A Novel General Framework for Evolutionary Optimization: Adaptive Fuzzy Fitness Granulation

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

Abstract— **Computational complexity is a major challenge in evolutionary algorithms due to their need for repeated fitness function evaluations. Here, we aim to reduce number of fitness function evaluations by the use of fitness granulation via an adaptive fuzzy similarity analysis. In the proposed algorithm, an individual's fitness is only computed if it has insufficient similarity to a queue 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 queue as a new fuzzy granule. The queue size as well as each granule's radius of influence is adaptive and will grow/shrink depending on the population fitness and the number of dissimilar granules. The proposed technique is applied to a set of 6 traditional optimization benchmarks that are for their various characteristics. In comparison with standard application of evolutionary algorithms, statistical analysis reveals that the proposed method will significantly decrease the number of fitness function evaluations while finding equally good or better solutions.**

I. Introduction

As the field of evolution-based algorithms matures and tackles more real-world applications, its limitations and challenges also become more clear. While nature is indeed the original inspiration as well as a 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, for an explicit fitness function may either be non existent or its computation is prohibitively costly. In both cases, 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, 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 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. A popular subclass of fitness function approximation methods known as fitness inheritance is introduced in [1] and [2] where fitness is simply inherited. Theoretical analyses of convergence time and population sizing when fitness is inherited is reported in [3]. An approach similar to fitness inheritance has also been suggested where the fitness of a child individual is the weighted sum of its parents [4]. Unfortunately, the performance of parents is not always indicative of the child, and this simple strategy can fail in sufficiently complex and multi-objective problems [5].

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 [10] and fitness estimation by association/friendship [25] 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.

Mohsen Davarynejad is M.Sc. student at Department of Electrical Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad 9177948944-1111, IR-IRAN, (phone +98-511-862-9540; fax: +98-511-862-9541; email: davarynejad@kiaeee.org). He is also working at Robotics and Intelligent Systems Laboratory of the stated department.

M.-R.Akbarzadeh-T. (Senior Member, IEEE) is with the Department of Electrical Engineering, Ferdowsi University of Mashhad. He is currently Visiting Scholar at BISC, UC Berkeley (e-mail: akbarzadeh@ieee.org).

N. Pariz is assistant professor with the same department (e-mail: n-pariz@um.ac.ir).

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 [15,16]. 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 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].

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], [9].

While the above methods aim to reduce computational cost by fitness function approximation, 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 converge to false optimums. Therefore, it can 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 [22] or evolution control in evolutionary computation [23]. 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 is not 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 peeks.

In this paper, fitness is not interpolated or estimated; rather the uncertainty in the similarity among real solutions is exploited. Nature's "survival of the fittest" is not about exact measures of fitness; rather it is about rankings among peers. By exploiting this natural tolerance for imprecision, we hope to preserve optimization performance by computing fitness only selectively and only to preserve this ranking among individuals in a given population. In the proposed algorithm, an adaptive queue 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 queue as a new fuzzy granule. In this fashion, irregardless 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 queue 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 queue from growing too large, granules that are not used are gradually eliminated.

This paper 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. A set of six conventional optimization benchmark problems are simulated in Section 3. 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) [26]

A. The Main Idea

The proposed fuzzy adaptive fitness granulation aims to minimize the number of exact fitness function evaluations by creating a queue 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 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 queue of dissimilar solutions/granules. When a new solution is introduced to this queue, granules compete by a measure of similarity to win the new solution and thereby to prolong their lives in the queue. In turn, the new individual simply assumes fitness of the winning (most similar) individual in this queue. 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 queue 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 queue. The proposed algorithm is shown in Figure 1.

As is shown in Figure 1, a random parent population $P_0 = \{X_1^1, X_2^1, ..., X_j^1, ..., X_t^1\}$ is initially created, where $X_j^i = \{x_{j,1}^i, x_{j,2}^i, ..., x_{j,r}^i, ..., x_{j,m}^i\}$ 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, $G = \{(C_k, \sigma_k, L_k) | C_k \in \Re^m, \sigma_k \in \Re, L_k \in \Re, k = 1,...,l\}$ 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. $X_1^1 = \{x_{1,1}^1, x_{1,2}^1, ..., x_{1,r}^1, ..., x_{1,m}^1\}$ is chosen as the center $C_1 = \{c_{1,1}, c_{1,2}, ..., c_{1,r}, ..., c_{1,m}\} = X_1^1$ of first granule. The membership function $\mu_{r,k}$ therefore describes a Gaussian similarity neighborhood for each parameter k as follows,

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

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

Then, the average similarity of a new solution $X_j^i = \{x_{j,1}^i, x_{j,2}^i, ..., x_{j,r}^i, ..., x_{j,m}^i\}$ 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 exact fitness function computing or estimated by associating it to one of the granules in the queue if there is a granule in the queue with higher similarity to X_j^i than a predefined threshold, as follows.

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

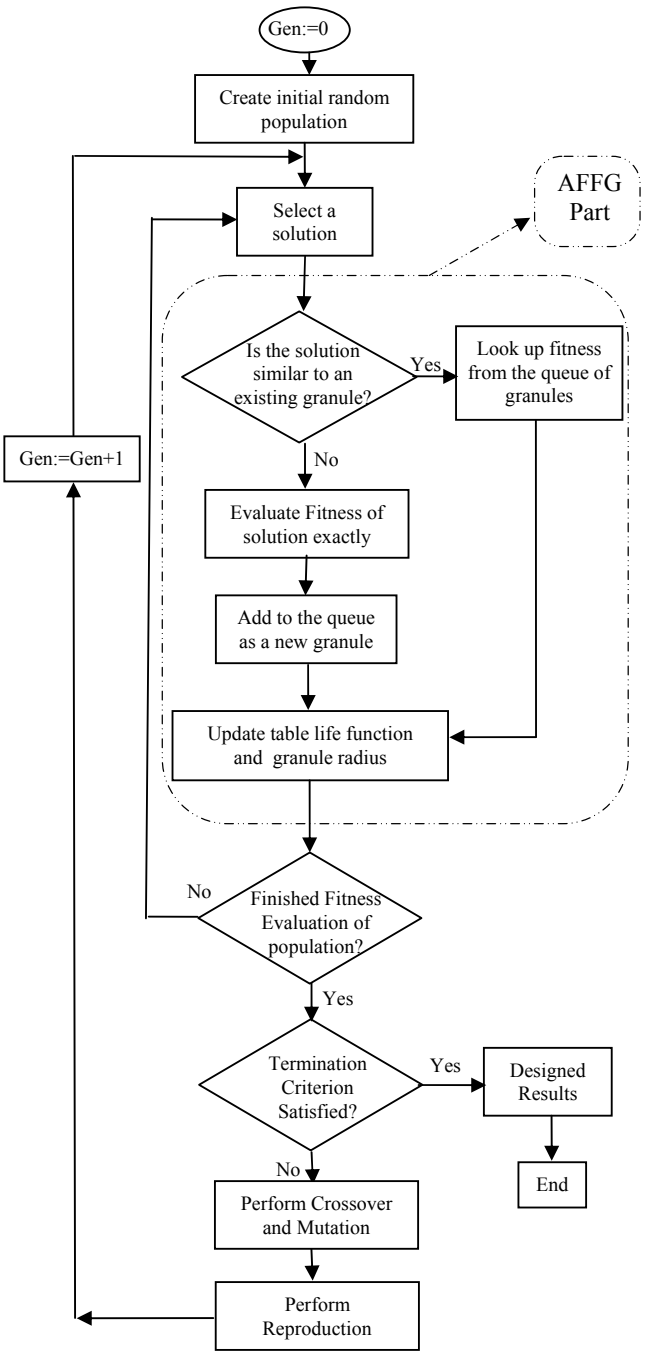

Fig. 1. Flowchart of the Purposed AFFG Algorithm

where $K = \underset{k \in \{1,\ 2,\ ...,\ l\}}{index \max} \left\{ \overline{\mu}_{j,k} \right\}$,

$\theta^i = \alpha . \dfrac{Max\left\{ f(X_1^{i-1}),\ f(X_2^{i-1}), ... f(X_t^{i-1}) \right\}}{\bar{f}^{i-1}}$, $\bar{f}^i = \sum_{j=1}^{t} \dfrac{f\left(X_j^i\right)}{t}$,

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 $\max_{k \in \{1,2,...,l\}} \left\{ \overline{\mu}_{j,k} \right\} < \theta^i$, 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.

$$\sigma_k = \frac{1}{\left(e^{f(C_k)}\right)^{\beta}} \qquad (2)$$

Where $\beta > 0$ is an emphasis operator. 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 queue 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 the last generations, the degree of similarity between two individuals must be larger than the first 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 latter generations.

C. Adaptation in the Length of Granule Queue

Finally, as the evolutionary algorithm proceeds, it is inevitable that new granules are increasingly generated and added to the queue. Depending on complexity of the problem, the size of this queue 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 queue. 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. Here we set M = 5.

The following example is provided to illustrate the competitive granule queue update.

Example: Suppose there are three granules 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 $\theta^i < \max_{\substack{k \in \{1,\ 2,\ ...,\ l\} \\ j=1}} \left\{ \overline{\mu}_{j,k} \right\}$, 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 $\theta^i > \max_{\substack{k \in \{1,\ 2,\ ...,\ l\} \\ j=1}} \left\{ \overline{\mu}_{j,k} \right\}$ 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.

III. Numerical Results

To illustrate the efficacy of the proposed granulation techniques, we have chosen a set of 6 traditional optimization benchmarks for their various characteristics. We need a test environment in order to prove the performance of the proposed method. De Jong [24] proposed that 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.

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

Table II. GA & GA-AFFG Comparison. Opt means average of optimum solution in fifteen separate runs. As described before if p-value ≥ α then there is not a statistically significant difference between two methods.

	Opt. Method	GA	AFFG
Function			
De Jong's 1 *F1*	No. FFE	2000	659.4
	OPT	75.121	74.497
	p -value	-	0.4531
De Jong's 4 *F2*	No. FFE	2000	763.3
	OPT	72.233	72.168
	p -value	-	0.7726
Michalewicz *F3*	No. FFE	2000	716.5
	OPT	7.80	7.770
	p -value	-	0.6352
Rastrigins *F4*	No. FFE	2000	989.86
	OPT	-61.9	-63.56
	p -value	-	0.3952
Schwefel's *F5*	No. FFE	2000	892.03
	OPT	6027	5927
	p -value	-	0.4504
Griewangk's *F6*	No. FFE	2000	848.20
	OPT	1638	1566.6
	p -value	-	0.3018

rejected. Here, the significance level α was assigned, and the p-value was calculated for each of the following applications.

The GA routines utilized random initial populations, binary-coded chromosomes, single-point crossover, mutation, fitness scaling, and an elitist stochastic universal sampling selection strategy. The probabilities of crossover ($P_{CROSSOVER}$ = 1) and mutation ($P_{MUTATION}$ = 0.01), and the population size (Number of chromosomes = 20) in each generation, and (Number of generation = 100). Finally chromosome length varies depending on the number of variables in a given problem but each variable is allocated 8 bits.

Comparison results are illustrated in Table II. The GA and GA-AFFG is run 15 times for each of the above 6 functions. While the two evolutionary schemes GA and GA-AFFG's reach statistically similar performance in terms of optimal fitness, the proposed technique reduces number of function evaluations by over 50%.

IV. 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.

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 50 percent of individuals in each generation. The Traditional Optimization problem simulations are standard optimization benchmarks chosen for their various features such as multimodality and nonlinearity. Statistical analysis confirms that the proposed method demonstrates an ability to reduce computation without sacrificing performance.

V. Acknowledgements

Authors wish 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] K. Sastry, D. Goldberg, and M. Pelikan, "Don't evaluate, inherit," in Proc. Genetic Evol. Comput. Conf., 2001, pp. 551–558.

[4] M. Salami and T. Hendtlass, "A fast evaluation strategy for evolutionary algorithms," Appl. Soft Comput., vol. 2, 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, 2003, pp. 31–42.

[6] 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.

[7] Y. Jin, "Acomprehensive survey of fitness approximation in evolutionary computation," Soft Comput., vol. 9, no. 1, pp. 3–12, 2005.

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

[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, 2004, pp. 688–699.

[10] B. Whitehead, "Genetic evolution of radial basis function coverage using orthogonal niches," IEEE Trans. Neural Netw., 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, 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, 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, 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] 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, pages 663–670. IEEE Press, 2003.
[20] H. Ulmer and F. Streichert and A. Zell, “Evolution strategies with controlled model assistance”, Congress on Evolutionary Computation, 1569--1576, 2004
[21] Yaochu Jin. “A comprehensive survey of fitness approximation in evolutionary computation”. Soft Computing, 9(1), 3-12, 2005
[22] 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
[23] Yaochu Jin, Markus Olhofer, Bernhard Sendhoff: "On Evolutionary Optimization with Approximate Fitness Functions". GECCO 2000: 786-793.
[24] K. A. De Jong, “An Analysis of the Behavior of a Class of Genetic Adaptive Systems”, Ph.D. dissertation, University of Michigan, 1975.
[25] 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.
[26] Mohsen Davarynejad, "Fuzzy Fitness Granulation in Evolutionary Algorithms for complex optimization", M.Sc. Thesis, Ferdowsi University of Mashhad/Iran, Department of Electrical Engineering, 2007.

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

	$c_{k,1}$	$c_{k,2}$	$c_{k,3}$	$c_{k,4}$	$f(G_k)$	σ_k	L_k
G_1, 1st granule	1	1	1	1	6	0.5488	3
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.

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

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

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