Evolutionary Hidden Information Detection by Granulation-Based Fitness Approximation

M. Davarynejad, C. W. Ahn, J. Vrancken, J. van den Berg, C. A. Coello Coello

Abstract— Spread spectrum audio watermarking (SSW) is one of the most powerful techniques for secure audio watermarking. SSW hides information by spreading the spectrum. The hidden information is called the 'watermark' and is added to a host signal, making the latter a watermarked signal. The spreading of the spectrum is carried out by using a pseudo-noise (PN) sequence. In conventional SSW approaches, the receiver must know both the PN sequence used at the transmitter and the location of the watermark in the watermarked signal for detecting the hidden information. This method has contributed much to secure audio watermarking in that any user, who is not able to access this secrete information, cannot detect the hidden information. Detection of the PN sequence is the key issue of hidden information detection in SSW. Although the PN sequence can be reliably detected by means of heuristic approaches, due to the high computational cost of this task, such approaches tend to be too computationally expensive to be practical. Evolutionary Algorithms (EAs) belong to a class of such approaches. Most of the computational complexity involved in the use of EAs arises from fitness function evaluation that may be either very difficult to define or computationally very expensive to evaluate. This paper proposes an approximate model, called *Adaptive Fuzzy Fitness Granulation with Fuzzy Supervisor* (AFFG-FS), to replace the expensive fitness function evaluation. First, an intelligent guided technique via an adaptive fuzzy similarity analysis for fitness granulation is used for deciding on the use of exact fitness function and dynamically adapting the predicted model. Next, in order to avoid manually tuning parameters, a fuzzy supervisor as auto-tuning algorithm is employed. Its effectiveness is investigated with three traditional optimization benchmarks of four different choices for the dimensionality of the search space. The effect of the number of granules on the rate of convergence is also studied. The proposed method is then extended to the hidden information detection problem to recover a PN sequence with a chip period equal to 63, 127 and 255 bits. In comparison with the standard application of EAs, experimental analysis confirms that the proposed approach has an ability to considerably

Mohsen Davarynejad is Ph.D. researcher with the Faculty of System Engineering, Policy analysis and Management, Delft University of Technology, 2600GA, Delft, The Netherlands. (email: m.davarynejad@tudelft.nl).

Chang Wook Ahn is with the School of Information and Communication Engineering, Sungkyunkwan University, Korea. (email: cwan@skku.edu).

Jos Vrancken is Associate Professor with the Faculty of System Engineering, Policy analysis and Management, Delft University of Technology, 2600GA, Delft, The Netherlands. (email: j.l.m.vrancken@tudelft.nl).

Jan van den Berg is Associate Professor with the Faculty of System Engineering, Policy analysis and Management, Delft University of Technology, 2600GA, Delft, The Netherlands. (email: j.vandenberg@tudelft.nl).

Carlos A. Coello Coello 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).

reduce the computational complexity of the detection problem without compromising performance. Furthermore, the auto-tuning of the fuzzy supervisor removes the need of exact parameter determination.

Keywords: Watermarked Signal, Fast Hidden Information Detection, Fuzzy Granulation, Fitness Approximation, Evolutionary Algorithms.

1. INTRODUCTION

In recent years, digital watermarking has received due attention from the security and cryptography research communities. Digital watermarking is a technique to hide information into an innocuous-looking media object, which is called ‘host,’ so that no one can suspect the existence of hidden information. It is intended to provide a degree of copyright protection as use of digital media mushrooms [1]. Depending on the type of the host signal to cover hidden information, watermarking is classified into *image watermarking* and *audio watermarking*. In this paper, we focus our attention on audio watermarking but the approach can be applied to image watermarking as well.

Numerous audio watermarking techniques have been proposed and the most important ones being Least Significant Bits (LSB) [2], Phase coding [3], Echo hiding [4] and spread spectrum watermarking (SSW) [5]. The latter, SSW, is known as the most promising watermarking method due to its high robustness against noise and high perceptual transparency. The main idea of SSW is to add the spread spectrum of hidden information to the spectrum of the host signal. Spreading the spectrum of the hidden information is performed by means of a pseudo-random noise (PN) sequence.

Detection of hidden information from the received watermark signal is performed using the exact PN sequence adopted for spreading the spectrum of hidden information. Therefore, the receiver should have access to the PN sequence for detection. This essential, private knowledge results in a highly secure transmission of information against any unauthorized user who does not have access to the PN sequence and the location of the watermark. Hence, the PN sequence can be regarded as a secret key which is shared between the transmitter and the receiver.

In [6], genetic algorithms (GAs) have been presented for detecting hidden information, even though the receiver has no prior knowledge on the transmitter’s spreading sequence. However, iterative fitness function evaluation for such a complex problem is often the most prohibitive and limiting segment of this approach. For the problem of recovering the PN sequence, sequences with different periods have different converging times. In the study reported in [6], it has been shown that converging time increases exponentially as the period of the PN sequence increases. So, the approach fails by losing the validity of information. The greater the PN sequence is, the more difficult is the situation for recovering the PN sequence and the more secure SSW will result. Note hereby that a greater period of the PN sequence decreases the capacity of the SSW algorithm for embedding hidden information. To alleviate the problem of exponentially increasing converging times, a variety of techniques for constructing approximation models – often referred to as *metamodels* – have been proposed [7]-[14]. For computationally expensive optimization problems such as the detection of hidden information, it may be necessary to strike a balance between exact fitness evaluation and approximate fitness evaluation. A popular subclass of fitness function approximation methods is fitness inheritance where fitness is simply transmitted (or "inherited") [7, 8]. A similar approach named "Fast Evolutionary Strategy" (FES) has also been suggested in [9], in which the fitness of a child individual is the weighted sum of its parents. In that approach, fitness and associated reliability values are assigned to each new individual, and then the actual fitness function is only evaluated when the reliability value is below a certain threshold. Further, Reyes Sierra and Coello Coello [17] incorporated the concept of fitness inheritance into a multi-objective particle swarm optimizer to reduce the number of fitness evaluations [17]. In [18], they tested their approach on a well-known test suite of multi-objective optimization problems. They generally reported lower computation cost, while the quality of their results improved in higher dimensional spaces. However, as also shown in [19] as well as in this paper, the performance of parents may not be a good predictor of their children for sufficiently complex and multiobjective problems in rendering fitness inheritance inappropriate under such circumstances.

Other common approaches based on learning and interpolation from known fitness values of a small population, (e.g. low-order polynomials and least square estimations [10], artificial neural networks (ANN), including multi-layer perceptrons [11] and radial basis function networks [12], support vector machines (SVM) [13], regression models [14], etc.) can also be employed.

In 1979, Zadeh [28] developed fuzzy information granulation as a technique by which a class of points (objects) is partitioned into granules, with a granule being a clump of objects drawn together by indistinguishability, similarity, and/or functionality. The fuzziness of granules and their attributes is characteristic of the ways by which human concepts and reasoning are 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 in mathematics.

In this paper, with a view to reducing computational cost, we employ the concept of fuzzy granulation to effectively approximate the fitness function in evolutionary algorithms (EAs). In other words, the concept of fitness granulation is applied to exploit the natural tolerance of EAs in fitness function computations. Nature's "survival of the fittest" does not necessarily mean exact measures of fitness; rather it is about rankings among competing peers [29]. By exploiting this natural tolerance for imprecision, optimization performance can be preserved through computing fitness only selectively based on the ranking among individuals in a given population. Unlike existing approaches, the fitness values are not interpolated or estimated; rather the similarity and indistinguishability among real solutions is exploited.

In the proposed algorithm as explained in details in [15, 16] and called *adaptive fuzzy fitness granulation* (AFFG), 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, the 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 self-adaptively grow or 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 then gradually shrunks in the course 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 efficiently detect hidden information from spread spectrum watermarked signals. Finally, a fuzzy supervisor is developed for adaptively, automatically adjusting system parameters.

The paper is organized as follows: Section 2 presents the framework of adaptive fuzzy fitness granulation (AFFG). An auto-tuning strategy for determining width of membership functions (MFs) is also presented in the section; by which the need of exact parameter setting is eliminated, without affecting the rate of convergence. This approach is called *adaptive fuzzy fitness granulation with fuzzy supervisory* (AFFG-FS). In section 3, the proposed algorithm is tested on three traditional optimization benchmarks with four different dimensions. In Section 4, the recovery of the PN sequence from a received watermarked signal using the proposed approach is illuminated. Some supporting simulation results and discussion thereof are also presented in the section. Finally, conclusions are drawn in Section 5.

2. The AFFG Framework [15]

Adaptive fuzzy fitness granulation (AFFG) was first proposed in [15]. It includes a global model of a genetic algorithm (GA) which is hybridized with a fuzzy granulation (FG) tool (see Figure 1). The expensive fitness evaluation of individuals required in traditional GA, can be partially replaced by an approximation model. Explicit control strategies are used for evolution control, leading to a considerable speedup without compromising heavily on the solution accuracy. While the approximation techniques themselves are widely known for accelerating the iterative optimization process, the focus of AFFG lies in promoting controlled speedup in view of avoiding detrimental effects of the approximation. The following section presents the main elements of the AFFG framework.

A. Basic Idea

The proposed adaptive fuzzy fitness granulation aims to minimize the number of exact fitness function (FF) evaluations by maintaining a pool of solutions (fuzzy granules) by which can be used to approximate solutions in further stages of the evolutionary process. 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., if their similarity is below a certain threshold), the new individual is instead added to the pool after its exact fitness is evaluated by the actual fitness function. Finally, granules that cannot win new individuals are gradually eliminated in order to avoid consistent growth of the pool. The basic idea of the proposed

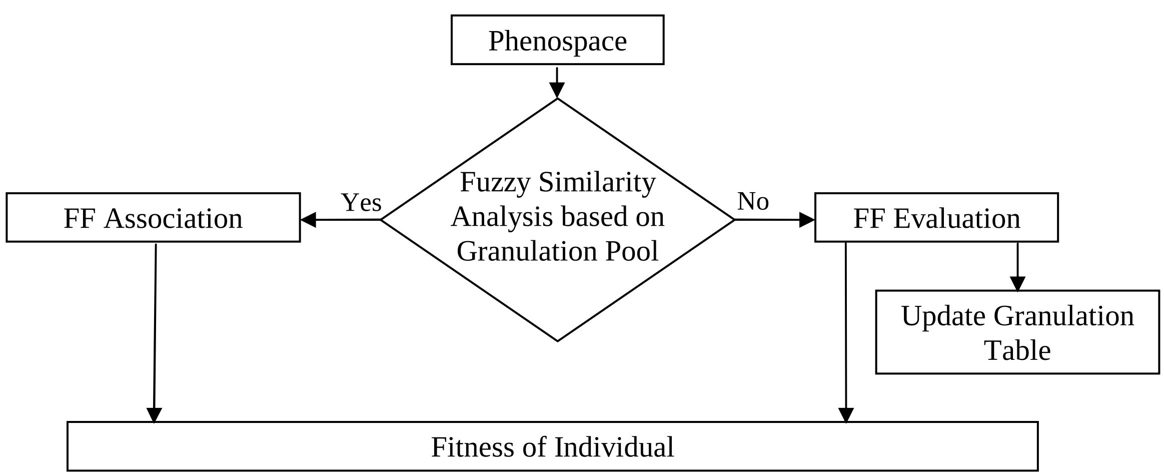


Figure 1. The architecture of the proposed algorithm

algorithm is graphically shown in Figure 1 and is discussed in more detail in the next section. For even more details, we refer to [15, 16].

B. Basic Algorithm Structure

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

Step 2: Define a multi-set *G* of fuzzy granules (C_k, σ_k, L_k) according to $G = \{(C_k, \sigma_k, L_k) \mid C_k \in \Re^m, \sigma_k \in \Re, L_k \in \Re, k = 1,...,l\}$; G is initially empty (i.e., $l = 0$). 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 3: Choose the phenotype of first chromosome ($X_1^1 = \{x_{1,1}^1, x_{1,2}^1, ..., x_{1,r}^1, ..., x_{1,m}^1\}$) as the center of the first granule ($C_1 = \{c_{1,1}, c_{1,2}, ..., c_{1,r}, ..., c_{1,m}\} = X_1^1$).

Step 4: Define the membership function $\mu_{r,k}$ by a Gaussian similarity neighborhood function for each parameter *k* according to

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

where *l* is the number of fuzzy granules.

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

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

where $\beta > 0$ is an emphasis operator and γ a constant of proportion. 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 in practical, number of trials is needed to adjust these parameters. This trial is based on a simple rule with respect the acceleration of the parameter optimization procedure: high speed needs to have enlargement in the granule spread and, as a consequence of this, less accuracy in fitness approximation, *vice versa*. To deal with this rule, a fuzzy controller with three inputs is adopted (see Section 2.D).

Step 5: 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 6: Either calculate the exact fitness function (FF) of X_j^i or estimate the FF by associating it to one of the granules in the pool in case there is a granule in the pool with higher similarity to X_j^i than a predefined threshold, i.e.:

$$f(X_j^i) = \begin{cases} f(C_K) & \text{if} \quad \underset{k \in \{1,\ 2,\ \ldots,\ l\}}{Max} \{\overline{\mu}_{j,k}\} > \theta^i \\ f(X_j^i) & \qquad\qquad otherwise \end{cases} \tag{3}$$

where $f(C_K)$ is the FF of the fuzzy granule, $f(X_j^i)$ is the real fitness calculation of the individual, $\theta^i = \alpha . \frac{Max\{f(X_1^{i-1}),\ f(X_2^{i-1}), \ldots f(X_t^{i-1})\}}{\bar{f}^{i-1}}$, $K = \underset{k \in \{1,\ 2,\ \ldots,\ l\}}{\arg Max}\{\overline{\mu}_{j,k}\}$, $\bar{f}^i = \sum_{j=1}^{t} \frac{f(X_j^i)}{t}$ and $\alpha > 0$ is a constant of proportionality that is usually set at 0.9 unless otherwise indicated. The threshold θ^i increases as the best individual's fitness in generation i increases. As the population matures and reaches higher fitness values (i.e., while 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 $\underset{k \in \{1,2,\ldots,l\}}{Max}\{\overline{\mu}_{j,k}\} < \theta^i$, X_j^i is chosen as a newly created granule.

Step 7: If the population size is not completed, repeat **Step 5** to **Step 7**.

Step 8: Select parents using suitable selection operator and apply genetic operators namely recombination and mutation to create new generation.

Step 9: When termination/evolution control criteria are not met, then update σ_k using eqn. (2) and repeat **Step 5** to **Step 9**.

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

As the evolutionary procedures proceed, it is inevitable that new granules are 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 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 to *N* and subsequently updated as below,

$$L_k = \begin{cases} L_k + M & if \quad k = K \\ L_k & Otherwise \end{cases} \tag{4}$$

where *M* is the life reward of the granule and *K* is the index of the winning granule for each individual in generation *i*. At each table update, only N_G granules with highest L_k index are kept, and others are discarded. In [16], an example has been provided to illustrate the competitive granule pool update law.

While adding a new granule to the granule pool and assigning a life index to it is a simple way of controlling the size of the granule pool, since the granules with the lowest life index will be removed from the pool, it may happen that the new granule is removed, even though it was just inserted into the pool. In order to prevent this, the pool is split into two parts with sizes εN_G and $(1-\varepsilon)N_G$. The first part is a FIFO (First In, First Out) queue and new granules are added to this part. If it grows above εN_G, then the top of the queue is moved to the other part. Removal from the pool takes place only in the $(1-\varepsilon)N_G$ part. In this way, new granules have a good chance to survive a number of steps. In all of the simulations that are conducted here, ε is set at 0.1.

The distance measurement parameter is completely influenced by granule enlargement/shrinkage in widths of the produced MFs. As in [34], the combined effect of granule enlargement/shrinkage is in accordance with the granule fitness and it requires the fine-tuning of two parameters, namely β and γ. These parameters are problem dependent and it seems critical to set up a procedure in order to deal with this difficulty. The next section presents an auto-tuning strategy for determining the width of MFs which removes the need of exact parameter determination, without negative influence on the convergence speed.

D. How to Determine the Width of the Membership Functions?

It is crucial to have accurate estimation of the fitness function of the individuals in the finishing generations. In the proposed method, it can be accomplished by controlling the width of the produced MFs. At early steps of evolution, by choosing relatively large WMFs, the algorithm accepts individuals with less degree of similarity as similar individual. Therefore, the fitness should be computed by more often by estimation/association to the granules. As the individuals mature and reach higher fitness values, the width

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

decreases and the similarity between individuals should increase in order to be accepted as similar individuals. This prompts higher selectivity for granule associability and higher threshold for estimation. In short, in later generations, the degree of similarity between two individuals must be larger than that in the early generations, to be accepted as similar individuals. This procedure ensures a fast convergence rate due to rapid computation at the early phase and accurate fitness estimation at the later stage.

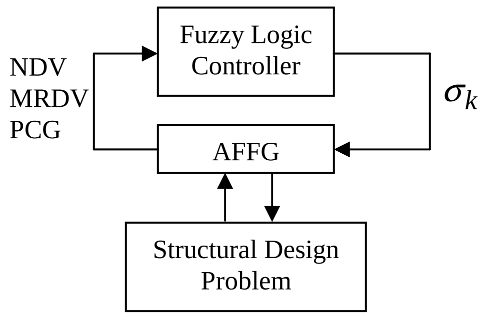


Figure. 2 Flow-diagram of Adaptive Fuzzy Controller

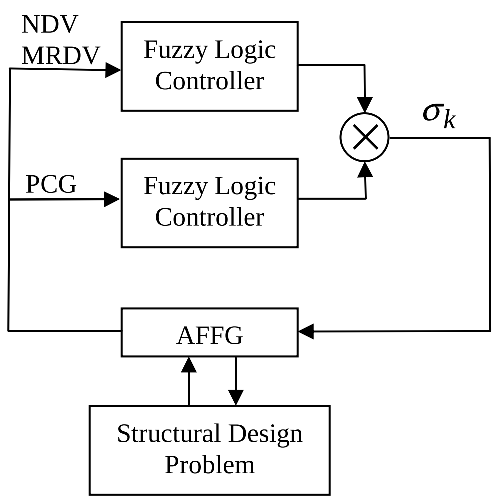


Figure. 3 Flow-diagram of Proposed Fuzzy Controller

To achieve these desiderata, a fuzzy supervisor with three inputs is employed. During the AFFG search, the fuzzy logic controller observes the Number of Design Variables (NDV), the Maximum Range of Design Variables (MRDV) and the percentage of completed trials, and specifies the WMFs. The first input is the NDV and the Range of the input variables (RIV) is the second one. Large values of the NDV and MRDV need big width in the MFs, *vice versa*. The Percent Completed Generations (PCG) is the third input, which takes a number in the range [0, 1], where ‘1’ signifies exhaustion of all allowed trials. This concerns the maturity level of search, given a fixed amount of resources. The combined effect of granule enlargement/shrinkage in accordance to PCG is to realize both rapid computation and accurate fitness estimation.

The architecture for adaptive fuzzy control of the WMFs is visualized in Figure 2. Gaussian MFs are used for specification of the knowledge base of the fuzzy logic controller. The knowledge base for controlling the WMFs based on the above architecture has a large number of rules and the extraction of these rules is very difficult. Consequently, a new architecture (as shown in Figure 3) is proposed, in which the controller is separated in two controllers to diminish the complexity of the system and to reduce the number 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, 0.3), the first number is the center and the second one is the spread), and the second controller has only one input. As shown in Figure 3, the spread of the granules is provided by the multiple output of the controllers. The knowledge base for the first controller 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 system (that employs singleton fuzzifier, products inference engine, and center average defuzzifier) adjusts σ_k after each generation.

3. BENCHMARK PROBLEMS AND NUMERICAL RESULTS

To illustrate the efficacy of the proposed granulation techniques, a set of 3 traditional optimization

TABLE II

Functions List of Test benchmark

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	$f(x)=10\cdot n+\sum_{i=1}^{n}(x_i^2-10\cdot\cos(2\cdot\pi\cdot x_i))$ $i=1:n;$ $-5.12\le x_i\le 5.12;$
Ackley	$f(x)=20+e-20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-e^{\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i}$ $i=1:n;$ $-32.768\le x_i\le 32.768;$

TABLE III

Parameters used for AFFG

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

benchmarks (shown in Table 2) are chosen namely: Griewangk, Rastrigin and Ackley. These benchmark functions are scalable and are commonly used to assess optimization algorithms. They have some intriguing features which most optimization algorithms find hard to deal with.

The Ackley function [31], [32] has an exponential term by which numerous local minima are produced. Analyzing a wider region helps to cross the valley along local optima, thereby achieving better solutions. The global optimum is always $f(x)=0$, which is obtained at $x_i=1$, $\forall i$.

The Rastrigin function [30] is created by adding a cosine modulation term to Sphere function. It consists of a large number of local minima whose values increase in receding from the global minimum. The global optimum is always $f(x)=0$ which occurs at $x_i=0$, $\forall i$.

The Griewangk function [33] is also highly multimodal. Unlike Ackley and Rastrigin functions, it has a product term that introduces interdependence among variables. It is hard to find the optimal solution without some information on the variables' dependences. Regardless of its dimensionality, the global optimum is $f(x)=0$ where $x_i=0$, $\forall i$.

The aim of the empirical study consists of investigating the search capability, as a function optimizer, of the proposed granulation technique (AFFG-FS), compared to the conventional GA, FES and AFFG techniques. The parameters are summarized in Table III.

The GA routine utilizes random initial populations, binary-coded chromosomes, single-point crossover, bit-wise mutation, fitness scaling, and an elitist stochastic universal sampling selection strategy. Moreover, crossover and mutation probabilities are $P_{XOVER} = 1$ and $P_{MUTATION} = 0.01$ respectively, the population size is 20, and the maximum number of generations is 100. Finally chromosome length varies depending on the number of variables in a given problem, but each variable's length is set to 8 bits. The total number of generations as well as the termination criterion is determined during several trial runs to ensure the convergence of the algorithm on the three benchmark problems.

AFFG and AFFG-FS uses all of the above evolutionary parameters as in a GA to establish analysis only from the perspective of granulation and in order to keep track of the best solution found. Ten independent runs of each experiment were executed.

As to FES, a fitness and an associated reliability values are assigned to each new individual. The fitness is actually evaluated if the reliability value is below a certain threshold. The reliability value varies between 0 and 1 and depends on two factors: the first one is the reliability of parents, and the second one is the closeness of parents and children in the solution space. Three different levels for T, i.e., 0.5, 0.7 and 0.9, have been used here which equal to ones proposed in [24].

In this experiment, four sets of dimensions are considered for each test function; namely n = 5, 10, 20 and 30. As for AFFG and AFFG-FS, the number of individuals in the granule pool is varied between 20, 20, 40 and 80 respectively. The reported results were obtained by achieving the same level of fitness evaluations for both the proposed method (AFFG-FS) and the comparative references (GA, FES and AFFG), namely 500 for 5-D (dimension), 1000 for 10-D, 2000 for 20-D and 3000 for 3-D.

The average convergence trends of the standard GA, FES, AFFG and AFFG-FG are summarized in Figures 4-15. All the results presented were averaged over 10 runs. In figure 4-18, the y-axis denotes the (average) fitness value in common logarithmic scale, and the x-axis denotes the number of exact function evaluation.

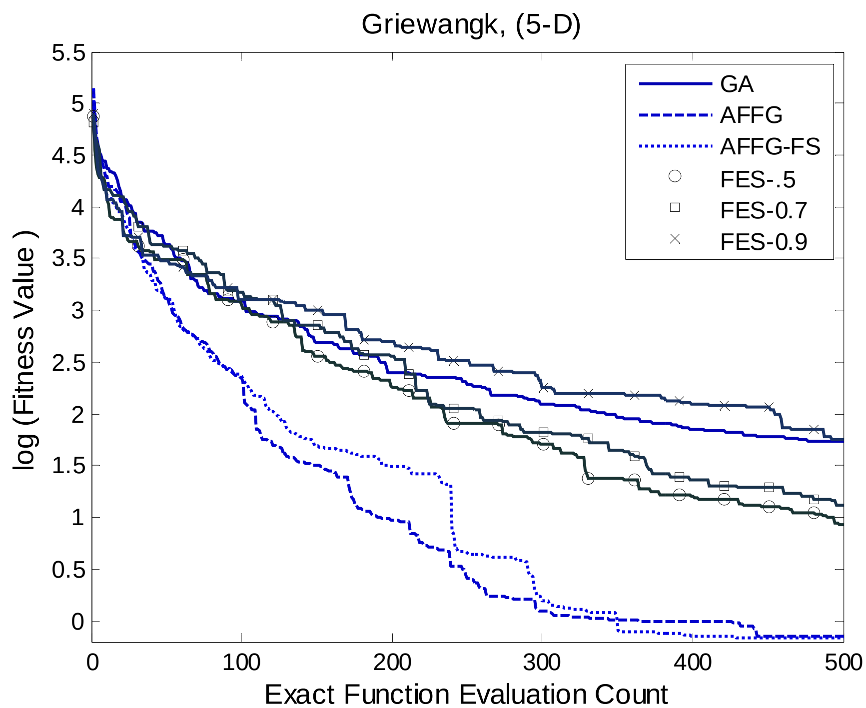


Figure 4: Convergence Trend of GA, FES, AFFG and AFFG-FS on 5-D Griewangk function.

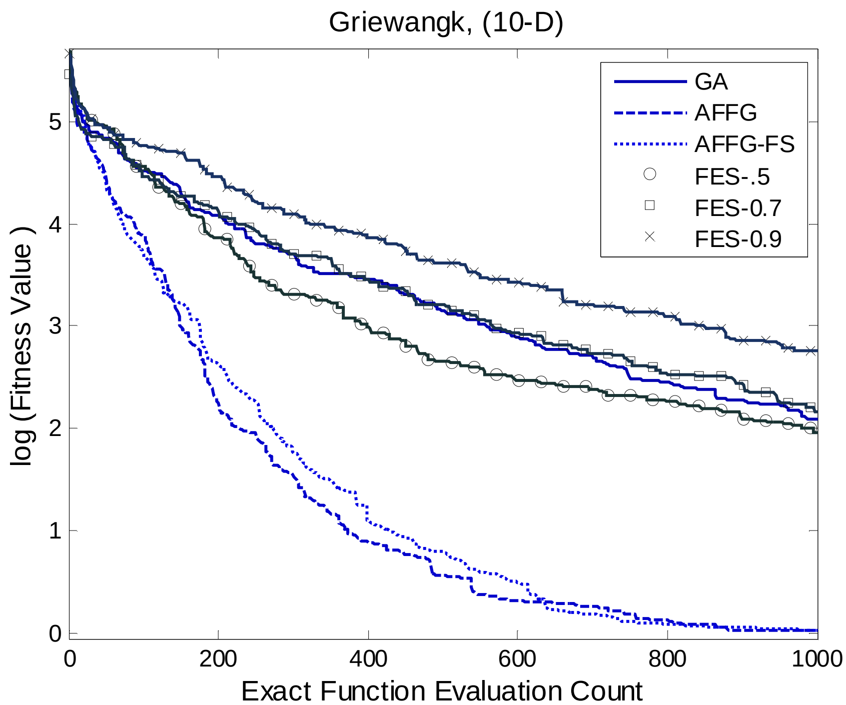


Figure 5: Convergence Trend of GA, FES, AFFG and AFFG-FS on 10-D Griewangk function.

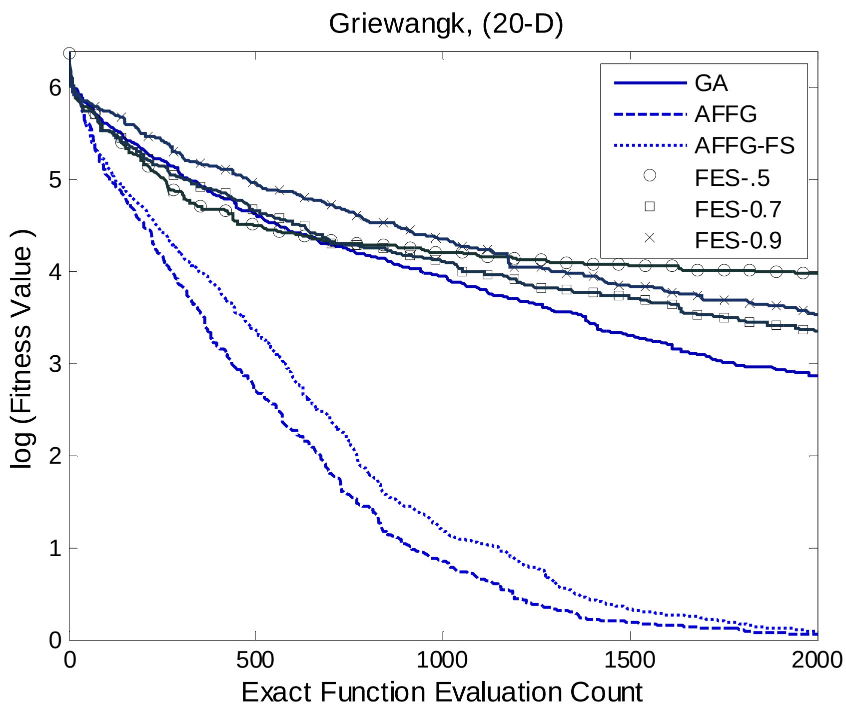


Figure 6: Convergence Trend of GA, FES, AFFG and AFFG-FS on 20-D Griewangk function.

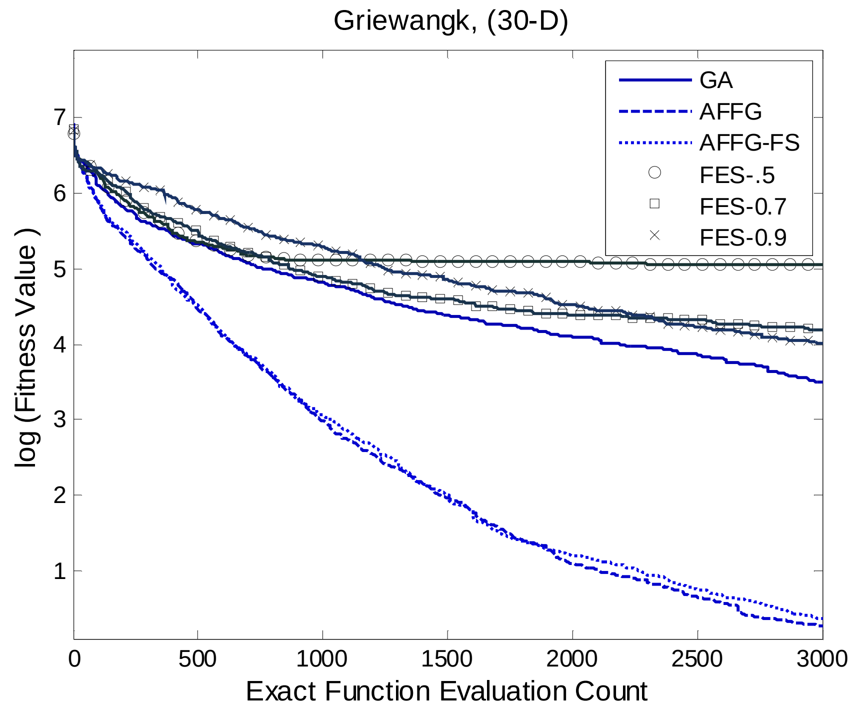


Figure 7: Convergence Trend of GA, FES, AFFG and AFFG-FS on 30-D Griewangk function.

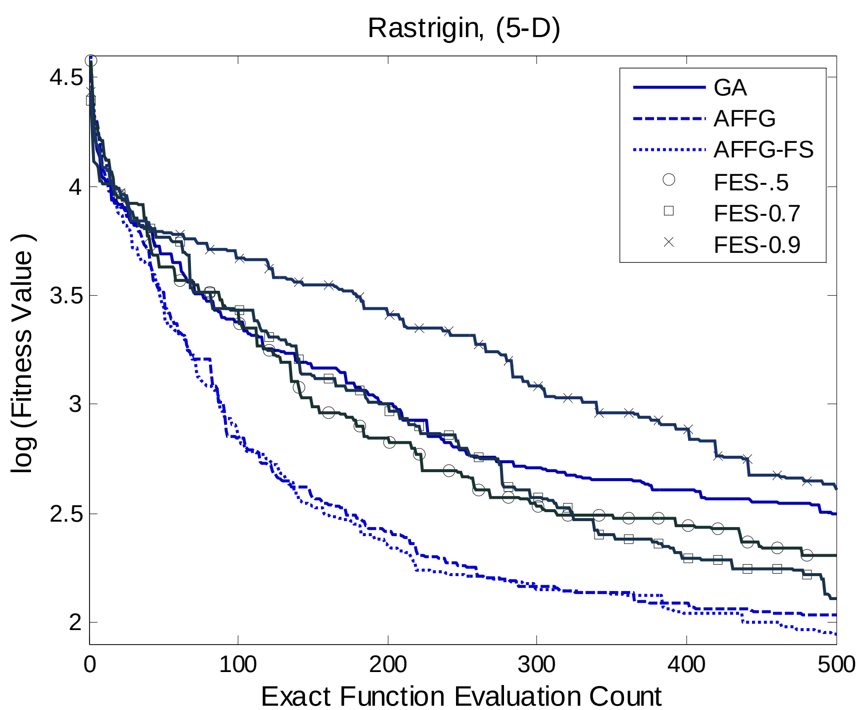


Figure 8: Convergence Trend of GA, FES, AFFG and AFFG-FS as applied to 5-D Rastrigin function.

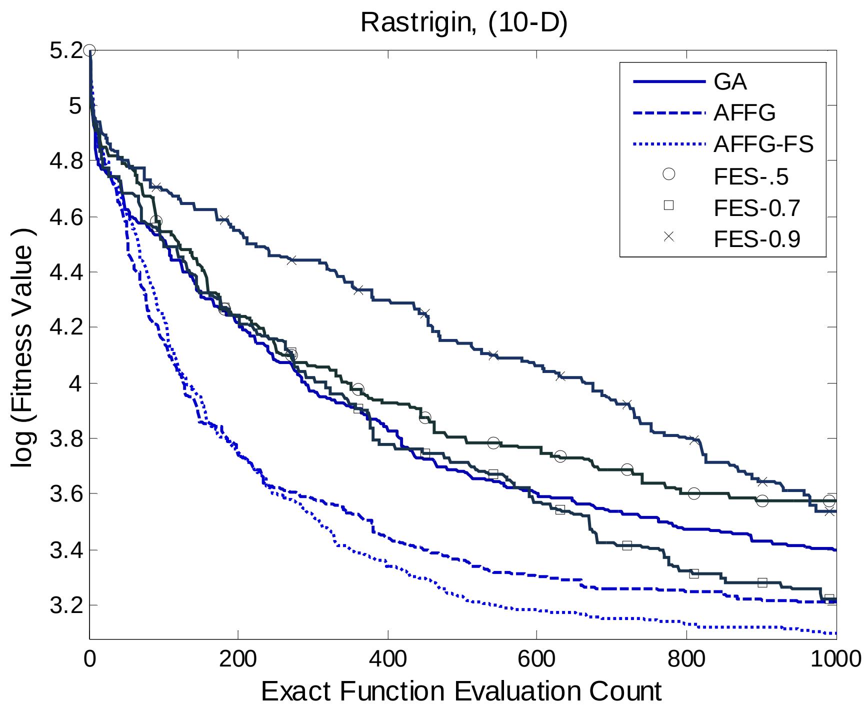


Figure 9: Convergence Trend of GA, FES, AFFG and AFFG-FS as applied 10-D Rastrigin function.

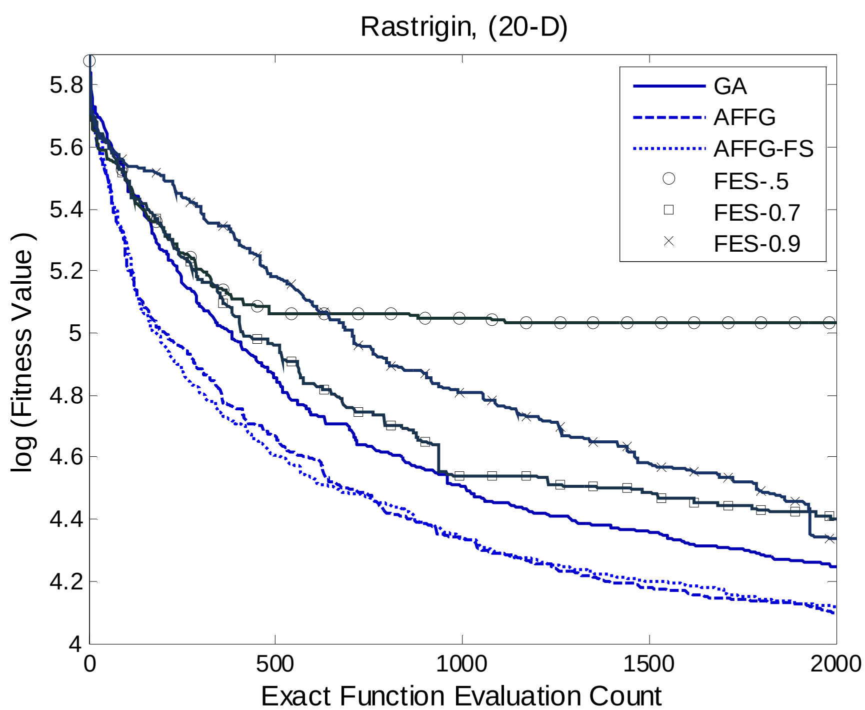


Figure 10: Convergence Trend of GA, FES, AFFG and AFFG-FS as applied to 20-D Rastrigin function.

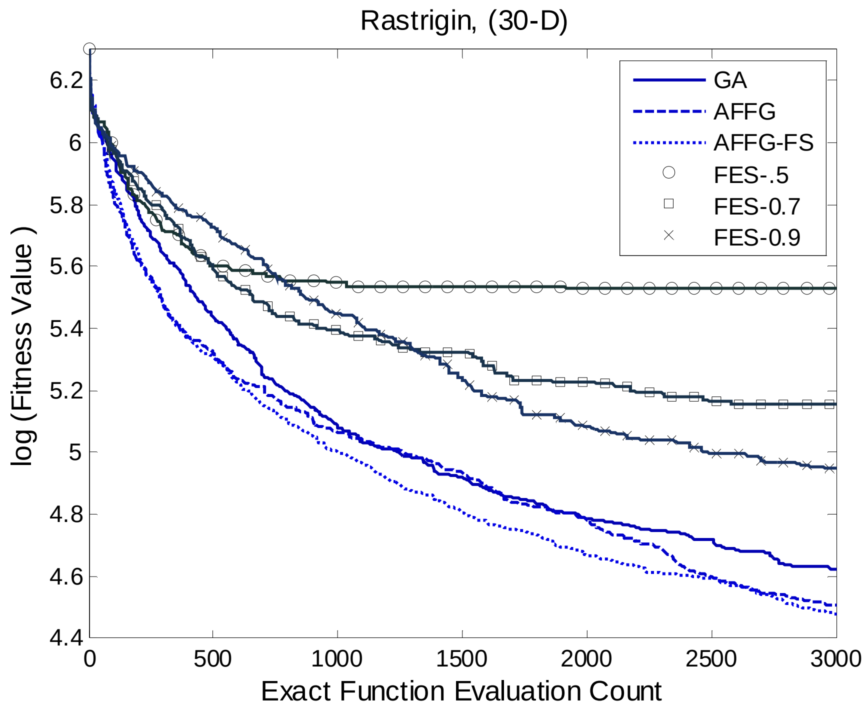


Figure 11: Convergence Trend of GA, FES, AFFG and AFFG-FS as applied to 30-D Rastrigin function.

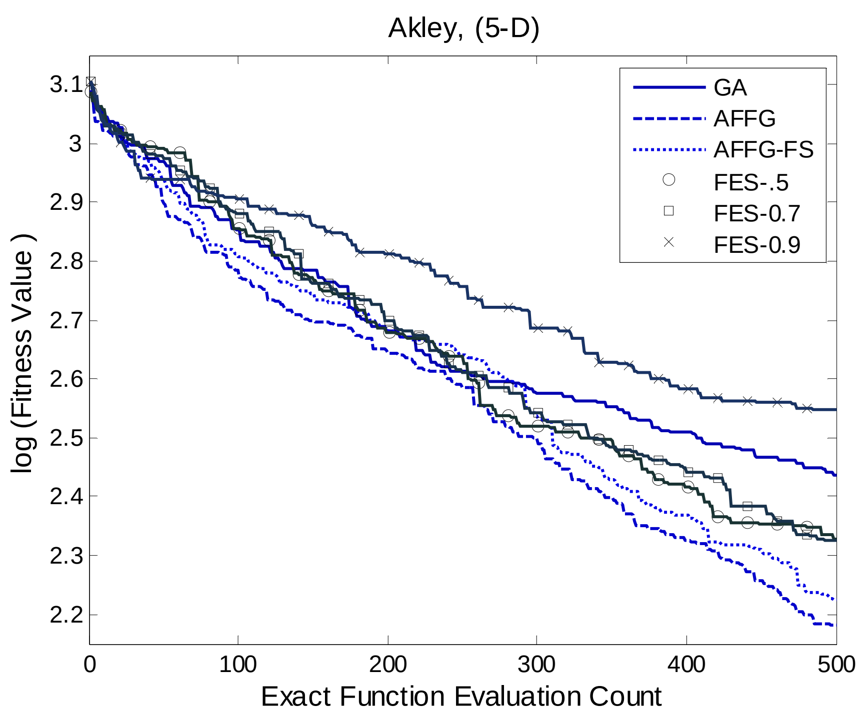


Figure 12: Convergence Trend of GA, FES, AFFG and AFFG-FS with regard to 5-D Ackley function.

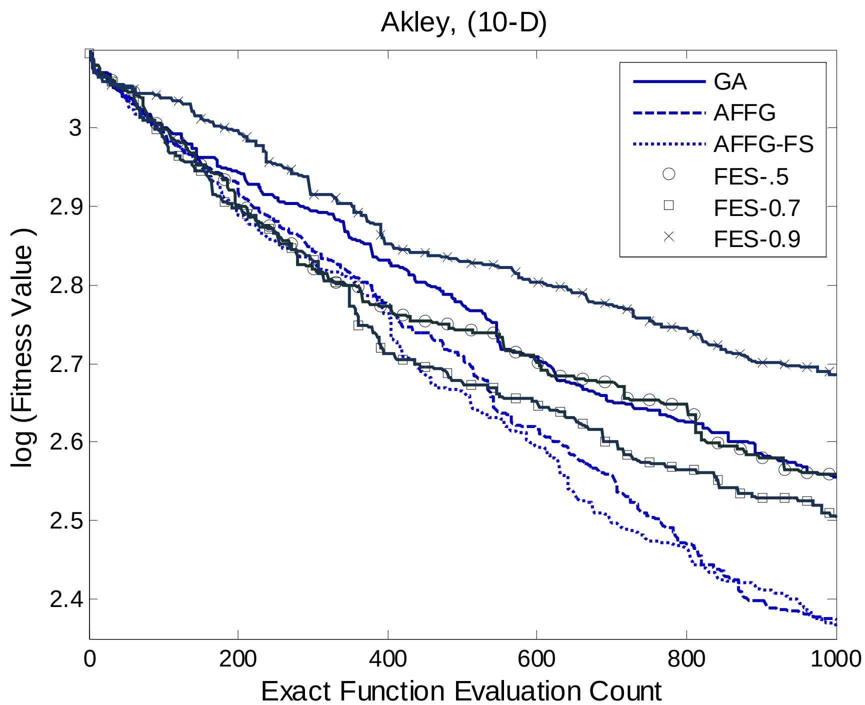


Figure 13: Convergence Trend of GA, FES, AFFG and AFFG-FS with regard to 10-D Ackley function.

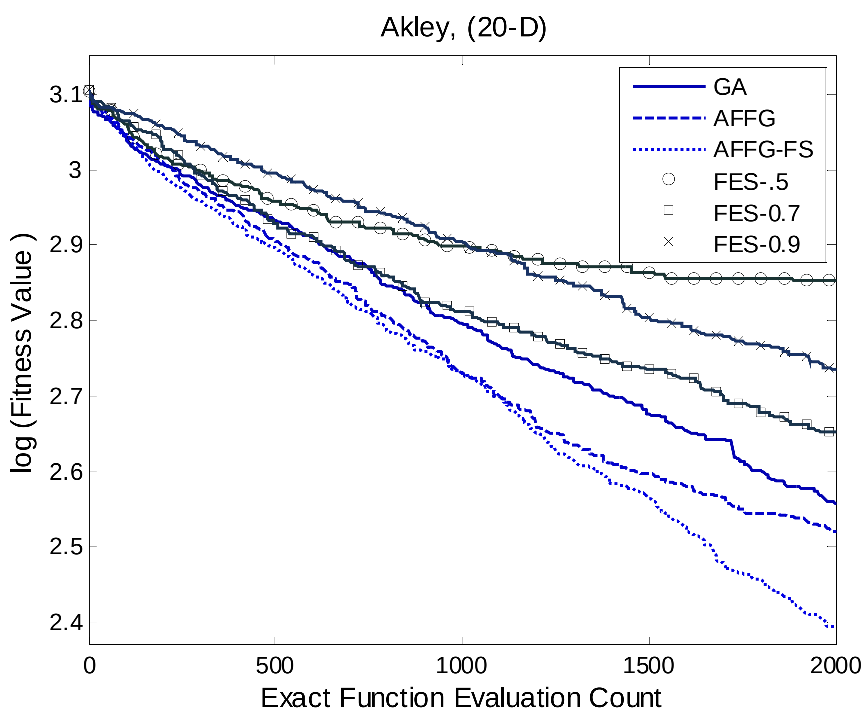


Figure 14: Convergence Trend of GA, FES, AFFG and AFFG-FS with regard to 20-D Ackley function.

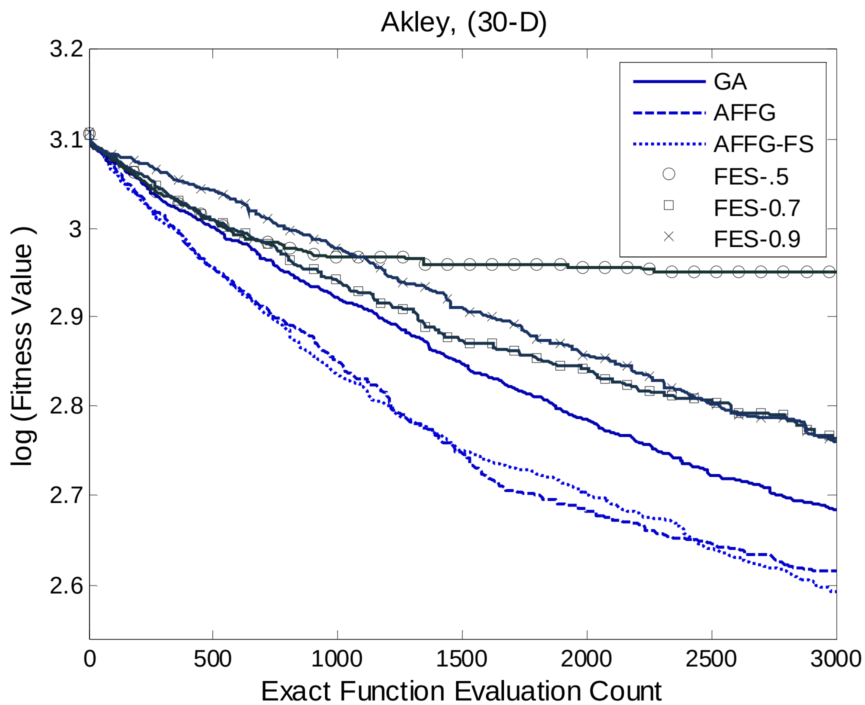


Figure 15: Convergence Trend of GA, FES, AFFG and AFFG-FS with regard to 30-D Ackley function.

As shown in Figures 4-15, the search performance of AFFG and AFFG-FS are superior to GA and FES, even with a small number of individuals in the granule pool. The results also illustrate that fitness inheritance method (i.e., FES), albeit being comparable in smaller dimensions, deteriorates as the problem size increases.

We also studied the effect of varying the number of granules N_G on the convergence behavior of AFFG and AFFG-FS. The comparison can be made by the results obtained in Figures 16-18. The good news from the results is that AFFG and AFFG-FS are not so sensitive to N_G. However, further increase of N_G slows down the rate of convergence due to the imposed computational complexity.

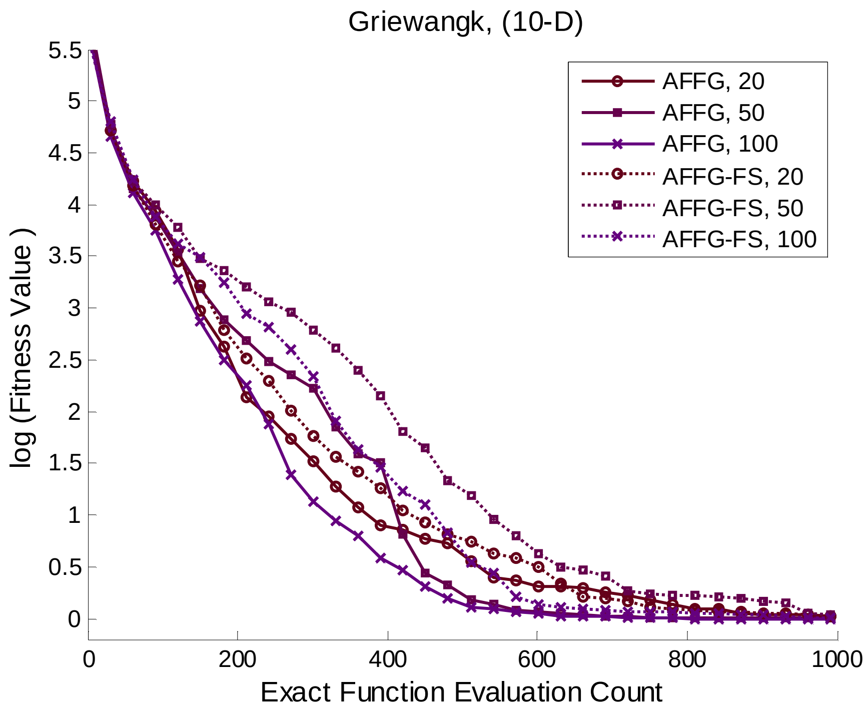


Figure 16: Effect of Varying $N_G \in \{20, 50, 100\}$ on Convergence Trend for Griewangk function.

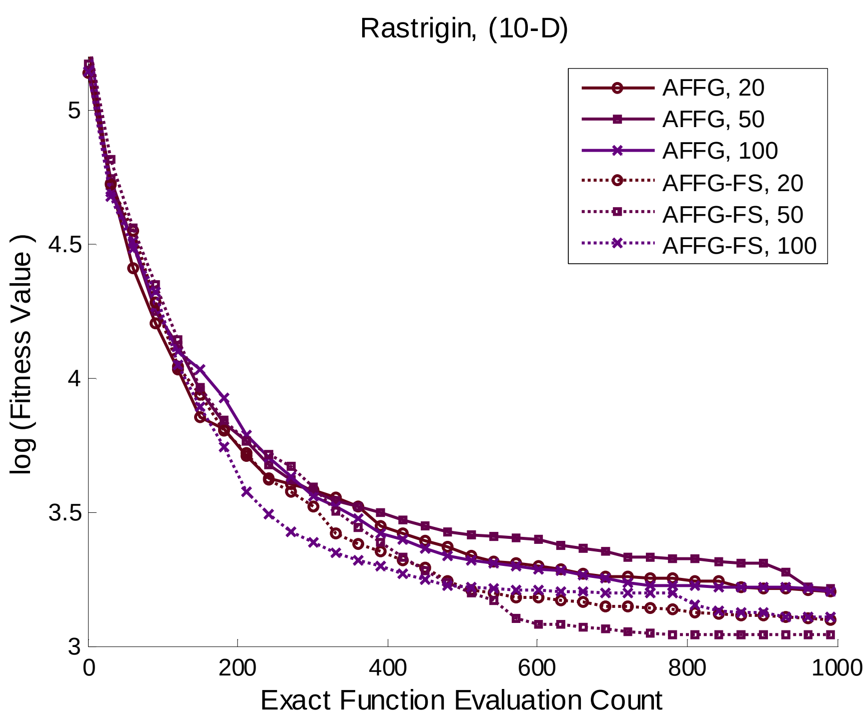


Figure 17: Effect of Varying $N_G \in \{20, 50, 100\}$ on Convergence Trend for Rastrigin function.

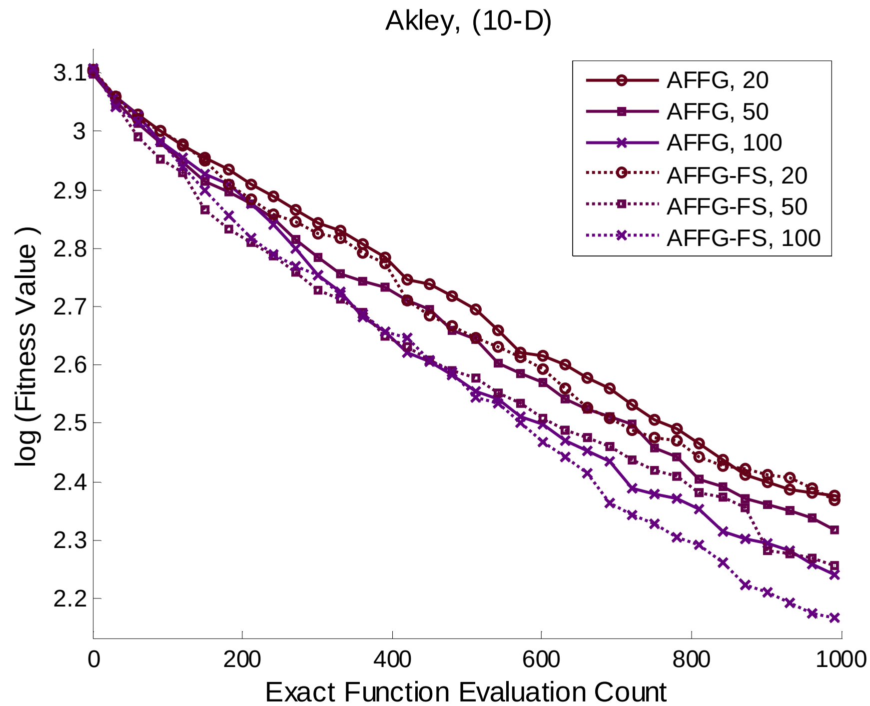


Figure 18: Effect of Varying of $N_G \in \{20, 50, 100\}$ on Convergence Trend for Ackley function.

4. SPREAD SPECTRUM WATERMARKING

This section bears out the effectiveness of the proposed granulation technique in real world applications. To this end, we consider a hidden information detection problem such that the correct PN sequence must be recovered from a spread spectrum watermarked signal. Spread spectrum watermarking (SSW) has been perceived to be a powerful watermarking scheme that offers high robustness (surviving hidden information after noise addition), high transparency (high quality of watermarked signal after addition of hidden information) and high security (against unauthorized users) to hide the bits of information. SSW uses the idea of spread spectrum communication to embed bits of information into a host signal. Spreading the spectrum of the hidden information is carried out by a pseudorandom noise (PN) sequence. A PN sequence is a zero mean, periodic binary sequence with a noise-like waveform whose bits are equal to +1 or –1 [25]. To embed each bit of hidden information $m(i)$, i = 1, 2…, into a host signal, the embedder conducts the following steps.

Step. 1: Generates one period of the PN sequence by a PN sequence generator.

Step. 2: Multiplies $m(i)$ by all the bits of the generated PN sequence to generate a watermark signal as follows:

$$w(i) = p(n)\, m(i), \quad n = 1,\ldots, N \tag{5}$$

where $p(n)$ is the *n-th* bit of the PN sequence and $w(i)$ is the *i-th* block of the watermark signal.

Step. 3: Produces a watermarked signal $s(w,x)$ as follows:

$$S(w, x) = \lambda\, w(n) + x(n) \tag{6}$$

Then the watermarked signal $S(w,x)$ is sent to the receiver.

Extraction of hidden information from a received watermarked signal at the detector can be done using the correlation property of the PN sequence. Cross correlation $C(.,.)$ between two PN sequences p_a and p_b is given as (7) [26]:

$$C(P_a, P_b) = \frac{1}{N}\sum_{i=0}^{N-1}(P_a(i)\; P_b(i)) = \begin{cases} i, & if\;\; a = b \\ -1/N, & otherwise \end{cases} \tag{7}$$

Hence, cross correlation between a watermarked signal and a PN sequence can be written as:

$$C(S, p') = C(w, p') + C(m.p, p') = \begin{cases} C(w, p') + m & if \quad p = p' \\ C(w, p') - \dfrac{m}{N} & Otherwise \end{cases} \tag{8}$$

Equation (8) expresses that the bit of hidden information can be determined by calculating the correlations between the received watermarked signal and the PN sequence employed at the transmitter, and comparing the result with a threshold.

A. RECOVERING THE PN SEQUENCE

In general, it is very hard to recover the PN sequence from a spread spectrum watermarked signal where no information about the PN sequence or its location is known. The reason is that there are vast regions for the solution sets of possible PN sequences. For instance, to recover a PN sequence with a period equal to 63 bits, 2^{63} PN sequences must be generated.

To make the problem of recovering the PN sequence more tractable, we assume that the exact location of the watermark in the watermarked signal is known. In this section, a novel algorithm for detecting the location of the watermarked signal will be explained. In [20], an approach for detecting hidden information from an image spread spectrum signal has been proposed. This approach detects abrupt jumps in the statistics of the watermarked signal to recover the PN sequence. However, the algorithm which is based on hypothesis tests for detection of abrupt jump in the statistics is very complicated and its performance suffers from low frequency embedding.

Our approach to recover the PN sequence is based on unconstrained optimization. We have a set of feasible solutions available in order to find the global minimum of a cost function. The feasible solutions are sequences with the period length of the PN sequence and elements of +1 and –1. A cost function for this problem can be defined by a exploring a very useful property of SSW (in detection), namely the correlation property of the PN sequence. Thus, the proper cost (fitness) function is the cross correlation between the generated sequence and the watermarked signal as is defined in Equation (8).

In [21], an interesting method for recovering the PN sequence of the spread spectrum signal with a predefined SNR has been proposed. The approach uses a GA approach with a fitness function based on the cross correlation between the estimated PN sequence and the spread spectrum. However, spread spectrum watermarking is more complicated than a single spread spectrum signal since, in SSW, the spread spectrum hidden information is like a white Gaussian noise for the host signal.

We observe here that the computation of the cross correlation between the sequences of possible solutions' set and the watermarked signal for only one block of the SSW signal would not converge to the PN sequence used at the transmitter. This is because the energy of host signal is at least 12 dB more than the energy of the watermark, and that has a strong effect on maximizing the cross correlation (i.e., the optimization algorithm converges to a sequence that maximizes the correlation with the host). As a solution to this problem, several consequence blocks of the watermark (i.e. several bits of hidden information) should be considered in the computation of the cross correlation. In this case, the watermark signal has a stronger effect than the host signal on maximizing the cross correlation function.

Carrying out the global optimization by searching over the entire solution set, as mentioned above, is the subject of deterministic methods such as covering methods, tunneling methods, zooming methods, etc. Such methods discover the global minimum by an exhaustive search over the entire solution set. For instance, the basic idea is to cover all the feasible solutions by evaluating the objective function at all points [22]. Although these schemes have high reliability and accuracy is always guaranteed, they are not practical due to their poor convergence [23].

Since the solution set is vast, we need an efficient optimization algorithm with high reliability and fast converging rate. Many stochastic optimization algorithms have been proposed such as GA, simulated annealing, ant colony, etc. However, the GA approach has been perceived to be promising in a wide range of applications. Moreover, it is apt to strike an attractive balance between reliability and converging rate. In this regard, we have chosen the GA framework for the global optimization task. In order to further enhance the search capability, we employ the proposed AFFG-FS (of Section II) with a view to reduce the number of expensive fitness evaluations by incorporating an approximate model.

B. Empirical Results for Recovering PN Sequence

This empirical study focuses on performance improvement of the proposed granulation technique (AFFG-FS) in comparison with conventional GA approaches [35]. In Section 3, it has been exhibited that the fuzzy supervisory part of AFFG-FS gets rid of the need of exact parameter determination of AFFG, and their performances are comparable to each other. Moreover, it has also been shown that FES is much worse than the granulation techniques. As such, we did not take into account the original AFFG and FES as comparative references in this experiment.

In order to reasonably keep track of the best solution found, the GA uses roulette-wheel selection with elitism. Moreover, one-point crossover and bit-wise mutation are implemented. Crossover and mutation probabilities used are 1.0 and 0.01, respectively. The population size is set to 20 with the elite size of 2.

For AFFG-FS, the number of individuals in the granule pool varies between 10, 20 and 50. The reported results were obtained by achieving the same level of fitness evaluations for both a canonical GA and the proposed AFFG-FS. In this experiment, all results were averaged over 10 runs.

Average convergence performance of GA and AFFG-FG is depicted in Figures 19-21 and is summarized in Table IV. It is seen that cross correlation values returned by AFFG with N_G ={10,20,50} are much better than that of GA. It is also observed that the cross correlation increases, albeit insensitive, with the number of granules. However, the increase of N_G slows down the rate of convergence due to its imposed computational complexity. Moreover, Table IV exhibits that the rate of convergence of AFFG-FS is, on average, 3.5 times faster than that of GA. It is noted that the performance gain is not so dependent on the chip length of the PN sequence (i.e., problem size). From the results, it can be concluded that the search performance of AFFG-FS is superior to that of the GA, even with the small number of individuals in the granule pool.

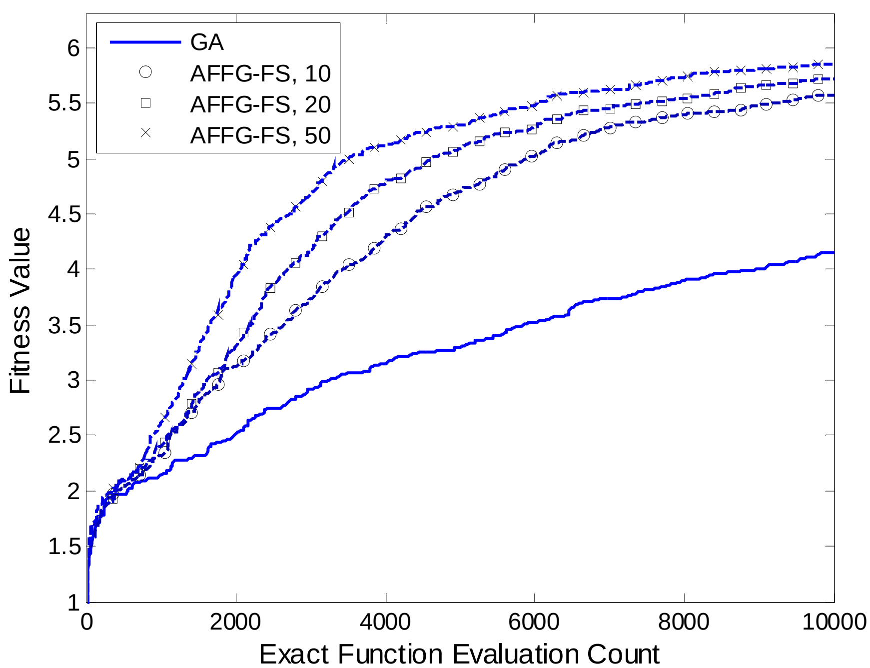


Figure 19: Cross correlation between the estimated PN sequence with the period of 255

chips and the watermarked signal

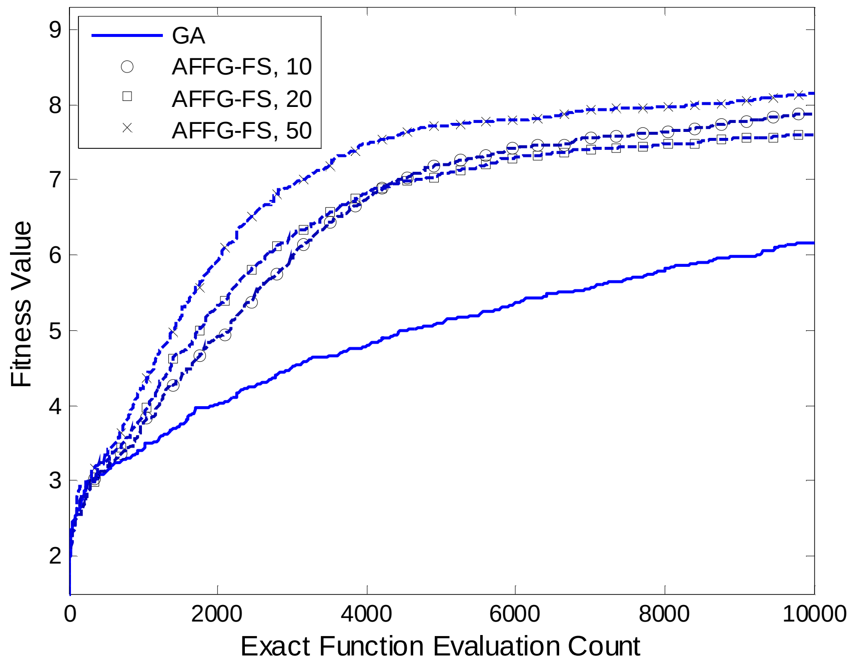


Figure 20: Cross correlation between the estimated PN sequence with the period of 127 chips and the watermarked signal

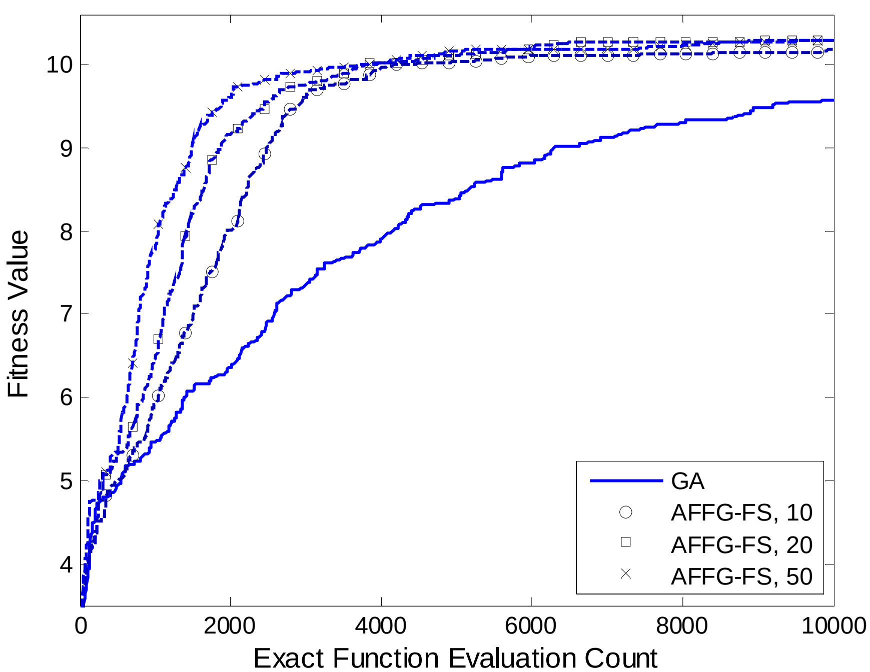


Figure 21: Cross correlation between the estimated PN sequence with the period of 63 chips and the watermarked signal

Table IV: Performance comparison of GA and AFFG-FS with N_G={10, 20, 50}

Chip Length		Criteria-I	Criteria-II	Criteria-III
255	GA	4.51	4.16	9934
	AFFG-FS, 10	5.90	5.58	3817
	AFFG-FS, 20	6.10	5.72	2969
	AFFG-FS, 50	6.19	5.86	2156
127	GA	6.54	6.16	9994
	AFFG-FS, 10	8.39	7.88	3211
	AFFG-FS, 20	8.00	7.60	2952
	AFFG-FS, 50	8.45	8.14	2194
63	GA	10.17	9.57	9965
	AFFG-FS, 10	10.29	10.18	2978
	AFFG-FS, 20	10.36	10.29	2547
	AFFG-FS, 50	10.39	10.28	1904

Criteria-I: The best cross correlation of population at the last generation.

Criteria-II: The average cross correlation of population at the last generation.

Criteria-III: The average number of fitness evaluations until the same cross correlation value is reached (the values are equal to the average cross correlation of population achieved by GA at the last generation); 4.16 for 255 chips, 6.16 for 127 chips, 9.57 for 63 chips.

5. Concluding Remarks

An intelligent guided technique via an adaptive fuzzy similarity analysis for fitness granulation, called *adaptive fuzzy fitness granulation with fuzzy supervisory* (AFFG-FS), has been presented. The aim was to decide on use of expensive function evaluation and adapt the predicted model in a dynamic manner. A fuzzy supervisor as an auto-tuning strategy has also been proposed in order to avoid the tuning of parameters. Empirical evidence on its effectiveness over existing approaches (i.e., GA and FES) was adduced with widely-known benchmark functions. In detail, numerical results showed that the proposed technique is capable of optimizing functions of varied complexity efficiently. It was seen that AFFG and AFFG-FS are not much sensitive to the number of granules (N_G), and smaller values of N_G still lead to good results. Moreover, the auto-tuning of fuzzy supervisor eliminated the need for exact parameter determination without compromising convergence performance.

The proposed AFFG-FS has been further extended into detecting hidden information from a spread spectrum watermarked signal. Under the assumption of knowing the location of hidden information, the knowledge necessary for detecting hidden information at the receiver (that is the PN sequence used at the transmitter) could be detected. Experimental studies demonstrated that AFFG-FS is capable of rapidly detecting hidden information.

Acknowledgment

This research received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. INFSOICT-223844, the Next Generation Infrastructures Research Program of Delft University of Technology and the Mexican CONACyT project No. 45683-Y.

References

[1] N. Cvejic, “Algorithms for Audio Watermarking and Steganography”, PhD thesis, Oulu University of Technology, June 2004.

[2] K. Gopulan “Audio steganography using bit modification”, Proceedings of the 2003 International Conference on Acoustic Speech and Signal Processing, 2003.

[3] R. Ansari, H. Malik, A. Khikhar “Data-hiding in audio using frequency-selective phase alteration”, Proceedings of the IEEE International Conference on Acoustic Speech and Signal Processing, 2004.

[4] H. Joong, Y. H. Choi, “a novel echo-hiding scheme with forward backward kernels” IEEE Transactions on Circuits and Systems for Video Technology, Volume 13, No 8, Aug 2003.

[5] Z. Liu, A. Inue, “Spread spectrum watermarking of audio signals”, IEEE Transactions on Circuits and System for Video Technology, Volume 13, No. 8, Aug 2003.

[6] Saeed Sedghi, Habib Rajabi Mashhadi, Morteza Khademi. “Detecting Hidden Information from a Spread Spectrum Watermarked Signal by Genetic Algorithm”, IEEE Congress on Evolutionary Computation, pp. 480--485, July 16-21, 2006

[7] J.-H. Chen, D. Goldberg, S.-Y. Ho, and K. Sastry, “Fitness inheritance in multiobjective optimization”, Proceedings of the 2002 International conference on Genetic and Evolutionary Computation Conference, pp. 319--326, 2002.

[8] Margarita Reyes-Sierra , Carlos A. Coello Coello, “Dynamic fitness inheritance proportion for multi-objective particle swarm optimization”, Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, July 2006.

[9] Mehrdad Salami , Tim Hendtlass, “The Fast Evaluation Strategy for Evolvable Hardware”, Genetic Programming and Evolvable Machines, Volume 6, No. 2, p.139-162, June 2005.

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

[11] Y.-S. Hong, H.Lee, and M.-J. Tahk, “Acceleration of the convergence speed of evolutionary algorithms using multi-layer neural networks”, Journal of Engineering Optimization, Volume 35, No. 1, pp. 91--102, 2003.

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

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

[14] 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.

[15] M. Davarynejad, “Fuzzy Fitness Granulation in Evolutionary Algorithms for Complex Optimization”, M.Sc. Thesis. Ferdowsi University of Mashhad, Department of Electrical Engineering, 2007.

[16] M. Davarynejad, M.-R. Akbarzadeh-T, N. Pariz, “A novel general framework for evolutionary optimization: adaptive fuzzy fitness granulation”, in: Proceedings of the 2007 IEEE International Conference on Evolutionary Computation (CEC'2007), pp. 951--956, IEEE Service Center, Singapore, September 2007.

[17] Margarita Reyes Sierra, Carlos A. Coello Coello “Dynamic fitness inheritance proportion for multi-objective particle swarm optimization”, GECCO, 2005.

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

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

[20] S. Trivedi and R. Chandramouli, ”Secret Key Estimation in Sequential Steganography”, IEEE Transaction on signal processing, Volume 53, No. 2, Feb 2005.

[21] V. R. Asghari and M. Ardebilipour, “Spread Spectrum Code Estimation by Genetic Algorithm”, International Journal of Signal Processing, Vol. 1,Number 4, 2004.

[22] J. S. Arora, O. A. Elwakeil and A. Chahande, “Global optimization methods for engeering applications: a review”, Optimal Design Laboratory, 1995.

[23] K. Yen and L. Hanzo, ”Genetic Algorithm Assisted Joint Multiuser Symbol Detection and Fading Channel Estimation for Synchronous CDMA Systems”, IEEE Transaction on Selected Areas in Communication, Vol. 19, No. 6, June 2001.

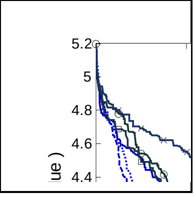
[24] M. Salami and T. Hendtlass, “A fast evaluation strategy for evolutionary algorithms”, Applied Soft Computing, Vol. 2, pp. 156--173, 2003.

[25] S. Haykin, “Communication Systems,” 4th edition, John Wiley & Sons, Inc, 2001.

[26] Z. Liu, Y. Kobayashi, S. Sawato, and A. Inoue, “A robust audio watermarking method using sine function patterns based on pseudorandom sequences”, in Proc. Pacific Rim Workshop on Digital Steganography, 2002.

[27] M. Davarynejad, M.-R. Akbarzadeh-T and Carlos A. Coello Coello, “Auto-Tuning Fuzzy Granulation for Evolutionary Optimization”, in 2008 Congress on Evolutionary Computation (CEC'2008), pp. 3573--3580, IEEE Service Center, Hong Kong, June 2008.

[28] L.A. Zadeh, “Fuzzy sets and information granularity”, in: M. Gupta, R. Ragade, R. Yager (Eds.), Advances in Fuzzy Set Theory and Applications, North-Holland Publishing Co., pp. 3--18, Amsterdam, 1979,.



[29] M. Husken, Y. Jin, B. Sendhoff, “Structure optimization of neural networks for evolutionary design optimization”, Soft Computing 9 (1), pp. 21--28, 2005.

[30] L. A. Rastrigin, “Extremal control systems”, In Theoretical Foundations of Engineering Cybernetics Series. Moscow: Nauka, Russian, 1974.

[31] D. Ackley, “An empirical study of bit vector function optimizacion”, Genetic Algorithms and Simulated Annealing, pp. 170--215, 1987.

[32] T. Bäck and H. P. Schwefel, “An overview of evolutionary algorithms for parameter optimization”, Evolutionary Computation, 1(1), pp. 1--23, 1993.

[33] T. Bäck, D. Fogel, and Z. Michalewicz, “Handbook of Evolutionary Computation”, Institute of Physics Publishing Ltd, Bristol and Oxford University Press, New York, 1997.

[34] M.-R. Akbarzadeh-T, M. Davarynejad, N. Pariz, “Adaptive Fuzzy Fitness Granulation for Evolutionary Optimization”, International Journal of Approximate Reasoning, Vol. 49(3), pp. 523--538, 2008.

[35] M. Davarynejad, S. Sedghi, M. Bahrepour, C. W. Ahn, M. Akbarzadeh, C. A. Coello Coello, “Detecting Hidden Information from Watermarked Signal using Granulation Based Fitness Approximation”, World Conference on Soft Computing in Industrial Applications, 2008.