use of expensive function evaluation and dynamically adapt the predicted model. A fuzzy supervisor as auto-tuning algorithm is introduced in order to avoid tuning of parameters used in this approach.

A comparison is provided between FES and the proposed approach using 3 traditional optimization benchmarks with 4 different dimensions in each. Numerical results showed that the proposed technique is capable of optimising functions of varied complexity efficiently. Furthermore in comparison with our previous work, it can be shown that AFFG and AFFG-FS are not significantly sensitive to N_G, and a small N_G can still produce good results. Moreover, the auto-tuning of fuzzy supervisor removes the need for exact parameter determination without obvious influence on convergence speed.

ACKNOWLEDGMENT

The first author wishes to thank Dr. M. Khademi for his kind help and generous support. The third author acknowledges support from CONACYT project No. 45683-Y.

REFERENCES

[1] M. Olhofer, T. Arima, T. Sonoda, and B. Sendhoff. “Optimisation of a stator blade used in a transonic compressor cascade with evolution strategies”. In Adaptive Computation in Design and Manufacture, pages 45–54. Springer, 2000.

[2] Ratle. A., “Accelerating the convergence of evolutionary algorithms by fitness landscape approximation”, Parallel Problem Solving from Nature-PPSN V, Springer-Verlag, pp. 87-96, 1998.

[3] El-Beltagy M. A. and Keane A. J., “Evolutionary optimization for computationally expensive problems using Gaussian processes”, Proc. Int. Conf. on Artificial Intelligence (IC-AI'2001), CSREA Press, Las Vegas, pp. 708-714, 2001.

[4] 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.

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

[6] 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.

[7] Won, K. S. and Ray, T., “A Framework for Design Optimization using Surrogates”, Journal of Engineering Optimization, pp. 685-703, 2005.

[8] Vapnik V., “The Nature of Statistical Learning Theory”, Springer-Verlag, NY, USA, 1999.

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

[10] 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.

[11] Gunn S.R., “Support Vector Machines for Classification and Regression”, Technical Report, School of Electronics and Computer Science, University of Southampton, (Southampton, U.K.), 1998.

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

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

[14] Amir R. Khorsand, Mohammad R. Akbarzadeh, “Multi-objective meta level soft computing-based evolutionary structural design”, Journal of the Franklin Institute, vol. 9, Issue 5, pp. 595–612, August 2007.

[15] Gunn S.R., “Support Vector Machines for Classification and Regression”, Technical Report, School of Electronics and Computer Science, University of Southampton, (Southampton, U.K.), 1998.

[16] 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.

[17] Y. Jin and B. Sendhoff, “Reducing fitness evaluations using clustering techniques and neural network ensembles”, in GECCO’04, Springer, pp. 688-699, 2004.