Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 6, No. 3, August 2015), Pages: 23-38
www.jacr.iausari.ac.ir

A Novel Approach for Discrimination Magnetizing Inrush Current and Internal Fault in Power Transformers Based on Neural Network

Mehran Taghipour-Gorjikolaie *[1], Mohammad Yazdani-Asrami[2], S. Asghar Gholamian[2], and S. Mohammad Razavi[1]

1) Department of Electrical and Computer Engineering, University of Birjand, Birjand ,Iran
2) Department of Electrical and Computer Engineering, Babol University of Technology, Babol ,Iran
mtaghipour@birjand.ac.ir yazdani@stu.nit.ac.ir gholamian@nit.ac.ir smrazavi@birjand.ac.ir

Received: 2014/05/17; Accepted: 2014/09/01

Abstract

One of the major problems that may occur in the differential protection systems of power transformers is mal-operation of the protection relays in sake of internal fault detection, because of similarity between this current and inrush current. This paper presents a novel approach for discriminating inrush current from internal fault in power transformers based on Improved Gravitational Search Algorithm (IGSA). For this purpose, an Artificial Neural Network (ANN) which is trained by IGSA has been applied to discrete sample data of internal fault and inrush currents in the transformers. Results show that, the used approach can discriminate between these two kinds of phenomenon, very well and also, has high accuracy and excellent reliability, in addition, it has less computational burden and complexity.

Keywords: Activation Function, Artificial Neural Network, Differential Protection, Improved Gravitational Search Algorithm, Magnetizing Inrush Current, and Transformer Fault.

1. Introduction

Discrimination between the internal faults and magnetizing inrush current must be rapid and accurate in decision making. One of the most significant distinguishing characteristics of the magnetizing inrush current is the second harmonic content, which has a higher amount in inrush current than in different faults or normal currents. Many conventional transformer protections employ the second harmonic restraint approach. Different algorithms such as Discrete Fourier Transform (DFT), Least Square Method (LSM), Rectangular Transforms, Walsh Functions, Haar Functions, Kalman-Filter, Cosine Correlation and etcetera are used to calculate the current harmonic contents. The main disadvantages of this approach are that the second harmonic may also exist in some internal faults within transformer windings. This may be due to the presence of a shunt capacitor or the distributive capacitance in a long EHV transmission line to which the transformer may be connected. Also, the new low-loss amorphous core materials in modern power transformers may produce lower second and fifth harmonic contents in the inrush current [1-4]. In recent decades, modern approaches have been utilized to discriminate between inrush and fault currents. Some of these methods are such as; autoregressive process based on power spectrum of harmonics and harmonic characteristics based schemes [5], artificial neural networks [6-7], Neuro-Fuzzy approach [8], wavelet transform based schemes [9-11], combination of wavelet transform and fuzzy logic [12], leakage inductance based techniques [13], induced voltage based method [14], correlation transform [15], similarity degree between voltage and current [16], Bayes probability theory [17], non-stationary signal analysis using S-transform [18], space vector analysis of differential signals [19], statistical signal processing using Support Vector Machine [20], Hidden Markov Models [21] and morphological method [22].

In this paper, a novel method has been presented for discriminating inrush current from internal fault current in power transformers. For this purpose, Multi-Layered Feed-Forward Neural Network (MLFFNN) has been utilized, here, in order to train ANN a swarm-based algorithm which is called Improved Gravitational Search Algorithm (IGSA) has been run beside ANN. Since IGSA is optimizing algorithm and it can be seen that this algorithm can minimize non-linear or non-smooth functions and because this classifier has non-linear function, therefore, it is demonstrated that this algorithm can deal with train of ANN and also give best results as well as mentioned approach in other published literatures. It is mentionable that, input data of this ANN are discrete samples of magnetizing inrush current and internal fault current of simulated transformers.

2. Proposed Approach

In this section, the proposed method of this paper has been described:

2.1 Introduction of IGSA

The IGSA is classified as swarm based algorithm. In order to find extremum (minimum or maximum) value of fitness function, this algorithm searches a multi-dimensional search space. Indeed, it is inspired from similarity of cinematic and classical movement laws of masses in gravitational field. As a matter of fact, IGSA is defined using gravity laws and moving in artificial system in discrete time, in which, researcher agents are set of masses that each mass senses location and situation of other masses. Therefore, according to exchange force between masses, this algorithm can create artificial system [23-24]. The gravitational force between two agents is directly proportional to their masses and inversely to the square of the distance between them.

$$F = G\frac{M_1 M_2}{R^2} \tag{1}$$

In addition, R has been used instead of R^2, because according to the experimental results, R has provided better results than R^2 in all experimental cases [23]. In addition, agents are considered as objects and their performance are expressed by their masses, which will be calculated using a fitness function. Each object's position corresponds to a solution of the problem. By the gravitational force, all these objects attract each other and a heavy mass has a large effective intensity of attraction. Therefore, bigger or better mass has higher attractions and move more slowly. It can be considered as an adaptive learning rate [23].
In a system with N agents (masses), the positions are defined, as follow:

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^N) \text{ For } i = 1, 2, \ldots, N \tag{2}$$

At a specific time or iteration (t), the applied force to ith mass from jth mass is defined, as follow:

$$F_{ij}^{d}(t) = G(t) \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \tag{3}$$

Where, M_i and M_j are the masses related to ith and jth agents, respectively. Also, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between ith and jth agents. In addition, G(t) is gravitational constant at time or iteration (t) which is a function of two controlling parameters. The proposed form for G(t) is shown in Equation (4):

$$G(t) = G_0 e^{\frac{-\alpha t}{T}} \tag{4}$$

Where, t is the algorithm iteration number and T is the total iteration of IGSA. The G_0 and α are known as IGSA controlling parameter constants. Precise choose of these two parameters can lead to the best results, so that, increasing amount of G_0 can cause increase of agents' acceleration. In contrast, increase of α will lead to acceleration reduction. Two others controlling parameters of IGSA are C_1 and C_2 that their proper values depend on problem.

Total force that acts on ith agent in a dth dimension is calculated, as follow:

$$F_i^d(t) = \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t) \tag{5}$$

In order to give a stochastic characteristic to the algorithm, the total force in dth dimension has been considered as random weighted sum of dth components of the forces, where $rand_j$ is a random number in the interval [0, 1].

In the IGSA, each mass has a velocity and an acceleration which will be expressed as $v_i(t)$ and $a_i(t)$, respectively. According to Equation (6), velocity variation or acceleration of each mass is equal to the force acted on the system divided by mass of inertia. Equation (7) shows current velocity of each mass which is equal to the sum of the fraction of its previous velocity and the velocity variation.

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{6}$$

$$v_i^d(t+1) = rand \times v_i^d(t) + a_i^d(t) + rand \times C_1 \times (x(t) - x_{pbest}) + rand \times C_2 \times (x(t) - x_{gbest}) \tag{7}$$

Where, rand is a uniform random number between 0 and 1, x_{pbest} is the best personal position for each agent and x_{gbest} is the best global position among all agents, so far. This random term is the second random number in the IGSA calculation and the presence of these terms will guarantee the randomized characteristic of the IGSA. Also, $v_i(t)$ and $a_i(t)$ are velocity and acceleration in tth iteration and $v_i(t+1)$ represents the velocity in the $(t+1)^{th}$ iteration. When, acceleration and velocity of each mass are calculated, the new position of the masses can be obtained, as follow:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{8}$$

2.2 Implementation of IGSA for training ANN

Structures of ANNs are based on human brain and its neural cells. Prominent trait of ANN is its ability to learn complicated problems between input and output vectors, in general, these networks are capable to model any non-linear functions. This ability causes neural networks are used in practical problems such as; comparative diagnosis and controlling non-linear systems. Structure of simple ANN has been shown in Figure 1, it can be seen that each neuron receive its inputs and then, applies a non-linear function to neurons which is called activation function.

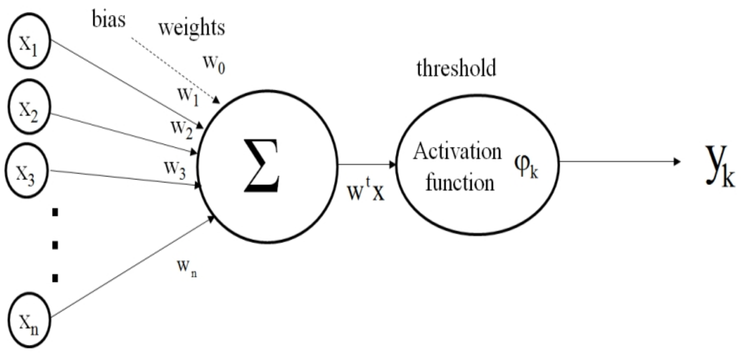

Figure 1. Simplified Structure of an ANN

The number of neurons in input and output layer depends on the number of input and output of problem, whereas; choose of hidden layers' number of neuron is a design problem. ANNs have mechanism for learning, this mechanism changes weights related to connections of ANN until desirable output is obtained. As it can be seen in Figure 1, activation function is effective part of ANN. Activation functions for the hidden units are needed to introduce nonlinearity into the network. Without nonlinearity, hidden units would not make networks, more powerful. Also, Different forms of activation functions are available that each of them has especial characteristic and application. For example, for the output units, an activation function which is proper for distribution of the target values should be chosen, as follow:

- For binary (0/1) targets, the logistic function such as "logsig" is an excellent choice.
- For categorical targets using 1-of-C coding, the "softmax" activation function is the logical extension of the logistic function.
- For continuous-value targets with unknown bounds, use the identity or "linear" activation function such as "purelin".
- For continuous-value targets with a bounded range, the logistic and tanh functions can be used, provided you either scale the outputs to the range of the targets or scale the targets to the range of the output activation function ("scaling" means multiplying by and adding appropriate constants).

- If the target values are positive but have unknown upper bound, an exponential output activation function can be used, but by consideration of overflow.

In this approach in order to train ANN, a new kind of heuristic algorithm has been used which is based on swarm intelligence. Here, Mean Squared Error (MSE) performance function which is a criterion of difference between actual output of ANN and target output has been used as a function which should be minimized. Hence, value of MSE is reduced until zero using IGSA. When MSE reaches to zero, it means that actual output is the same as target output and it can be understood ANN has been well trained. The implementation procedure of proposed approach has five steps, as follow:

Step 1) Determining initial parameters:

In this step, the number of layers and neurons in each layer and design structure of ANN are determined. Then, initial parameters of IGSA are determined such as; number of masses/agents, number of problem/solution dimensions that is dependent on number of synaptic weights of MLFFNN. The number of weights (or the number of solution dimension) can be calculated by Equation (9).

$$[\mathrm{NID} \times \theta + b_1] + [\theta \times \beta + b_2] + [\beta \times \gamma + b_3] = w \ (\text{Number of Neural Network weights}) \tag{9}$$

Where, θ, β, γ and NID are the number of neurons in first layer, second layer, third one and the number of input data, respectively. And, b is bias for each neuron.

Then, value of controlling parameters G_0 and α and the number of iteration are determined, eventually in this part, primary location of masses in solution space is determined randomly, that each mass has w dimensions.

Step 2) application of optimizing IGSA

In this step, IGSA has been run to train MLFFNN. In order to use IGSA, value of MSE should be loaded; therefore, command "feval" has been used. In this part which is related to MLFFNN, this structure is created.

At first input data, target matrix and matrix of weights that expresses initial position of masses are loaded. Dimension of weight matrix is 1×w as follow:

$$W_1 = \begin{bmatrix} P_1 & P_2 & \ldots & P_w \end{bmatrix} \tag{10}$$

Where, W_1 is the position of the first mass in solution space which expresses weights of MLFFNN of the first mass. After dividing this matrix into sub-matrixes which express weights of one layer and its biases, structure of MLFFNN has been made.

At the end of this structure, MSE value of subtract of actual output and target output is calculated, this value is fitness value of MLFFNN for IGSA and proposed algorithm will change weights till this value becomes minimum in next iterations. MSE function can be seen in Equation (11).

$$\mathrm{MSE} = \frac{\sum_{M,N} (\text{target output - acual output})^2}{M \times N} \tag{11}$$

Step 3) updating IGSA parameters

In this step, parameters of IGSA are updated. At first, acceleration is updated then, velocity and position of masses will be updated. These new positions for one mass express new weights for MLFFNN.

Step 4) inspection of stop criterion

In order to stop this algorithm, two criterions are considered. One of them is the number of iteration and the other one is MSE value. Whenever MSE value becomes zero, it means that, actual output and target output are same.

Step 5) End

Flowchart of proposed approach for training ANN by IGSA has been illustrated in Figure 2.

It is impossible to apply analog signal to ANN, hence, discrete sample data have been extracted from inrush and internal fault current waveforms, and in order to have valid data as input of ANN, data in [25] has been used. But because of severe variation in data they cannot be used without any processing, therefore, this data has been normalized, as follow:

$$\text{Normalized Data} = \frac{\text{data - min(row)}}{\text{max(row) - min(row)}} \tag{12}$$

Normalized data of simulated transformers have been applied to MLFFNN as input data, also three classes have been considered as output for normal condition, magnetizing inrush current and internal fault current. In output, "1 0" represents normal condition, "0 1" inrush current and "1 0" internal fault current. In Figure 3, block diagram of simulation can be seen.

3. Simulations and Analysis

In this section, simulation results have been shown and analysis and discussion has been done:

3.1 Simulation Results

After extracting discrete sample data of three mentioned conditions. Data of 12 transformers out of 15 have been chosen as train data and other were test data. Also, for this work controlling parameters are; C_1 and C_2 are considers as 2, amount of α is 10 and magnitude of G_0 is 260, also, the number of masses is equal to "10", in other hand, MLFFNN has one input layer with 16 neurons, two hidden layers with 16 and 16 neurons and one output layer with two neurons based on three output classes.

At first ANN has been trained with train data then test data have been used to test trained ANN, in Table 1, result data or in other word actual output of MLFFNN for test data has been listed.

Therefore, by using IGSA for training ANN, MSE value becomes 4.2746e-012 and its values' curve has been shown in Figure 4, this figure shows that algorithm is stopped at iteration 1000, that is, algorithm arrive at one of the stop criterions, also after 50 times run, it has been observed that mean value of MSE is about 7.1822e-010. From results, it can be understood that proposed algorithm can deal with discriminating between normal condition, inrush current and internal fault current very well and can do this work with a slight error.

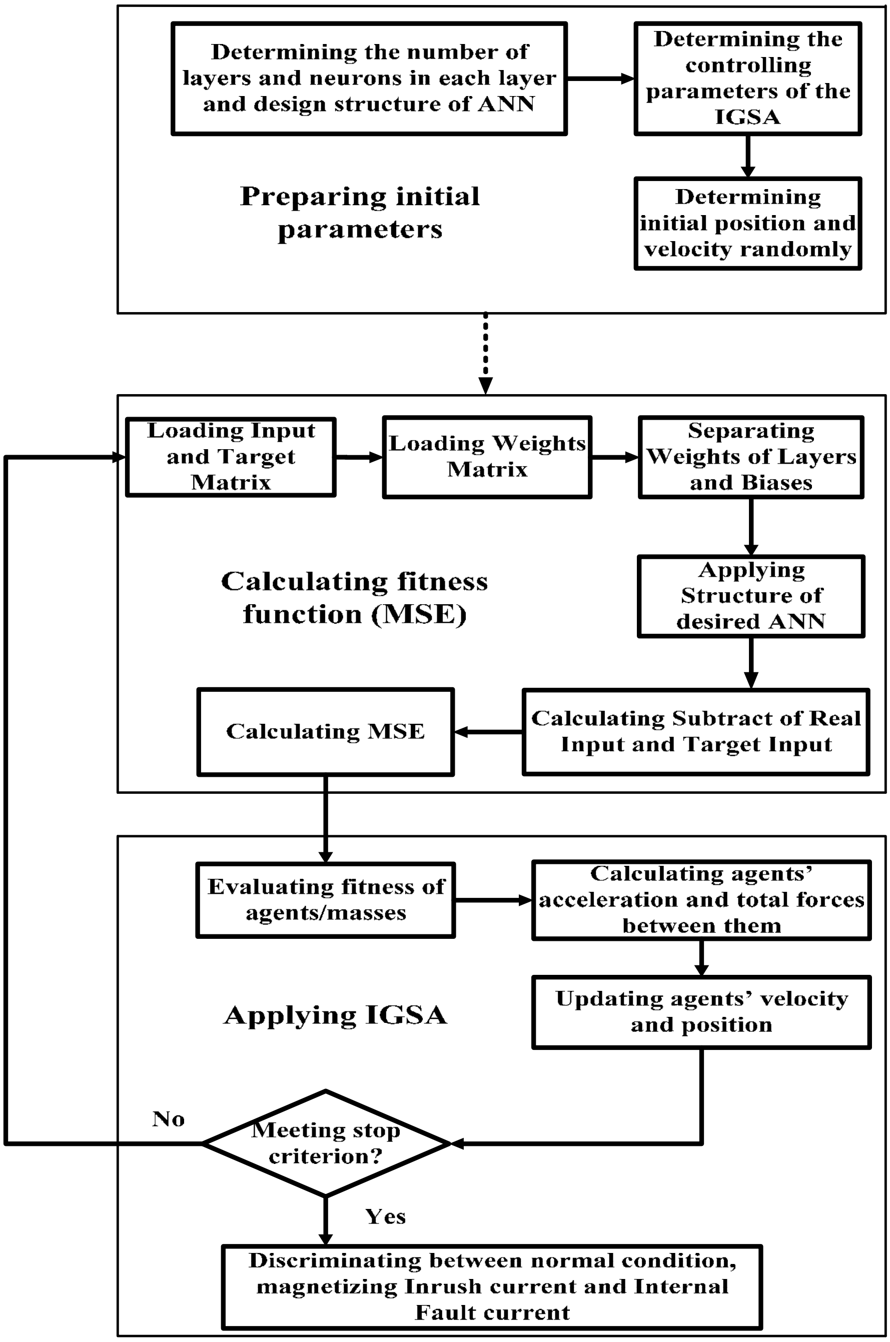

Figure 2. Flowchart of proposed approach for training ANN by IGSA

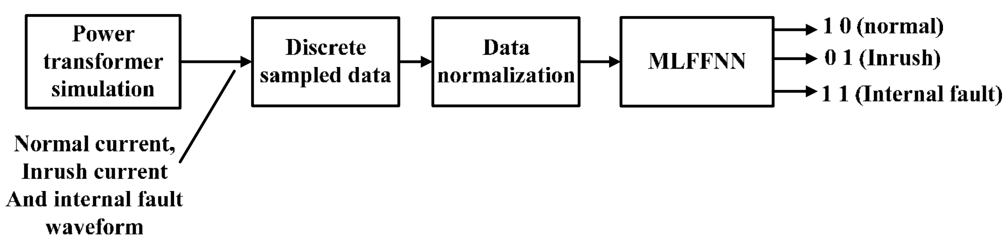

Figure 3. Block diagram of simulation of discrimination Inrush current from internal fault and normal condition

Table 1 Test Results for Train and Test Data

Train data	Condition	Target output	Actual output		Error	
16MVA, 110/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
25MVA, 110/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
5MVA, 110/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
3MVA, 110/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
2MVA, 110/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
16MVA, 110/11kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
25MVA, 110/11kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
5MVA, 110/11kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
3MVA, 110/11kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000

	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
2MVA, 110/11kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
16MVA, 66/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
25MVA, 66/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
Test data	**condition**	**Actual output**	**Target output**		**Error**	
5MVA, 66/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	09996	0.0000	0.0004
3MVA, 66/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000
2MVA, 66/33kV	Normal current	1.0000 0.0000	1.0000	0.0000	0.0000	0.0000
	Inrush current	0.0000 1.0000	0.0000	1.0000	0.0000	0.0000
	Internal fault current	1.0000 1.0000	1.0000	1.0000	0.0000	0.0000

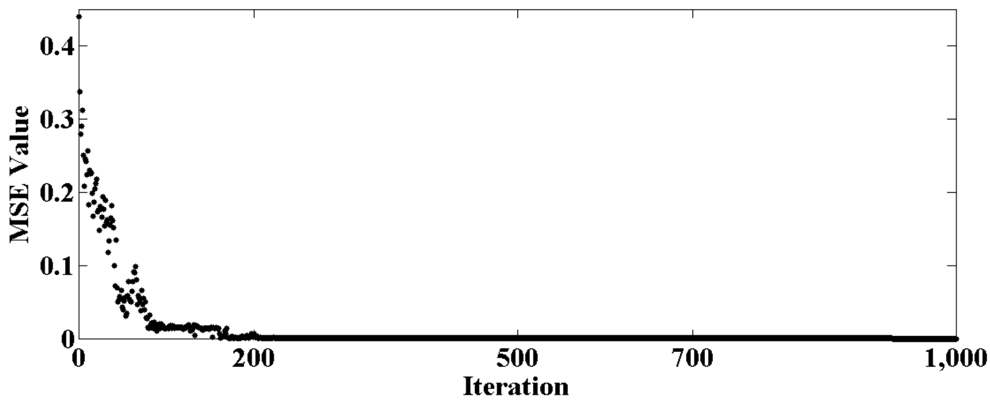

Figure 4. Variation of MSE versus Iteration

3.2 Sensitivity Analysis on IGSA Controlling and ANN Structural Parameters

The IGSA has some parameters that their variations can affect results such as controlling parameters (α, G_0 and the number of masses), also, the number of neurons in each layer and activation function in output layer change desired results. In addition, the values of C_1 and C_2 can impact on the final result, assessments show that the best values for these two parameters are when they are considered equal as 2, in this case. So, in this part, sensitivity of results to these parameters has been analyzed. This work has been done for 5MVA, 66/33kV transformer, that is, for each condition; trained MLFFNN in different conditions has been used for testing data of this transformer. Therefore, sensitivity to number of mass, α and G_0 have been shown in Tables 2, also time of testing mentioned transformer has been calculated. After obtained following results, it becomes obvious that test time for one transformer by proposed approach is about 10 ms (about half of cycle) and it seems good time for discriminating Inrush current from internal fault. To do this, structure of MLFFNN is 16-16-16-2, for sensitivity to number of mass, G_0= 260 and α = 10 have been considered. For sensitivity to α, G_0 = 260 and number of masses = 10 have been considered. And for sensitivity to G_0, number of masses = 10 and α = 10 have been considered, also reported MSE value is average value in 50 times run.

Table 2. Sensitivity analysis on controlling parameters of IGSA, when C_1 and C_2 are 2

Sensitivity analysis for	Controlling Parameters	condition	Target output	Actual output	Error	MSE of trained MLFFNN
Sensitivity analysis on the number of masses	m = 5	Normal current	1.0000	1.0000	0.0000	0.0001
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	0.9994	0.0006	
		Internal fault current	1.0000	0.9999	0.0001	
			1.0000	0.9918	0.0082	
	m =10	Normal current	1.0000	1.0000	0.0000	4.2746e-012
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	1.0000	0.0000	
			1.0000	1.0000	0.0000	
	m = 20	Normal current	1.0000	1.0000	0.0000	0.0003
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	0.9650	0.0718	
			1.0000	0.9219	0.0781	
Sensitivity analysis on the α	α = 10	Normal current	1.0000	1.0000	0.0000	4.2746e-012
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	1.0000	0.0000	
			1.0000	1.0000	0.0000	

	α =20	Normal current	1.0000	1.0000	0.0000	0.0015
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0533	0.0533	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	0.9113	0.0887	
			1.0000	0.8285	0.1715	
	α = 25	Normal current	1.0000	0.9998	0.0002	0.0096
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0021	0.0021	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	0.9456	0.0544	
			1.0000	0.7593	0.2407	
Sensitivity analysis on the G_0	G_0 = 100	Normal current	1.0000	1.0000	0.0000	2.2304e-005
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	0.9985	0.0015	
		Internal fault current	1.0000	1.0000	0.0000	
			1.0000	0.9995	0.0005	
	G_0 = 150	Normal current	1.0000	1.0000	0.0000	5.0800e-007
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0021	0.0021	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	1.0000	0.0000	
			1.0000	0.9954	0.0056	
	G_0 = 200	Normal current	1.0000	1.0000	0.0000	1.4558e-008
			0.0000	0.0000	0.0000	
		Inrush current	0.0000	0.0000	0.0000	
			1.0000	1.0000	0.0000	
		Internal fault current	1.0000	0.9993	0.0007	
			1.0000	0.9995	0.0005	

As it can be seen in Table 2, finding and using optimum value of controlling parameters can help engineers to solve problem in best way, by different testing for different conditions it has been understood that optimum range for number of masses is bigger than 9 and less than 14, that is, in this range by increasing the number of masses accuracy of solution will be better, but time of training is increased, whereas, results are approximately same, therefore in order to save time we can choose optimum number of mass with best performance. Also for sensitivity to α, as it is observed, by increasing magnitude of α accuracy of approach and MSE value are reduced and it can be seen that best range is 7 to 15, considering MSE value for α = 20 and 25, this algorithm sticks in minimum local and cannot reach to global answer. In analysis of variation of G_0, it can be inferred that optimum value of this parameter for this problem is in interval [150-270] and by increasing this parameter acceleration would be increased and better results can be achieved.

In addition to above analysis, variations of result by changing the number of neurons in each layer and activation function in output layer has been evaluated. To do this, different structures for MLFFNN have been considered which can be seen in Table 3, and also, results for using three forms of activation function in output layer that can be used in this kind of study are illustrated in Table 4.

Table 3. Sensitivity Analysis for different structure of MLFFNN

Structure of MLFFNN (the number of synaptic weights)	condition	Target output	Actual output	Error	MSE of trained MLFFNN
16-16-16-2 (578)	Normal current	1.0000 0.0000	1.0000 0.0000	0.0000 0.0000	4.2746e-012
	Inrush current	0.0000 1.0000	0.0000 1.0000	0.0000 0.0000	
	Internal fault current	1.0000 1.0000	1.0000 1.0000	0.0000 0.0000	
16-8-8-2 (226)	Normal current	1.0000 0.0000	1.0000 0.0048	0.0000 0.0048	0.0257
	Inrush current	0.0000 1.0000	0.1528 0.9998	0. 1528 0.0002	
	Internal fault current	1.0000 1.0000	1.0000 0.9841	0.0000 0.0159	
16-24-24-2 (1058)	Normal current	1.0000 0.0000	0.9996 0.0000	0.0004 0.0000	0.0016
	Inrush current	0.0000 1.0000	0.0010 0.9996	0. 0010 0.0004	
	Internal fault current	1.0000 1.0000	0.9999 0.9220	0.0001 0.0780	
16-16-8-2 (426)	Normal current	1.0000 0.0000	1.0000 0.0009	0.0000 0.0009	0.0012
	Inrush current	0.0000 1.0000	0.0076 0.9991	0.0076 0.0009	
	Internal fault current	1.0000 1.0000	1.0000 0.9999	0.0000 0.0001	
16-24-16-2 (842)	Normal current	1.0000 0.0000	0.9982 0.0000	0.0018 0.0000	0.0033
	Inrush current	0.0000 1.0000	0.0063 0.9998	0.0063 0.0002	
	Internal fault current	1.0000 1.0000	0.9652 0.9573	0.0348 0.0427	
16-8-24-2 (402)	Normal current	1.0000 0.0000	1.0000 0.0003	0.0000 0.0003	0.0013
	Inrush current	0.0000 1.0000	0.0002 0.9999	0.0002 0.0001	
	Internal fault current	1.0000 1.0000	0.9989 1.0000	0.0011 0.0000	

Table 4. Sensitivity Analysis for Different Activation Function in Output Layer

activation function	condition	Target output	Actual output	Error	MSE of trained MLFFNN
Purelin (Linear)	Normal current	1.0000	1.0451	0.0451	0.1094
		0.0000	-0.1103	0. 1103	
	Inrush current	0.0000	0.3669	0. 3669	
		1.0000	0.9945	0.0055	
	Internal fault current	1.0000	0.8545	0.1455	
		1.0000	0.6447	0.3553	
Softmax	Normal current	1.0000	0.9999	0.0001	0.0834
		0.0000	0.0001	0.0001	
	Inrush current	0.0000	0.0000	0.0000	
		1.0000	1.0000	0.0000	
	Internal fault current	1.0000	0.5389	0.4611	
		1.0000	0.4611	0.5389	
logsig	Normal current	1.0000	1.0000	0.0000	4.2746e-012
		0.0000	0.0000	0.0000	
	Inrush current	0.0000	0.0000	0.0000	
		1.0000	1.0000	0.0000	
	Internal fault current	1.0000	1.0000	0.0000	
		1.0000	1.0000	0.0000	

Results show that for training MLFFNN by IGSA, best structure would be obtained with two input layer and best form of ANN is 16-16-16-2. Using this structure, the number of synoptic weights which defines solution's dimension is 578. Also, best activation function to discriminate inrush current from normal and internal fault current is "logsig".

4. Comparison

- Using IGSA is the first contribution of this paper, advantages of this optimization algorithm in comparison with other same methods is presented as follow:

The IGSA has slight similarities with some optimizing algorithm such as Central Force Optimization (CFO) and Particle Swarm Optimization (PSO). Some differences between IGSA and CFO/PSO are mentioned in following [23-28]:

1) Differences between IGSA and CFO

CFO is a deterministic multi-dimensional search algorithm. It models the probes that fly through the search space under the influence of gravity. In both CFO and IGSA the probes positions and accelerations are inspired by particle motion in a gravitational field but they use different formulations.

– One of major differences is that CFO is inherently deterministic and does not use any random parameter in its formulation while IGSA is a stochastic search algorithm.

– The acceleration and movement expressions and calculation of the masses in IGSA are different from CFO.

– In CFO, the initial probe distribution is systematic (based on a deterministic rule) and has a significant effect on the algorithm’s convergence but in IGSA the initial distribution is random.

– Another difference is that in CFO, G is a constant while in IGSA, G is a control parameter.

2) Differences between IGSA and PSO

In both IGSA and PSO the optimization is obtained by agents' movement in the search space, however the movement strategy is different. Some important differences are as follows:

– In PSO the direction of an agent is calculated using only two best positions, pbest and gbest. But in IGSA, the agent direction is calculated based on the overall force obtained by all other agents.

– In PSO, updating is performed without considering the quality of the solutions, and the fitness values are not important in the updating procedure while in IGSA the force is proportional to the fitness value and so the agents see the search space around themselves in the influence of force.

– In PSO, updating is performed without considering the distance between solutions while in IGSA the force is reversely proportional to the distance between solutions.

– Finally, note that the search ideas of these algorithms are different. PSO simulates the social behavior of birds and IGSA inspires by a physical phenomena i.e. gravitational force.

- The second Contribution of proposed method in this paper is using ANN trained and optimized by IGSA, obtained results show that this kind of ANN can present promising performance for classifying different kind of possible occurrence in transformer.
- The third contribution is sensitivity analysis that leads to best and optimum parameters for designing the Intelligent and optimum protection system for power transformer.

5. Conclusion

This paper presents a novel approach based on trained ANN classifier to deal with the problem of discriminating between transformer internal faults and magnetizing inrush current. This approach can be presented in four steps; normalizing current sampled data, preparing ANN, running IGSA and finally, discrimination step. The introduced approach neither depend on the transformer equivalent circuit model nor the harmonic content of differential current, rather make the decision based on the transformer's current directly which is more accurate than traditional harmonic restraint based technique. Since, high second harmonic component may be created during internal faults and low second harmonic component during magnetizing inrush, the conventional harmonic restraint technique may face to mal-function. So, need to a precise and fast approach seems necessary. Results of proposed method show that this classifier can discriminate inrush current from internal fault very quickly and accurately.

References

[1] A.A.H. Eldin, and M.A. Refaey, "A novel algorithm for discrimination between inrush current and internal faults in power transformer differential protection based on discrete wavelet transform," Electric Power System Research, vol. 81, 2011.
[2] M.Steurer, and K. Frohlich, "The Impact of Inrush Currents on the Mechanical Stress of High Voltage Power Transformer Coils," IEEE Transactions on Power Delivery, vol. 17, 2002.
[3] M. Nagpal, T.G. Martinich, A. Moshref, K. Morison, and P. Kundur, "Assessing and limiting impact of transformer inrush current on power quality," IEEE Transactions on Power Delivery, vol. 21, 2006.
[4] S.A. Saleh, and M.A. Rahman, "Off-line Testing of a Wavelet Packet-Based Algorithm for Discriminating Inrush Current in Three-phase Power Transformers," IEEE Conference on Power Engineering and Large Engineering Systems, vol. 1, 2003.
[5] M.E.H. Golshan, M. Saghaian-nejad, A. Saha, and H. Samet, "A new method for recognizing internal faults from inrush current conditions in digital differential protection of power transformers," Electric Power System Research, vol. 71, 2004.
[6] P.L. Mao, and R.K. Aggarwal, "A Novel Approach to the Classification of the Transient Phenomena in Power Transformers Using Combined Wavelet Transform and Neural Network," IEEE Transactions on Power Delivery, vol. 16, 2001.
[7] A. Chatterjee, M. Maitra, and S.K. Goswami, "Classification of Over-Current and Inrush Current for Power System Reliability using Slantlet Transform and Artificial Neural Network," Expert Systems with Applications, vol. 36, 2009.
[8] H. Khorashadi-Zadeh, "Fuzzy-Neuro Approach to Investigating Transformer Inrush Current Differential Protection for Power Transformer," IEEE PES Transmission and Distribution Conference and Exhibition, vol. 1, 2006.
[9] S.A. Saleh, and M.A. Rahman, "A New Transient Model for Three-Phase Power Transformers Using a Wavelet Filter Bank," IEEE Transactions on Power delivery, vol. 20, 2005.
[10] S.A. Saleh, and M.A. Rahman, "Modeling and Protection of a Three-Phase Power Transformer Using Wavelet Packet Transform," IEEE Transactions on Power delivery, vol. 20, 2005.
[11] M. Kitayama, "Wavelet-Based Fast Discrimination of Transformer Magnetizing Inrush Current," Electrical Engineering in Japan, vol. 158, 2007.
[12] O.A.S. Youssef, "Combined Fuzzy-Logic-Wavelet-Based Fault Classification Technique for Power System Relaying," IEEE Transactions on Power delivery, vol. 19, 2004.
[13] G. Baoming, A.T. Almeida, Z. Qionglin, and W. Xiangheng, "An equivalent instantaneous inductance-based technique for discrimination between inrush current and internal faults in power transformers," IEEE Transactions on Power Delivery, vol. 20, 2005.
[14] Y.C. Kang, B.E. Lee, and S.H. Kang, "Transformer protection relay base on induced voltages," International Journal of Electrical Power and Energy Systems, vol. 29, 2007.
[15] X.N. Lin, P. Liu, and O.P. Malik, "Studies for Identification of the Inrush Based on Improved Correlation Algorithm," IEEE Transactions on Power delivery, vol. 17, 2002.
[16] S. Guocai, and Y. Dachuan, "Identifying internal faults of transformers through the similarity degree between voltage and current," IEEE Power Engineering Society Winter Meeting, vol. 1, 2000.
[17] T. Babnik, and F. Gubina, "Fast power transformer fault classification methods based on protection signals," IEE Proceeding Generation, Transmission, and Distribution, vol. 150, 2003.
[18] S.R. Samantaray, B.K. Panigrahi, P.K. Dash, and G. Panda, "Power transformer protection using S-transform with complex window and pattern recognition approach," IET Generation, Transmission, and Distribution, vol. 1, 2007.
[19] G. Diaz, P. Arboleya, and J. Gomez-Aleixandre, "A new transformer differential protection approach on the basis of space-vectors examination," Electrical Engineering, vol. 87, 2005.
[20] V. Malathi, and N.S. Marimuthu, "Support Vector Machine for Discrimination between Fault and Magnetizing Inrush Current in Power Transformer," Journal of Computer Science, vol. 3, 2007.
[21] X. Ma, and J. Shi, "A new method for discrimination between fault and magnetizing inrush current using HMM," Electric Power System Research, vol. 56, 2000.
[22] Z. Lu, W.H. Tang, T.Y. Ji, and Q.H. Wu, "A Morphological Scheme for Inrush Identification in Transformer Protection," IEEE Transactions on Power Delivery, vol. 24, 2009.

[23] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," Information Sciences, vol. 179, 2009.
[24] R.A. Formato, "Central force optimization: a new meta-heuristic with applications in applied electromagnetics," Progress in Electromagnetics Research, vol. 77, 2007.
[25] M. Geethanjali, S.M.R. Slochanal, and R. Bhavani, "PSO trained ANN-based differential protection scheme for power transformers," Neurocomputing, vol. 71, 2008.
[26] R.A. Formato, "Central force optimization: a new nature inspired computational framework for multidimensional search and optimization," Studies in Computational Intelligence, vol. 129, 2008.
[27] J. Kennedy, and R.C. Eberhart, "Particle swarm optimization," IEEE International Conference on Neural Networks, vol. 4, 1995.
[28] F.V.D. Bergh, and A.P. Engelbrecht, "A study of particle swarm optimization particle trajectories," Information Sciences, vol. 176, 2006.