

Research Article

Automatic Curve Fitting Based on Radial Basis Functions and a Hierarchical Genetic Algorithm

G. Trejo-Caballero,^{1,2} H. Rostro-Gonzalez,¹ C. H. Garcia-Capulin,² O. G. Ibarra-Manzano,¹ J. G. Avina-Cervantes,¹ and C. Torres-Huitzil³

¹Department of Electronics, DICIS-University of Guanajuato, Comunidad de Palo Blanco s/n, 36885 Salamanca, GTO, Mexico

²Department of Mechatronics, ITESI, Carretera Irapuato-Silao Km. 12.5, 36821 Irapuato, GTO, Mexico

³Information Technology Laboratory, CINVESTAV, Parque Científico y Tecnológico, 87130 Ciudad Victoria, TAMP, Mexico

Correspondence should be addressed to G. Trejo-Caballero; getrejo@itesi.edu.mx

Received 30 September 2015; Revised 21 November 2015; Accepted 7 December 2015

Academic Editor: Akemi Gálvez

Copyright © 2015 G. Trejo-Caballero et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Curve fitting is a very challenging problem that arises in a wide variety of scientific and engineering applications. Given a set of data points, possibly noisy, the goal is to build a compact representation of the curve that corresponds to the best estimate of the unknown underlying relationship between two variables. Despite the large number of methods available to tackle this problem, it remains challenging and elusive. In this paper, a new method to tackle such problem using strictly a linear combination of radial basis functions (RBFs) is proposed. To be more specific, we divide the parameter search space into linear and nonlinear parameter subspaces. We use a hierarchical genetic algorithm (HGA) to minimize a model selection criterion, which allows us to automatically and simultaneously determine the nonlinear parameters and then, by the least-squares method through Singular Value Decomposition method, to compute the linear parameters. The method is fully automatic and does not require subjective parameters, for example, smooth factor or centre locations, to perform the solution. In order to validate the efficacy of our approach, we perform an experimental study with several tests on benchmarks smooth functions. A comparative analysis with two successful methods based on RBF networks has been included.

1. Introduction

In the literature, there are many methods to tackle the curve fitting problem, which remains challenging and elusive. In this study, the goal is to build a compact representation of the curve that corresponds to the best estimate of the unknown relationship between two variables from a set of data points. Curve fitting is a fundamental tool in scientific and engineering applications such as system identification, data analysis and visualization, geometric modeling, CAD/CAM systems, medical imaging, and reverse engineering [1–6].

The curve fitting problem has been mainly addressed by using typical methods based on linear models [7–10]. These methods consider that, given a set of data points, any function can be properly approximated on a specific interval using a linear combination of a set of m fixed functions often called basis functions. The main basis functions used to address

the curve fitting problem are polynomials, piecewise polynomials (splines), and radial basis functions (RBFs). RBFs have typically shown a successful performance in methods based on interpolation, such as in [11, 12].

In [13], the authors present a method for spike classification enhancement based on the 3-Gaussian model fitting. In [14], the peak wavelength detection accuracy in fiber Bragg grating sensors is improved by using a wavelet filter and curve fitting based on RBF. In [15], it is demonstrated that a Gaussian curve fitting substantially reduces the measurement errors of spectrally distorted FBG sensors by employing a binary search algorithm that calculates and compares MSE values at only logically selected positions. In [12] the authors propose a method based on RBF interpolation to subpixel mapping of remote sensing images by fully exploiting the spatial information in the input images. The method uses RBF interpolation to predict the soft values at each subpixel.

However, the interpolation methods present some limitations, especially when the number of data samples increases or the set of data samples is disturbed by noise. Thus, in order to solve the limitations of the interpolation methods, other methods have been proposed based on regression techniques. Nevertheless, these methods require the specification of global parameters such as the number of RBFs or centre locations to perform their solution. If this is not the case, if the basis functions and any parameters which they might contain change, then the model is nonlinear and the adequate choice of parameters becomes a continuous multimodal and multivariate nonlinear optimization problem, which must be addressed using modern techniques of Computational Intelligence.

In [16] the authors propose a method based on curve fitting with Gaussian functions for modeling carotid and radial artery pulse pressure waveforms. The method uses a fixed number of Gaussian functions, and their parameters are determined by using an algorithm based on particle swarm. Neural networks based on RBF have been applied to curve fitting in [17]. This paper considers fitting noisy curves using two neural networks: the multilayer feed forward network and the radial basis function network. A comparative analysis shows that when the noise level increases, the RBF networks are best suited for the reconstruction of noisy curves. However, most of these approaches use parameter values predefined without any justification, that is, in an empirical way. In [18], the spread parameter value of a neural network is selected by trial and error from a reasonably small interval previously determined.

It is found that RBFs have rarely been applied to curve fitting by using a linear combination to change the number of radial basis functions and simultaneously to optimize their parameters. This paper elucidates the feasibility of using RBFs in a linear combination to fit a set of data points disturbed by noise; that is, we propose a new method to tackle the curve fitting problem using a strictly linear combination of RBFs. The proposed approach divides the parameter search space into linear and nonlinear parameter subspaces. We use a hierarchical genetic algorithm to minimize a model selection criterion, which allows us to automatically and simultaneously determine the nonlinear parameters: spread, centre locations, and number of RBFs. The coefficients of the linear model (linear parameters) are computed solving a set of linear equations using the least-squares method through Singular Value Decomposition (SVD) method. In order to validate the efficacy of the proposed approach, we perform an experimental study with several tests on smooth functions benchmarks. The comparative analysis with two successful methods based on RBF networks has been included.

2. Background

2.1. Radial Basis Functions. RBFs are a particular class of functions whose value depends only on the distance from the centre. Specifically, their response decreases (*local response*) or increases (*global response*) monotonically with respect to

TABLE 1: Types of RBFs commonly used.

Name	Function
Linear	$\phi(x) = r$
Cubic spline	$\phi(x) = r^3$
Thin-plate spline	$\phi(x) = r^2 \log(r)$
Multiquadric	$\phi(x) = \sqrt{1 + r^2}$
Gaussian	$\phi(x) = \exp^{-r^2/\sigma^2}$

the distance from a central point. In general, a radial basis function is represented by the following equation:

$$\Phi(x) = \phi(|x - c|), \quad (1)$$

where $|\cdot|$ denotes the norm used to measure the distance between any point x and the centre c of the basis function and $\phi(\cdot)$ is a specific type of RBF. Usually, the norm used is Euclidean distance [19], and the types of RBFs commonly used are expressed in Table 1, where $r = |x - c|$.

2.2. Linear Models. A linear model for a function $y(x)$ can be expressed as a linear combination of a set of m basis functions as

$$f(x) = \sum_{i=1}^m \lambda_i \Phi_i(x), \quad (2)$$

where λ_i are coefficients and $\Phi_i(x)$ are the basis functions of the model. In a linear model, the basis functions and any parameter which they might contain are fixed. Its ability to fit different functions is derived from the freedom to choose different coefficient values. Otherwise, if the basis functions or its parameters are not fixed, then the model is a nonlinear model. Any set of basis functions can be used as basis set; however, models containing only basis functions from a particular class have a special interest [20].

2.3. Hierarchical Genetic Algorithms. HGA is a revised version of a genetic algorithm, which is an optimization scheme based on the biological evolution. Unlike the traditional genetic algorithm where the genotype structure is fixed, the chromosome in the HGA does not have these restrictions. The HGA has a bicoded chromosome scheduled in a hierarchical mode, which consists of two kinds of genes: control and parametric genes [21].

Located at the first level, the control genes are encoded as binary digits, while parametric genes at the next levels are encoded as real numbers generally, and their activation depends upon the value of associated control genes. Specifically, the control genes activate or inactivate the parametric genes; this is particularly important to determine the phenotype with different lengths within the same chromosomal representation. Hence, HGA will search over a larger search space and converge to the right solution with an increasingly high degree of accuracy.

2.4. Curve Fitting with RBFs. The goal of curve fitting is to find the best estimate of the unknown functional relationship

f from a set of n pairs of measurements (x_i, y_i) , where y_i is the i th observation at the point x_i . This relationship can be expressed as

$$y_i = f(x_i) + \epsilon_i, \quad (3)$$

where $i = 1, \dots, n$ and the term ϵ_i represents the i th element of a vector with zero-mean random error.

In this study, we assume that f is a smooth function, which can be properly approximated on the interval $[a, b]$ by a linear combination of a set of m basis functions. The linear model is constructed from m basis functions placed at certain locations considering that $C = \{c_1, \dots, c_{\max}\}$ is a set of max locations placed along the domain of the independent variable x and that the model has at most max basis functions. Under these assumptions, the function f can be written as the following linear combination:

$$f(x) = \sum_{i=1}^m \lambda_i \phi_i(|x - c_i|), \quad (4)$$

where $m \leq \max$ and λ_i is the coefficient associated with i th basis function Φ_i , which is defined on the set of centres C .

If the type of basis function Φ is specified beforehand, we can define f completely if we find the number m of basis functions and the parameters associated with them and then we calculate the model coefficients λ_i .

3. Automatic Curve Fitting Using RBF and HGA

In this study, we use a typical radial function with local response, the Gaussian function. The popular Gaussian RBF is radially symmetric with a maximum value at central point c . It is selected for maximal trend sensing with minimal parameter representations for function approximation [22], where such parameters are the centre c when the maximum value occurs (frequently called mean), and the width of the function σ is referred to as spread.

When the type of basis functions has been specified, our method applies the HGA to determine the number m of the basis functions (model structure) and the centre locations and spreads (basis parameters) simultaneously. Then, the coefficients of linear model are calculated using the least-squares method through the Singular Value Decomposition (SVD) method. To find the best estimate \hat{f} of the unknown functional relationship f by using HGA, the fitness function is defined to minimize a model selection criterion. Next, we specified the main features of HGA.

3.1. Chromosome Encoding. To simultaneously determine the number of the basis functions, the centre locations, and the spread values, we propose a two-level bicoded description, where each bit in the binary chromosome carries the implication of presence ("1") or absence ("0") of a specific basis function in the solution. The real chromosome represents the magnitude of the spread parameters of the basis functions. We represent the chromosome of an individual as

$$\theta = \{b_1, \dots, b_{\max}, \sigma_1, \dots, \sigma_{\max}\}, \quad (5)$$

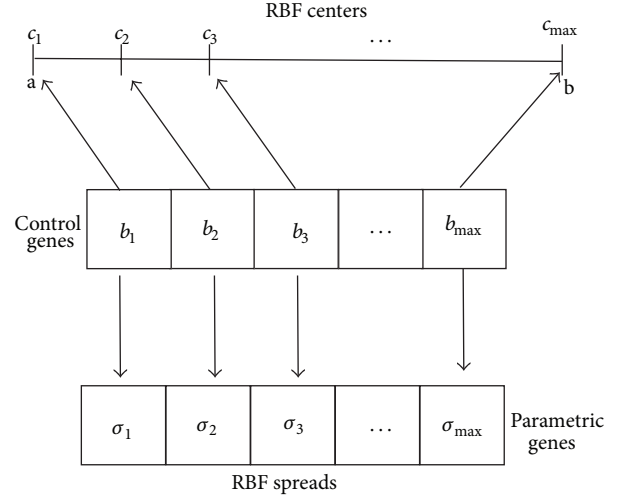


FIGURE 1: Hierarchical structure of a chromosome.

where each bit b_i is a control gene and σ_i is a real value. Each control bit simultaneously enables or disables one basis function ϕ and its associated parameter σ . In the present paper, each control gene b_i corresponds to one basis function located at a specific centre c_i . Therefore, if the i th control gene is "1," the i th basis function located in the i th centre and with spread σ_i exists in the candidate solution. We use a fixed length binary string to represent the max number of the basis functions ϕ . The general structure of a chromosome is graphically shown in Figure 1.

3.2. Fitness Function. To determine the model structure, as well as the basis parameters, we propose a fitness function based on the Akaike's Information Criterion (AIC). Originally, for parametric problems, this criterion was developed as a measure of the fitness of the model and the complexity of the model. The AIC is given by

$$\text{AIC}(\hat{f}) = n \log \sum_{i=1}^n \{y_i - \hat{f}(x_i)\}^2 + 2p, \quad (6)$$

where n is the number of points in \hat{f} and \hat{f} is the estimation of f . The term p is the number of the model parameters. In AIC, the residual sum of squares in the first term is used as a measure of the deviation of the estimated function \hat{f} from f , and the second term is a penalty for increasing the number of parameters p . In this study, the number of parameters consists of c , σ , and λ for each basis function. As a result, the number of parameters p is 3 times the number of basis function m , and thus $p = 3m$.

3.3. Reproduction Process and Genetic Operators. Reproduction is the process whereby the best fit individual in the population receives a corresponding large number of copies in the next generation. We use the roulette wheel method as a reproduction process. In this method, each individual in the population has a roulette wheel slot sized in proportion to its fitness value. This selection strategy favors the most fit

individuals, but it gives a chance to the less fit individuals to reproduce. Otherwise, in order to prevent premature convergence, it tries to keep the selection pressure relatively constant over the entire evolution process by using the sigma scaling method [23], which is calculated according to

$$F_{\text{new}} = \begin{cases} F_{\text{act}} - (\bar{F} - c \cdot \sigma) & \text{if } (F_{\text{act}} > \bar{F} - c \cdot \sigma) \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

where F_{new} is the new scaled fitness value, F_{act} is the current fitness value, \bar{F} is the average fitness, σ is the standard deviation of the population, and c is a constant that allows it to control the selection pressure. In addition, elitism is used in order to keep the elite individuals in the next population to prevent losing the best solution found. After a collection of good chromosomes is selected, they exchange information using the crossover operator.

The crossover process selects genes from the parent chromosomes and creates new offspring. In the uniform crossover scheme, two parents are chosen to recombine into a new individual. Each bit of the new individual is selected from one of the parents depending on a fixed probability. We use the uniform crossover operator for both the control and the parametric genes with the same probability.

In the mutation process, there are also two types of mutations. The bit mutation method is applied over the control genes. With this method, each bit is inverted or not depending on the mutation probability. For the parametric genes mutation each numeric value γ is changed depending on the same mutation probability according to

$$\gamma_i = \gamma_i + \delta (\text{rand} - 0.5), \quad (8)$$

where δ is the maximum increment or decrement of the real value and rand is a function that generates a random value between 0 and 1.

4. Experimental Results

In order to evaluate the effectiveness of the proposed method (HGA), we carried out numerical simulations on five test functions, which are mathematically defined by the equations in Table 2. Our experimental study includes a spatially inhomogeneous smooth function, that is, with a sharp peak (f_3), an oscillating function but not periodic (f_4), and a function with a discontinuity (f_5). Note that these test functions have been used in previous studies [24–28] and they could be considered as a benchmark in smooth functions. Furthermore, we present a comparison with the well-known methods based on radial basis neural networks (RBNN) and general regression neural networks (GRNN).

The problem addressed in this paper is the automatic curve fitting given a set of noisy data points. Thus, the experimental setup was designed to study the effects of varying the number of samples and noise levels on the curve fitting task. The idea is that each test function is evaluated at n design points x_i uniformly distributed over the interval $[0, 1]$. Here, a zero-mean normal noise with a known σ is added to

TABLE 2: Mathematical definition of test functions.

Test functions
$f_1(x) = \sin^3(2\pi x^3)$
$f_2(x) = (4x - 2) + 2\exp(-16(4x - 2)^2)$
$f_3(x) = \sin(x) + 2\exp(-30x^2)$
$f_4(x) = \sin(2(4x - 2)) + 2\exp(-16(4x - 2)^2)$
$f_5(x) = 10(4x - 2)/(1 + 100(4x - 2)^2)$

TABLE 3: Setup for the HGA.

Parameter	Value
Crossover probability	0.90
Mutation probability	0.15
Number of elite individuals	11
Generations	3000
Population size	300

the data set in order to have a specific signal-to-noise ratio (SNR), defined as $\text{SD}(f)/\sigma$.

For each function, a collection of 101 noisy data sets is generated from each of the different configurations. In this study we use three different SNR values (2, 3, and 4) and three different sized samples for n (50, 100, and 200), and hence there are altogether 9 different configurations. Thus, we tested our method over a total of 909 data sets for each test function. For completeness, in test function, 3 design points x_i are uniformly distributed in $[-2, 2]$ and then they are scaled to the interval $[0, 1]$. Finally, the proposed method and the RBNN and GRNN methods were used to estimate the test functions.

For the HGA, we use a maximal number of basis function ($m = 30$) and C is defined as a subset of design points. At the beginning, the population is randomly initialized, each control gene b_i is randomly selected from $[0, 1]$, and each parametric gene σ_i is calculated as a random real number defined over the range $[0.1, 1.0]$. The parameters used for the HGA were tuned experimentally as in [29] and are summarized in Table 3.

To compare and evaluate qualitatively the performance of the methods, the best estimates obtained using each method for the different configurations were obtained. These best estimates correspond to the minimal MSE value obtained using each method from the collection of 101 noisy data sets generated from each specific configuration. For practical reasons, we present only the results of 1 of the 9 configurations from the experimental study. These results are shown in Figure 2 and correspond to data sets with 50 samples and $\text{SNR} = 3$. The figure suggests that the HGA has more ability to adapt to discontinuities and sharp peaks in curves despite the level of noise and the small number of samples in contrast to GRNN and RBNN methods. These methods generate less smooth curves with overfitting in some sections. Similar results have been obtained for the other configurations.

For each simulated data set, the numerical measure that was used to quantitatively evaluate the quality of

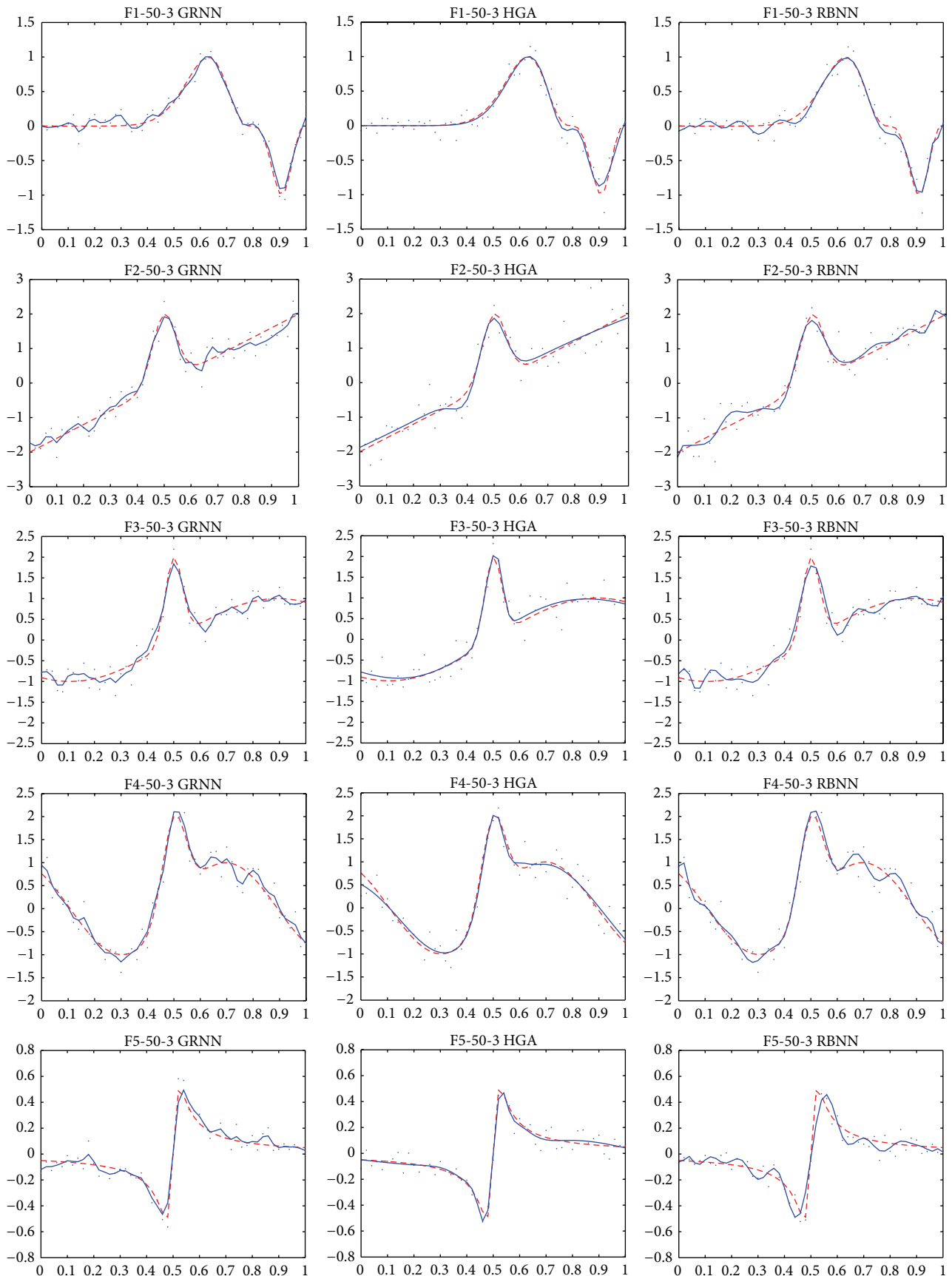


FIGURE 2: Data and estimates correspond to the best function estimated from the experimental setup with 50 samples and SNR = 3. The columns correspond from left to right to the GRNN, HGA, and RBNN methods, respectively (solid line estimated function; dashed line real function). The rows correspond from top to bottom to test functions 1 to 5.

TABLE 4: Mean of the computed MSEs with estimated *standard errors* based on 101 data sets with 50 samples.

Test	Method	SNR = 2	SNR = 3	SNR = 4
1	RBNN	0.019261 ± 0.000658	0.008170 ± 0.000246	0.004879 ± 0.000139
	GRNN	0.016372 ± 0.000510	0.007463 ± 0.000218	0.004504 ± 0.000133
	HGA	0.016149 ± 0.000704	0.006008 ± 0.000268	0.003776 ± 0.000160
2	RBNN	0.159658 ± 0.005110	0.076066 ± 0.002543	0.045209 ± 0.001382
	GRNN	0.133178 ± 0.003940	0.063440 ± 0.001958	0.038152 ± 0.001054
	HGA	0.104318 ± 0.006079	0.052129 ± 0.002207	0.030713 ± 0.001295
3	RBNN	0.084509 ± 0.002430	0.043169 ± 0.001251	0.026841 ± 0.000652
	GRNN	0.069057 ± 0.001997	0.033152 ± 0.000889	0.019380 ± 0.000496
	HGA	0.054656 ± 0.002430	0.026877 ± 0.001443	0.012641 ± 0.000754
4	RBNN	0.072612 ± 0.002118	0.033507 ± 0.001000	0.018269 ± 0.000509
	GRNN	0.061318 ± 0.001749	0.027931 ± 0.000723	0.016344 ± 0.000420
	HGA	0.055247 ± 0.002197	0.027116 ± 0.001242	0.013517 ± 0.000612
5	RBNN	0.007108 ± 0.000104	0.005094 ± 0.000051	0.004412 ± 0.000028
	GRNN	0.004042 ± 0.000102	0.002322 ± 0.000050	0.001768 ± 0.000033
	HGA	0.004424 ± 0.000196	0.001990 ± 0.000125	0.001046 ± 0.000076

the estimated function \hat{f} is the mean square error (MSE) given by

$$\text{MSE}(\hat{f}) = \frac{1}{n} \sum_{i=1}^n \{\hat{f}(x_i) - f(x_i)\}^2. \quad (9)$$

The smaller $\text{MSE}(\hat{f})$ value corresponds to a better quality of \hat{f} . To facilitate the comparison of the distribution of MSE values among the different methods, we use the boxplot. In descriptive statistics, a boxplot is a convenient visual representation that partitions a data distribution through their quartiles. Boxplots of the $\text{MSE}(\hat{f})$ values for the different configurations are shown in Figures 3, 4, and 5. In Figure 3, the boxplots of MSE values for data sets with 50 samples are shown. In each panel, the boxplots correspond to a specific configuration. The columns correspond from left to right to noise ratios 2, 3, and 4, respectively. The rows correspond from top to bottom to the test functions 1 to 5. Similarly, Figures 4 and 5 show the boxplots of MSE values for data sets with 100 and 200 samples, respectively.

In Figure 3, it can be observed that in most cases the medians of the MSE values from HGA are lower than those resulting from the other methods, although there is no significant difference among the results. For example, in test function 5 with 50 samples and SNR = 2 (bottom-left of the figure), the GRNN performs a slightly smaller median of the MSE value than the HGA method, but the difference is not significant. In this case, the upper whisker of the MSE values for the HGA method is considerably bigger than those from the other methods, which implies a bigger dispersion for 25% of their MSE values. The most significant difference among the boxplots can be observed for test function 5 with 50 samples and SNR = 4 (bottom-right of the figure). In this case, the upper quartile of the MSE values for the HGA is under the first quartile of those from GRNN, which implies that 75% of MSE values from HGA are lower than 75% of MSE values from GRNN. In most cases in Figures 4 and 5,

the upper quartile of the MSE values for the HGA is under the first quartile of those from the RBNN or GRNN methods. Although the whiskers of the boxplots from HGA have a different size, in all cases the lower whisker is smaller than the upper whisker, which implies an asymmetric distribution of the MSE values. Although the distribution of the MSE values for the HGA method is considerably bigger than those from the other methods, in most cases they are lower than those from the other methods.

Additionally, we compute the mean and standard error of mean for MSE values from all methods. These results are summarized in Tables 4, 5, and 6, corresponding to results for data sets with 50, 100, and 200 samples, respectively. In each table, the results corresponding to noise ratios 2, 3, and 4 are included. In Tables 7, 8, and 9 we include the results corresponding to parameters for the best function estimated from the experimental setup with 50 samples and SNR = 2, 3, and 4, respectively.

To clarify the effects of varying the number of samples and noise levels on the curve fitting task by using the HGA method, we include Figures 6 and 7. The first figure shows the best estimates obtained using HGA method for data sets from left to right with 50, 100, and 200 samples and SNR = 3. The second figure shows the best estimates obtained by using the HGA method for data sets with 50 samples and SNR = 2, 3, and 4, from left to right, respectively. In both figures, the rows correspond from top to bottom to the test functions 1 to 5, respectively. The performance of the methods significantly improves if the noise level is reduced or the number of samples increases, as shown in Tables 4, 5, and 6, and Figures 7 and 6. This comes as no surprise, because, as we increase the number of samples or decrease the noise level, we have more information to build the curve.

Finally, it could be assumed that the success of our approach is due to the simultaneous global search performed over all the parameters of the linear model. This shows the potential of HGA to handle the loss of information and high

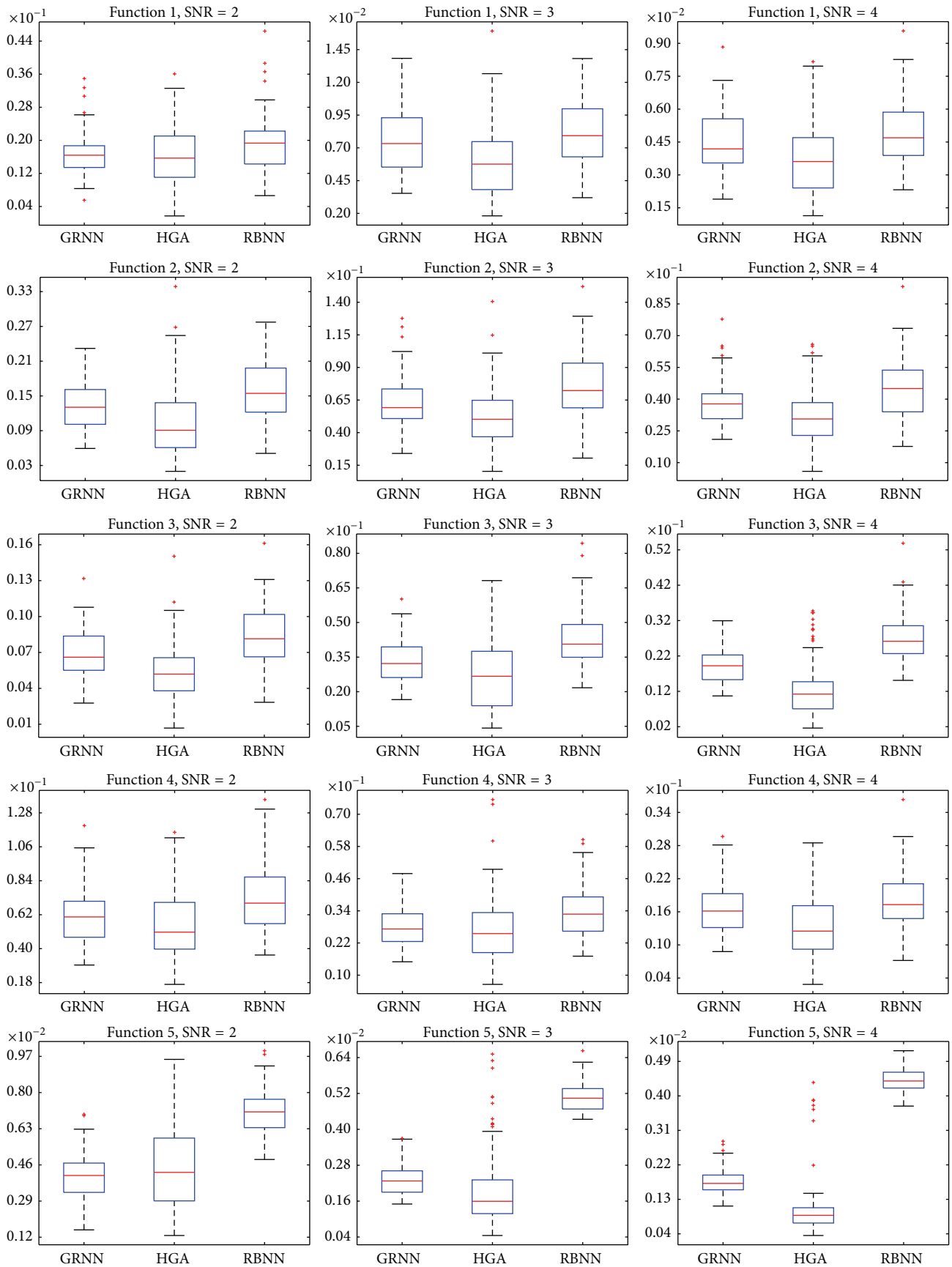


FIGURE 3: *Boxplots* of MSE values for data sets with 50 samples. In each panel, the *boxplots* correspond to the algorithms tested for a specific SNR. From *left to right* the columns correspond to SNR = 2, 3, and 4, respectively. The rows correspond from *top to bottom* to the test functions 1 to 5.

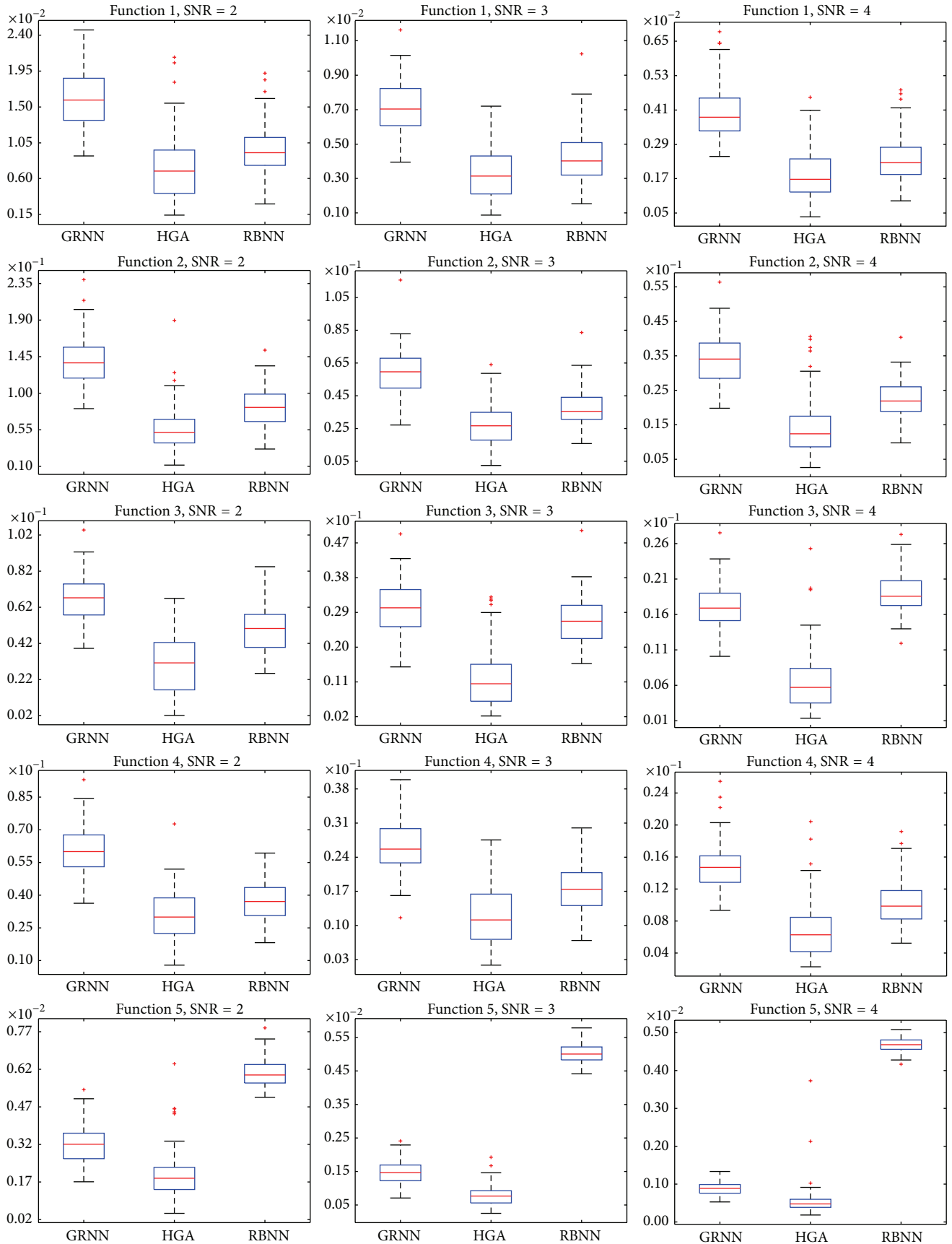


FIGURE 4: *Boxplots* of MSE values for data sets with 100 samples. In each panel, the *boxplots* correspond to the algorithms tested for a specific SNR. From *left to right* the columns correspond to SNR = 2, 3, and 4, respectively. The rows correspond from *top to bottom* to the test functions 1 to 5.

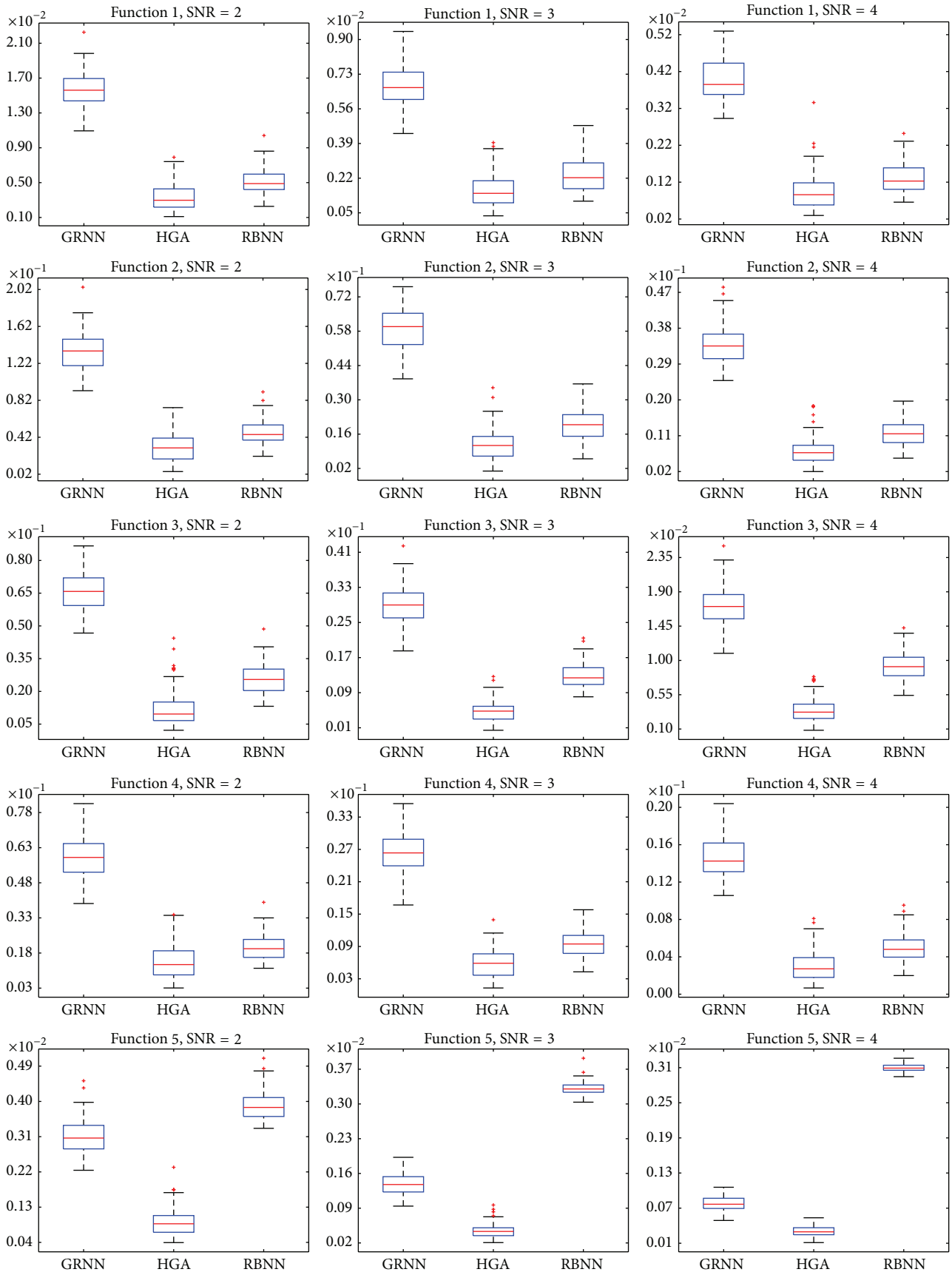


FIGURE 5: *Boxplots* of MSE values for data sets with 200 samples. In each panel, the *boxplots* correspond to the algorithms tested for a specific SNR. From *left to right* the columns correspond to SNR = 2, 3, and 4, respectively. The rows correspond from *top to bottom* to the test functions 1 to 5.

TABLE 5: Mean of the computed MSEs with estimated *standard errors* based on 101 data sets with 100 samples.

Test	Method	SNR = 2	SNR = 3	SNR = 4
1	RBNN	0.009500 ± 0.000308	0.004182 ± 0.000141	0.002413 ± 0.000082
	GRNN	0.016042 ± 0.000341	0.007198 ± 0.000150	0.004040 ± 0.000094
	HGA	0.007340 ± 0.000399	0.003269 ± 0.000150	0.001877 ± 0.000087
2	RBNN	0.084075 ± 0.002368	0.037623 ± 0.001145	0.022545 ± 0.000542
	GRNN	0.139300 ± 0.002681	0.059419 ± 0.001318	0.034220 ± 0.000686
	HGA	0.056166 ± 0.002621	0.027498 ± 0.001296	0.014220 ± 0.000814
3	RBNN	0.049175 ± 0.001244	0.026765 ± 0.000574	0.018885 ± 0.000278
	GRNN	0.066580 ± 0.001346	0.030152 ± 0.000643	0.017198 ± 0.000292
	HGA	0.029524 ± 0.001533	0.011812 ± 0.000723	0.006706 ± 0.000410
4	RBNN	0.037467 ± 0.000964	0.017486 ± 0.000488	0.010188 ± 0.000284
	GRNN	0.060238 ± 0.001136	0.026240 ± 0.000548	0.014598 ± 0.000295
	HGA	0.030636 ± 0.001139	0.012113 ± 0.000589	0.006764 ± 0.000346
5	RBNN	0.006034 ± 0.000051	0.005044 ± 0.000030	0.004677 ± 0.000020
	GRNN	0.003228 ± 0.000070	0.001473 ± 0.000030	0.000889 ± 0.000017
	HGA	0.001964 ± 0.000088	0.000779 ± 0.000030	0.000549 ± 0.000039

TABLE 6: Mean of the computed MSEs with estimated *standard errors* based on 101 data sets with 200 samples.

Test	Method	SNR = 2	SNR = 3	SNR = 4
1	RBNN	0.005115 ± 0.000138	0.002373 ± 0.000078	0.001352 ± 0.000043
	GRNN	0.015742 ± 0.000201	0.006733 ± 0.000101	0.003975 ± 0.000055
	HGA	0.003302 ± 0.000211	0.001568 ± 0.000109	0.000949 ± 0.000067
2	RBNN	0.046566 ± 0.001347	0.020088 ± 0.000603	0.011702 ± 0.000313
	GRNN	0.135356 ± 0.002033	0.059186 ± 0.000828	0.033560 ± 0.000464
	HGA	0.030766 ± 0.002031	0.011812 ± 0.000853	0.007242 ± 0.000488
3	RBNN	0.025609 ± 0.000663	0.012887 ± 0.000282	0.009299 ± 0.000179
	GRNN	0.065318 ± 0.000899	0.029221 ± 0.000416	0.017189 ± 0.000246
	HGA	0.012078 ± 0.001150	0.004865 ± 0.000327	0.003424 ± 0.000233
4	RBNN	0.020309 ± 0.000532	0.009341 ± 0.000243	0.005035 ± 0.000147
	GRNN	0.058907 ± 0.000878	0.026362 ± 0.000377	0.014725 ± 0.000215
	HGA	0.014401 ± 0.001042	0.005852 ± 0.000367	0.003016 ± 0.000222
5	RBNN	0.003896 ± 0.000026	0.003310 ± 0.000013	0.003097 ± 0.000007
	GRNN	0.003088 ± 0.000043	0.001378 ± 0.000022	0.000777 ± 0.000011
	HGA	0.000912 ± 0.000050	0.000444 ± 0.000020	0.000310 ± 0.000013

noise level. The average execution time for the methods based on neural networks was 0.29 seconds for experiments with 200 samples, compared to 132 seconds when using HGA. However, this drawback can be solved if we consider a parallel implementation of the model HGA. Our algorithm was implemented in C++ language. All methods were executed on a PC using an Intel Core i7 processor running at 2.2 GHz with 8 GB of RAM.

5. Conclusions

In this paper, we have proposed an efficient hierarchical genetic algorithm to tackle the automatic curve fitting problem. The method introduces a novel hierarchical gene structure for the chromosomal representation that permits the optimization of all free degrees of a linear model based on RBFs. To be more specific, our algorithm finds the best

linear model using the fewest basis functions, optimal radii, and centre locations, and coefficients of the model simultaneously. The method does not require subjective parameters such as smooth factors or centre locations to perform the solution. On the other hand, the main drawback is the computational time required; however, it can be solved using parallel computing in order to reduce costs. This study focused on smooth functions, but it can be extended to approximate unsmooth functions. The algorithm was tested on several benchmark functions and compared to two successful methods based on RBF neural networks. Simulation results show that the proposed method presents qualitatively and quantitatively better results than the methods based on RBF neural networks because the method can handle the loss of information and high noise level more effectively. One of the attractive features of our method is that it can be easily extended to surface fitting.

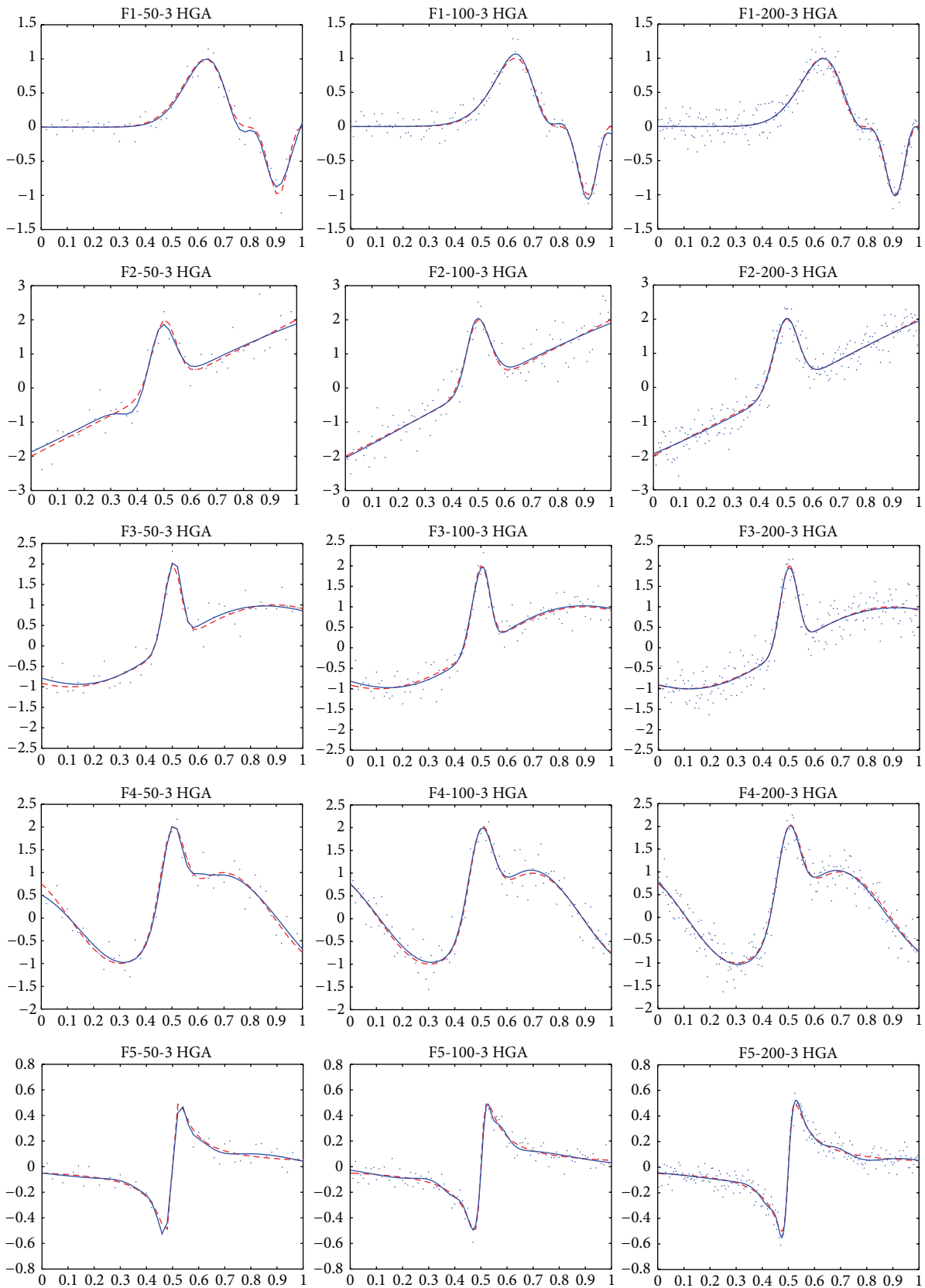


FIGURE 6: Effect of varying the number of samples on the curve fitting task by using the HGA method. Data and estimates correspond to the best function estimated from the experimental setup with SNR = 3 (solid line estimated function; dashed line real function). From left to right the columns correspond to data sets with 50, 100, and 200 samples, respectively. The rows correspond from top to bottom to the test functions 1 to 5.

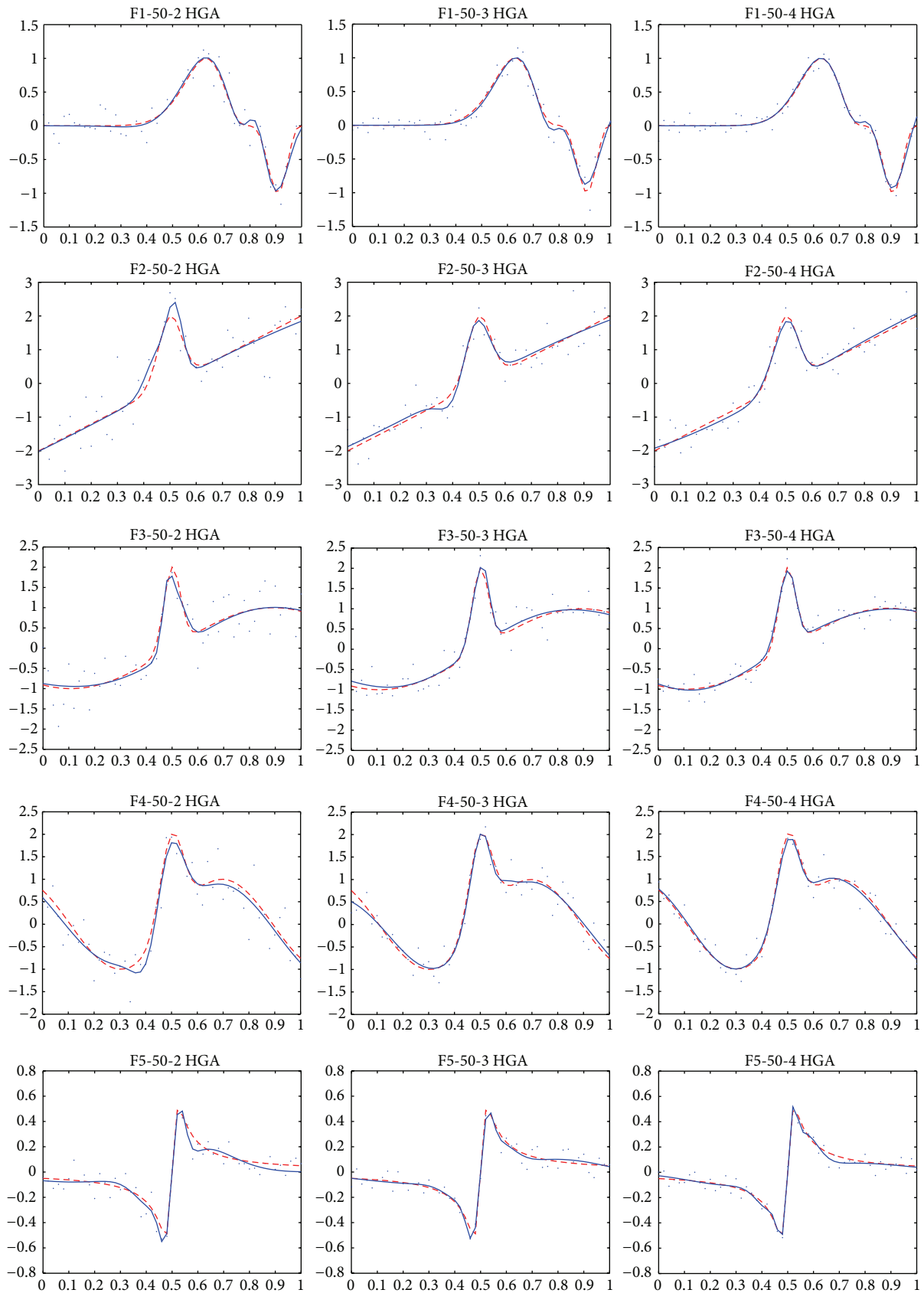


FIGURE 7: Effect of varying the signal-to-noise ratio on the curve fitting task by using the HGA method. Data and estimates correspond to the best function estimated from the experimental setup for data sets with 50 samples (solid line estimated function; dashed line real function). From left to right the columns correspond to SNR = 2, 3, and 4, respectively. The rows correspond from top to bottom to the test functions 1 to 5.

TABLE 7: Basis functions corresponding to the best function estimated from the experimental setup with 50 samples and SNR = 2.

Test	Basis	λ	C	σ
1	Φ_1	2.624918	0.689655	0.175274
	Φ_2	1.306060	0.827586	0.057716
	Φ_3	-3.316849	0.896552	0.278344
	Φ_4	2.739275	1.000000	0.126029
2	Φ_1	-18.659943	0.103448	2.150076
	Φ_2	1.013608	0.448276	0.057623
	Φ_3	2.113588	0.517241	0.043472
	Φ_4	18.093719	0.620690	2.118272
3	Φ_1	-1.048773	0.206897	0.533407
	Φ_2	1.155269	0.482759	0.027828
	Φ_3	1.225578	0.517241	0.049997
	Φ_4	1.236929	0.827586	0.423159
4	Φ_1	-9.654642	0.344828	0.409970
	Φ_2	-1.730152	0.448276	0.083107
	Φ_3	3.042676	0.482759	0.075049
	Φ_4	11.017281	0.551724	0.656271
	Φ_5	-7.058442	0.965517	0.455500
5	Φ_1	-0.079952	0.137931	0.358084
	Φ_2	-0.374830	0.448276	0.108851
	Φ_3	-0.580771	0.482759	0.035342
	Φ_4	0.715895	0.517241	0.039981
	Φ_5	0.225844	0.586207	0.193723

TABLE 8: Basis functions corresponding to the best function estimated from the experimental setup with 50 samples and SNR = 3.

Test	Basis	λ	C	σ
1	Φ_1	2.581205	0.724138	0.162365
	Φ_2	1.355051	0.827586	0.064088
	Φ_3	-9.424124	0.862069	0.168524
	Φ_4	6.909147	0.896552	0.168821
2	Φ_1	-17.234314	0.137931	1.933229
	Φ_2	-1.310536	0.448276	0.086132
	Φ_3	2.834948	0.482759	0.072595
	Φ_4	16.774952	0.586207	1.913092
3	Φ_1	-1.116542	0.206897	0.391998
	Φ_2	1.133581	0.482759	0.037636
	Φ_3	1.355708	0.517241	0.032508
	Φ_4	1.067411	0.793103	0.463377
4	Φ_1	-3.792799	0.344828	0.292119
	Φ_2	2.455837	0.517241	0.070303
	Φ_3	-1.031267	0.551724	0.044479
	Φ_4	5.217404	0.655172	0.611903
	Φ_5	-4.459106	1.000000	0.570569
5	Φ_1	-0.209363	0.448276	0.097197
	Φ_2	-0.561304	0.482759	0.042001
	Φ_3	0.580368	0.517241	0.035747
	Φ_4	0.254328	0.551724	0.094056
	Φ_5	-0.512819	0.724138	0.508370
	Φ_6	0.607906	0.758621	0.402507

TABLE 9: Basis functions corresponding to the best function estimated from the experimental setup with 50 samples and SNR = 4.

Test	Basis	λ	C	σ
1	Φ_1	2.019124	0.689655	0.158368
	Φ_2	1.461036	0.827586	0.067736
	Φ_3	-2.826506	0.862069	0.201430
	Φ_4	1.857911	1.000000	0.126618
2	Φ_1	-8.652872	0.275862	1.484126
	Φ_2	1.139507	0.482759	0.072935
	Φ_3	0.928305	0.517241	0.052756
	Φ_4	8.930352	0.896552	1.567421
3	Φ_1	-1.052161	0.137931	0.321876
	Φ_2	1.214086	0.482759	0.041275
	Φ_3	1.065797	0.517241	0.033858
	Φ_4	0.987050	0.896552	0.404330
4	Φ_1	-4.415701	0.310345	0.292465
	Φ_2	5.736092	0.448276	0.605733
	Φ_3	1.014713	0.482759	0.056770
	Φ_4	0.868994	0.517241	0.047533
	Φ_5	-3.297666	0.931035	0.889548
5	Φ_1	-0.160610	0.344828	0.281685
	Φ_2	-0.272302	0.448276	0.081592
	Φ_3	-0.413769	0.482759	0.029738
	Φ_4	0.464189	0.517241	0.023852
	Φ_5	0.333063	0.551724	0.080594
	Φ_6	0.101939	0.620690	0.396612

Given the performance characteristics of the proposed approach, our future work will be to apply this method to fit real experimental data. We are interested in extending our approach to surface fitting, to experiment with variable length chromosomes and different basis functions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors acknowledge the support of CONACYT Mexico, DAIP of the University of Guanajuato, the Information Technology Laboratory of CINVESTAV, Tamaulipas, and ITESI, Mexico. Also, they gratefully acknowledge the help of Dr. Carlos Montoro Rubin to improve the English language of their paper.

References

[1] K. Das and S. C. Mishra, "Non-invasive estimation of size and location of a tumor in a human breast using a curve fitting technique," *International Communications in Heat and Mass Transfer*, vol. 56, pp. 63–70, 2014.

[2] J.-Y. Yeh and J. C. Fu, "A hierarchical genetic algorithm for segmentation of multi-spectral human-brain MRI," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1285–1295, 2008.

- [3] M. Zhang, W. Yan, C. Yuan, D. Wang, and X. Gao, "Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines," *Science China Information Sciences*, vol. 54, no. 7, pp. 1407–1418, 2011.
- [4] A. Gálvez and A. Iglesias, "A new iterative mutually coupled hybrid GA-PSO approach for curve fitting in manufacturing," *Applied Soft Computing Journal*, vol. 13, no. 3, pp. 1491–1504, 2013.
- [5] K. Zhang, C.-M. Yuan, and X.-S. Gao, "Efficient algorithm for time-optimal feedrate planning and smoothing with confined chord error and acceleration," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9, pp. 1685–1697, 2013.
- [6] J. Wu, H. Zhou, X. Tang, and J. Chen, "Implementation of CL points preprocessing methodology with NURBS curve fitting technique for high-speed machining," *Computers and Industrial Engineering*, vol. 81, pp. 58–64, 2015.
- [7] W. Zheng, P. Bo, Y. Liu, and W. Wang, "Fast B-spline curve fitting by L-BFGS," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 448–462, 2012.
- [8] A. Gálvez and A. Iglesias, "Firefly algorithm for explicit B-spline curve fitting to data points," *Mathematical Problems in Engineering*, vol. 2013, Article ID 528215, 12 pages, 2013.
- [9] A. Gálvez, A. Iglesias, A. Avila, C. Otero, R. Arias, and C. Manchado, "Elitist clonal selection algorithm for optimal choice of free knots in B-spline data fitting," *Applied Soft Computing Journal*, vol. 26, pp. 90–106, 2015.
- [10] Z. Yang, L.-Y. Shen, C.-M. Yuan, and X.-S. Gao, "Curve fitting and optimal interpolation for CNC machining under confined error using quadratic B-splines," *Computer-Aided Design*, vol. 66, pp. 62–72, 2015.
- [11] C. E. Torres and L. A. Barba, "Fast radial basis function interpolation with Gaussians by localization and iteration," *Journal of Computational Physics*, vol. 228, no. 14, pp. 4976–4999, 2009.
- [12] Q. Wang, W. Shi, and P. M. Atkinson, "Sub-pixel mapping of remote sensing images based on radial basis function interpolation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 1–15, 2014.
- [13] S.-E. Roh, J. H. Choi, and T. Kim, "A new action potential classifier using 3-Gaussian model fitting," *Neurocomputing*, vol. 71, no. 16–18, pp. 3631–3634, 2008.
- [14] X. Wen, D. Zhang, Y. Qian, J. Li, and N. Fei, "Improving the peak wavelength detection accuracy of Sn-doped, H₂-loaded FBG high temperature sensors by wavelet filter and Gaussian curve fitting," *Sensors and Actuators A: Physical*, vol. 174, no. 1, pp. 91–95, 2012.
- [15] H.-W. Lee, H.-J. Park, J.-H. Lee, and M. Song, "Accuracy improvement in peak positioning of spectrally distorted fiber Bragg grating sensors by Gaussian curve fitting," *Applied Optics*, vol. 46, no. 12, pp. 2205–2208, 2007.
- [16] C. Liu, D. Zheng, A. Murray, and C. Liu, "Modeling carotid and radial artery pulse pressure waveforms by curve fitting with Gaussian functions," *Biomedical Signal Processing and Control*, vol. 8, no. 5, pp. 449–454, 2013.
- [17] Kavita and N. Rajpal, "Comparative analysis of feed forward and radial basis function neural networks for the reconstruction of noisy curves," in *Proceedings of the International Conference on Optimization, Reliability, and Information Technology (ICROIT '14)*, pp. 353–358, Faridabad, India, February 2014.
- [18] Q. P. Ha, H. Wahid, H. Duc, and M. Azzi, "Enhanced radial basis function neural networks for ozone level estimation," *Neurocomputing*, vol. 155, pp. 62–70, 2015.
- [19] H. Rocha, "On the selection of the most adequate radial basis function," *Applied Mathematical Modelling*, vol. 33, no. 3, pp. 1573–1583, 2009.
- [20] M. J. L. Orr, "Introduction to radial basis function networks," Tech. Rep., Center for Cognitive Science, University of Edinburgh, 1996.
- [21] K. F. Man, K. S. Tang, and S. Kwong, "Hierarchical genetic algorithm," in *Genetic Algorithms*, pp. 65–74, 1999.
- [22] K. Ganapathy, V. Vaidehi, and J. B. Chandrasekar, "Optimum steepest descent higher level learning radial basis function network," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8064–8077, 2015.
- [23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Automatic Curve Fitting Based on RBF and HGA 23, Addison-Wesley Professional, 1989.
- [24] D. G. Denison, B. K. Mallick, and A. F. Smith, "Automatic Bayesian curve fitting," *Journal of the Royal Statistical Society B. Statistical Methodology*, vol. 60, no. 2, pp. 333–350, 1998.
- [25] I. DiMatteo, C. R. Genovese, and R. E. Kass, "Bayesian curve-fitting with free-knot splines," *Biometrika*, vol. 88, no. 4, pp. 1055–1071, 2001.
- [26] A. Gálvez and A. Iglesias, "Efficient particle swarm optimization approach for data fitting with free knot B-splines," *Computer-Aided Design*, vol. 43, no. 12, pp. 1683–1692, 2011.
- [27] J. Pittman, "Adaptive splines and genetic algorithms," *Journal of Computational and Graphical Statistics*, vol. 11, no. 3, pp. 615–638, 2002.
- [28] T. C. Lee, "Smoothing parameter selection for smoothing splines: a simulation study," *Computational Statistics and Data Analysis*, vol. 42, no. 1-2, pp. 139–148, 2003.
- [29] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero, and H. Rostro-Gonzalez, "Hierarchical genetic algorithm for B-spline surface approximation of smooth explicit data," *Mathematical Problems in Engineering*, vol. 2014, Article ID 706247, 11 pages, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

