Multiobjective Cellular Genetic Algorithm with Adaptive Fuzzy Fitness Granulation

Iman Kamkar[1] Mohammad-R. Akbarzadeh-T[1]
[1]Department of Artificial Intelligence, Islamic Azad University, Mashhad Branch, Mashhad, Iran
Emails: Kamkar.iman@gmail.com, Akbarzadeh@ieee.org

Abstract— **Computational complexity is a major challenge in evolutionary algorithms due to their need for repeated fitness function evaluations. In the context of multiobjective evolutionary algorithms, there are a few attentions to the computational complexity of this kind of algorithms. Here, we aim to reduce number of fitness function evaluations in multiobjective cellular genetic algorithms 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 method is applied to a set of 6 test problems. In comparison with two well-known multiobjective evolutionary algorithms, NSGA-II, and MoCell, computational results show that the proposed method is competitive with these algorithms.**

I. Introduction

Many real world problems are multiobjective problems. As the optimized goals in multiobjective problems are generally conflicting and competing, these problems become more complex than single objective problems. Unlike single objective problems, multiobjective problems do not restrict to find a unique single solution therefore for solving this kind of problems we should find a set of solutions called *non-dominated solution* (also called *Pareto-optimal solutions*). When these solutions are plotted in the objective space they are called *Pareto front*. The main goal of multiobjective optimization is obtaining the Pareto front of a given multiobjective optimization problem (MOP). As search spaces in MOPs are very large, and time complexity for solving these problems by deterministic techniques is high, stochastic methods have been proposed for this kind of problems. Among them, evolutionary algorithms (EAs) have been investigated by many researchers, and some of the most well-known algorithms for solving MOPs belong to this class (e.g. NSGA-II [1], PAES [2], SPEA2 [4], and MoCell [3]).

Although all of these algorithms can tackle most real world applications, but they have some limitations. Fitness function evaluation is often the most prohibitive and limiting segment of artificial evolutionary algorithms, for an explicit fitness function may either be nonexistent 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 some problems such as design of mechanical structures, 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. 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 [5] and [6] where fitness is simply inherited. Theoretical analyses of convergence time and population sizing when fitness is inherited is reported in [7]. An approach similar to fitness inheritance has also been suggested where the fitness of a child individual is the weighted sum of its parents [8]. 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 [9].

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 [11] 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 [12], and the Kriging model that is also called the Design and Analysis of Computer Experiments (DACE) model [15]. In Kriging model, a global polynomial approximation is combined with a local gaussian process and the maximum likelihood method is used for parameter estimation.

Here, we use the method proposed in [13], for fitness function approximation in cellular genetic algorithms for solving multiobjective optimization problems.

This method reduces the number of fitness function evaluations by the use of fitness granulation via an adaptive

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

fuzzy similarity analysis. Here, 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.

II. ADAPTIVE FUZZY FITNESS GRANULATION

Adaptive fuzzy fitness granulation (AFFG) 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. The method 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. Basics of AFFG algorithm are as follows:

1. The first generation of individuals $P_0 = \{X_1^1, X_2^1, ..., X_j^1, ..., X_t^1\}$ is determined. Where $X_j^i = \{x_{j,1}^i, x_{j,2}^i, ..., x_{j,r}^i, ..., x_{j,m}^r\}$ 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 function variables. Also $G = \{(C_k, \sigma_k, L_k) | \quad C_k \in R^m, L_k \in R, k = 1,...,l\}$ is a set of fuzzy granules that is initially empty, where C_k is an m dimensional vector of centers, σ_k is the width of membership functions of the k-th fuzzy granules and L_k is the index of granule's life.

2. Center of first granule is equal to the phenotype of the first chromosome, i.e. $C_1 = \{c_{1,1}, c_{1,2}, ..., c_{1,r}, ..., c_{1,m}\} = X_1^1$.

3. Then the membership function $\mu_{r,k}$ describes a Gaussian similarity neighborhood for each parameter *k* as follows:

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

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

4. 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 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.

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

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

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

, $\bar{f}^i = \sum_{j=1}^{t} \frac{f(X_j^i)}{t}$, $\alpha > 0$ is a constant of probability. 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 $\max_{k \in \{1,2,...,l\}} \{\overline{\mu}_{j,k}\} < \theta^i$, X_j^i is chosen as newly created granule.

Adaptation in the Width of Membership Functions

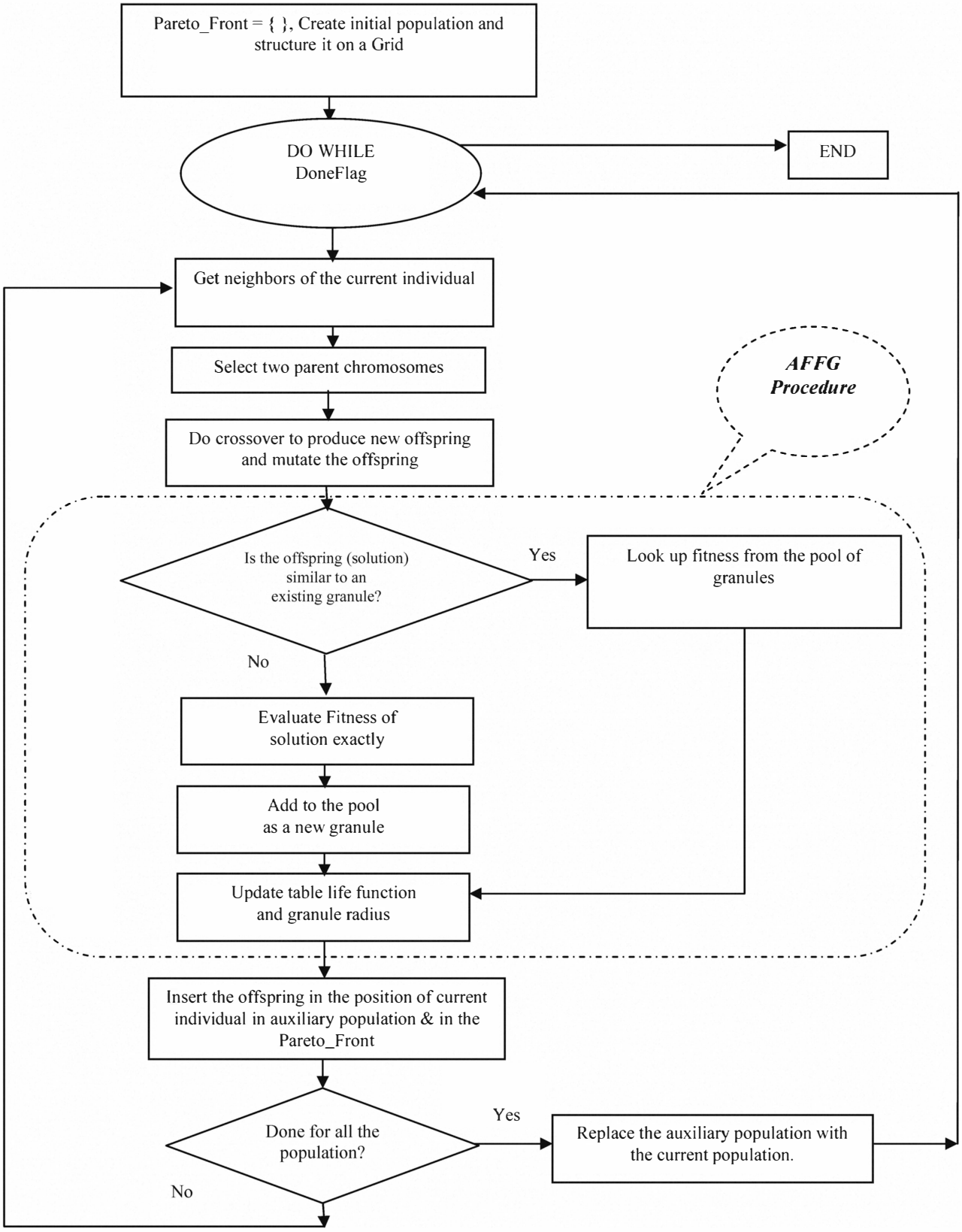

Fig 1. Flowchart of the CGA-AFFG algorithm

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.

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

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.

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,

$$L_k = \begin{cases} L_k + M & if \quad k = K \\ L_k & otherwise \end{cases} \qquad (3)$$

Table 1.Unconstrained test function

Problem	n	Variable bounds	Objective functions
ZDT 1	30	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1 / g(x)}\right]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT 2	30	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - (x_1 / g(x))^2\right]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT 3	30	[0,1]	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1 - \sqrt{x_1 / g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$

Table 2.Constrained test function

Problem	n	Variable bounds	Objective functions	Constraints	
Tanaka	2	$x_i \in [-\pi, \pi]$	$f_1(x) = x_1$ $f_2(x) = x_2$	$-x_1^2 - x_2^2 + 1 + 0.1\cos(16\arctan(x_1 / x_2)) \le 0$ $(x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 \le \frac{1}{2}$	
ConstrEx	2		$x_1 \in [0.1, 1.0]$ $x_2 \in [0, 5]$	$f_1(x) = x_1$ $f_2(x) = (1 + x_2) / x_1$	$x_2 + 9x_1 \ge 6$ $-x_2 + 9x_1 \ge 1$
Srinivas	2		$x_i \in [-20, 20]$	$f_1(x) = (x_1 - 2)^2 +$ $(x_2 - 1)^2 + 2$ $f_2(x) = 9x_1 - (x_2 - 1)^2$	$x_1^2 + x_2^2 \le 225$ $x_1 - 3x_2 \le -10$

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.

III. CELLULAR GENETIC ALGORITHM WITH ADAPTIVE FUZZY GRANULATION (CGA-AFFG)

Flowchart of the proposed method is shown in the Figure 1. CGA-AFFG starts by creating an empty front, and individuals are arranged on a toroidal grid. For each individual the algorithm consists of selecting two parents from its neighborhood, recombining them in order to obtain an offspring, mutating it, and evaluating the resulting offspring. In order to evaluate the offspring, we use AFFG method discussed in the previous section. After that, the resulting individual if is not dominated by the current individual is inserted in both the auxiliary population and in the Pareto front. Finally, after each generation, the old population is replaced by the auxiliary one.

Table 3- Mean and standard deviation of the convergence metric *GD*

Problem	Opt. method	NSGA II	MoCell	CGA-AFFG
ZDT1	No.FFE	25000	25000	2465.3354
	Mean and Std.	2.176e-4± 3.53e-5	4.087e-4±6.45e-5	4.165e-4±4.76e-5
ZDT2	No.FFE	25000	25000	4375.7689
	Mean and Std.	1.765e-4± 3.56e-5	2.432e-4±8.36e-5	2.563e-4±4.54e-5
ZDT3	No.FFE	25000	25000	3879.787
	Mean and Std.	2.207e-4 ±3.62e-5	2.535e-4 ± 2.81e-5	2.641e-4±3.52e-5
Tanaka	No.FFE	25000	25000	7534.6470
	Mean and Std.	1.177e-3 ±7.54e-5	7.587e-4 ± 7.15e-5	1.354e-4±6.87e-5
ConstrEx	No.FFE	25000	25000	2367.3542
	Mean and Std.	2.867e-4 ±3.28e-5	1.960e-4 ± 2.52e-5	2.935e-4±2.37e-5
Srinivas	No.FFE	25000	25000	6989.6482
	Mean and Std.	1.907e-4 ±3.12e-5	5.165e-5 ± 1.65e-5	2.107e-4±2.98e-5

Table 4- Mean and standard deviation of the convergence metric *Delta*

Problem	Opt. method	NSGA II	MoCell	CGA-AFFG

ZDT1	No.FFE	25000	25000	2465.3354
	Mean and Std.	3.125e-1± 2.98e-2	1.176e-1± 1.27e-2	3.342e-1±2.54e-2
ZDT2	No.FFE	25000	25000	4375.7689
	Mean and Std.	3.544e-1±3.71e-2	1.132e-1± 2.12e-2	4.012e-1± 2.98e-2
ZDT3	No.FFE	25000	25000	3879.787
	Mean and Std.	7.514e-1± 2.15e-2	7.087e-1± 1.24e-2	7.546e-1±2.31e-2
Tanaka	No.FFE	25000	25000	7534.6470
	Mean and Std.	7.176e-1± 3.45e-2	6.624e-1± 3.18e-2	8.152e-1± 3.26e-2
ConstrEx	No.FFE	25000	25000	2367.3542
	Mean and Std.	4.252e-1± 3.24e-2	1.423e-1± 2.04e-2	4.126e-1±3.26e-2
Srinivas	No.FFE	25000	25000	6989.6482
	Mean and Std.	3.265e-1± 2.91e-2	6.198e-2± 1.02e-2	3.362e-1± 3.21e-2

IV. COMPUTAIONAL RESULTS

In this section the proposed algorithm is evaluated. For that, 6 test problems have been chosen which were taken from the specialized literature, and, in order to assess how competitive CGA-AFFG is, we compared it to NSGA II and MoCell. These test problems are depicted in table 1 and 2. The parameters which are used by the proposed algorithm are as follows. We have organized 100 individuals on a square toroidal grid and the neighborhood defined on it always contains 5 individuals: the considered one plus the North, East, West, and South individuals. We have used arithmetic crossover and uniform mutation. Crossover and mutation rates are p_c=1 and p_m= 0.05.

Since multi-objective algorithms find a set of non-dominated solutions with respect to multiple objectives (not a single final solution with respect to a single objective), the comparison between different multi-objective algorithms is not easy. There are two goals in a multi-objective optimization: 1) convergence to the Pareto-optimal set and 2) maintenance of diversity in solutions of the Pareto-optimal set. These two tasks cannot be measured adequately with one performance metric. Here, we use two different metrics for evaluating each of the above two goals in a solution set obtained by a multi-objective optimization algorithm.

- **Generational Distance** This metric was introduced by Van Veldhuizen and Lamont [14] for measuring how far the elements are in the set of non-dominated vectors found so far from

  those in the Pareto optimal set, and it is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^{\,2}}}{n} \qquad (4)$$

  Where n is the number of vectors in the set of non-dominated solutions, and d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the Pareto optimal set.

- **Delta** This metric [8] is a diversity metric that measures the extent of spread achieved among the obtained solutions. This metric is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}} \qquad (5)$$

  Where, d_i is the Euclidean distance between consecutive solutions in the obtained non-dominated set of solutions. $\bar{d}$ is the mean of these distances, and d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set.

The results which are summarized in Table 3 and 4, shows generational distance (GD) and delta (Δ) respectively. For each problem, 30 independent runs were carried out, and the tables include the mean, standard deviation and number of fitness function evaluation of the results.

As illustrated in Tables 3 and 4, the CGA-AFFG is competitive with NSGA II and MoCell. By the less

number of fitness function evaluation; the proposed method illustrated admissible Pareto optimal solutions.

V. CONCLUSION

Cellular genetic algorithms are one of promising evolutionary algorithms in solving many optimization algorithms. But due to their nature, they need repeated fitness function evaluation and so they are time consuming. In order to overcome this problem, an adaptive fuzzy fitness granulation is used. This method selectively reduces number of fitness function evaluations without approximating or on-line training. This 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. The simulations show that the proposed method could lead to improvement in computation time without sacrificing performance.

REFERENCES

[1] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elist Multi-objective Genetic Algorithm: NSGA-II. *IEEE TEC*, 6(2):182{197, 2002.

[2] J. Knowles and D. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 9{105, Piscataway, NJ, 1999. IEEE Press.

[3] A.J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. A Cellular Genetic Algorithm for Multiobjective Optimization. *International Journal of Intelligent Systems Volume 24 Issue 7, Pages 726 – 746-2009*

[4] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), 2001.

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

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

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

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

[9] 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.

[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] 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.

[12] R. Myers and D. Montgomery. Response Surface Methodology. John Wiley & Sons, Inc., New York, 1995. 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] M.-R.Akbarzadeh-T., M.Davarynejad and N.Pariz, "Adaptive fuzzy fitness granulation for evolutionary optimization," *International Journal of Approximate Reasoning Volume 49 Issue 3, Pages 523-538-2008*

[14] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective Evolutionary Algorithm Research: AHistory and Analysis. *Technical Report TR-98-03, Dept. Elec. Comput. Eng., Air Force Inst.Technol., Wright-Patterson, AFB, OH, 1998.*

[15] J. Sacks,W. J.Welch, T. J. Michell, and H. P.Wynn. Design andanalysis of computer experiments. Statistical Science, 4:409–435,1989.