Brazilian Journal
of Chemical
Engineering

ISSN 0104-6632
Printed in Brazil
www.abeq.org.br/bjche

Vol. 28, No. 01, pp. 157 - 168, January - March, 2011

DESIGNING A NEURAL NETWORK FOR CLOSED THERMOSYPHON WITH NANOFLUID USING A GENETIC ALGORITHM

H. Salehi, S. Zeinali Heris*, M. Koolivand Salooki and S. H. Noei

Chemical Engineering Department, Faculty of Engineering-Ferdowsi University of Mashhad,
Phone: + (98) (511) 8816840, Fax: + (98) (511) 8816840, Mashhad, Iran.
E-mail: zeinali@ferdowsi.um.ac.ir

(Submitted: January 15, 2010 ; Revised: July 13, 2010 ; Accepted: December 1, 2010)

Abstract - Heat transfer of a silver/water nanofluid in a two-phase closed thermosyphon that is thermally enhanced by magnetic field has been predicted by an optimized artificial Neural Network. Artificial neural network is a technique with flexible mathematical structure that is capable of identifying complex non-linear relationships between input and output data. A multi-layer perception neural network was used to estimate the thermal efficiency and resistance of a thermosyphon during application of a magnetic field and using nanoparticles in the water as the working fluid. The magnetic field strength, volume fraction of nanofluid in water and inlet power were used as input parameters and the thermal efficiency and thermal resistance were used as output parameters. The results were compared with experimental data and it was found that the thermal efficiency and resistance estimated by the multi-layer perception neural network are accurate. The GA-ANN (Genetic Algorithm-Artificial Neural network) predicts the thermosyphon behavior correctly within the given range of the training data. In this study, a new approach for the auto-design of neural networks, based on a genetic algorithm, has been used to predict collection output of a closed thermosyphon.
Keywords: Thermosyphon; Nanofluid; Magnetic field; Genetic algorithm; Neural network.

INTRODUCTION

Development of heat pipe thermosyphons to sink the heat dissipated from industrial processes and miniaturized equipment is being intensively researched. Although considerable progress has been made in macro devices, the control of heat transfer in micro devices is still a challenge. Thus, with the idea of enhancing the heat transfer capacity, as well as controlling the heat transferred, a magnetic field with nanofluid has been proposed as the working fluid (Nakatsuka et al., 1990; Jeyadevan et al., 1990).

Researchers have observed that the thermal conductivity of nanofluid is much higher than that of the base fluids, even for low solid volume fraction of nanoparticles in the mixture (Eastman et al., 2001; Xuan and Li, 2000). The nanofluid is stable, introduces very little pressure drop and can pass through micro channels (Wang and Mujumdar, 2008 a, b). Das et al. (2003) observed that the thermal conductivity of nanofluid increases with increasing temperature. Due to the lack of an accepted theory to predict the effective thermal conductivity of a nanofluid, several researchers have used different correlations to predict the apparent thermal conductivity of this two-phase mixture; among these correlations, the models proposed by Hamilton and Crosser (1962), Wasp (1977), Yu and Choi (2003) can be cited.

The thermosyphon, as applied to the control of the heat transport rate and/or the operating temperature, utilizes one of the following four means of control: (1) varying the thermal resistance between the heat source and the evaporator section or between the condenser section and the heat sink; (2) varying the effective

*To whom correspondence should be addressed

condenser area in combination with a non-condensable gas or with an excess inventory of working fluid; (3) interrupting or throttling the vapor flow from the evaporator forward to the condenser section (vapor flow control); and (4) interrupting or impeding the liquid flow through the wick from the condenser forward to the evaporator section (liquid flow control). Except for (1), all the above methods of control govern the performance characteristics of the thermosyphon itself. This nanofluid possesses high thermal conductivity and, in a magnetic field, its flow is impeded by the magneto hydrodynamic effect. This introduces the possibility of controlling the liquid flow of a thermosyphon nanofluid by regulating the applied magnetic field.

Kang et al.(2009) experimentally investigated the sintered heat pipe thermal performance using silver-nanofluid as the working fluid. Anoop et al. (2009) observed that nanofluids showed higher heat transfer characteristics than the base fluid and that a nanofluid with 45 nm particles had a higher heat transfer coefficient than one with 150 nm particles. Noie et al. (2009) investigated heat transfer enhancement using Al_2O_3/water nanofluids in a two-phase closed thermosyphon.

The magnetizing force has been known for many years (Faraday, 1847; Pauling et al., 1946), but, until recently, it had been neglected almost completely. Braithwaite et al. (1991) again highlighted this force for Bénard convection of a paramagnetic water solution to give larger or smaller heat transfer rates than the average heat transfer rate of a nonmagnetic field, although they did not compare their findings with the data of Silveston (1958). Maki et al. (2002) applied magnetizing force for natural convection of air in a shallow cylindrical enclosure heated from below and cooled from above. Schmidt (1966) studied the effects of a magnetic field on the conduction heat transfer at the stagnation point of a partially ionized argon gas.

The present paper covers a **GA-ANN** modeling of experimental study of the heat transport characteristics of a nanofluid thermosyphon in a magnetic field. Experimental measurements have been made of the heat transport rate at different volume fractions of nanoparticles in various magnetic fields as compared to water in the thermosyphon.

EXPERIMENTAL SECTION

Experimental Setup

In this study, experiments were performed with five different volume fractions of Ag nanoparticles in water (0, 20, 40, 60 and 80 ppm). By changing inlet power, the thermal efficiency of the thermosyphon was calculated. The effect of the magnetic field coupled to the thermosyphon on the performance of the experimental system was investigated experimentally. The thermal efficiency is a suitable parameter to evaluate the effective thermal conductivity of the nanofluid with permanent magnetics. The vertical type thermosyphon was linked with a vacuum pump and valve unit. A mechanical vacuum pump capable of up to -0.9 bar relative pressure and pumping capacity of 142 L/min was used for partial elimination of the non-condensable gases from the thermosyphon. An electrical resistance of nominal power 250 W, which was wrapped around the evaporator section, heated the evaporator section. To prevent heat loss, the electrical elements were insulated with rock wool having a thickness of 20 mm and 10 mm in the evaporator, adiabatic and condensation sections, respectively. The power supplied to the evaporator section was determined by monitoring the applied voltage and current. Local temperatures in the TPCT were measured by two thermocouples. The mini-thermocouples were mechanically attached to the surface of the pipe. One thermocouple was mounted on the evaporator section and one on the condenser section. Running water inlet and outlet temperatures were measured using a digital mini-thermometer with an accuracy ±0.1ºC. The nanofluid was charged into the tube under -0.9 bar relative vacuum pressure. The mini-thermocouples were mechanically attached to the surface of the pipe.

Permanent magnets (0T, 0.12T, 0.35T, and 1.2 T) are placed on the evaporate section.

All the electrical (such as thermocouples, thermometers, ammeter and voltmeter) and mechanical equipment (such as rotameter) was calibrated initially.

Figure 1 shows a schematic of experimental setup. Table 1 shows the different parts of the Thermosyphon.

To ensure that there were no remaining remnants, prior to all experiments it was necessary to clean the inside of the main tube, especially whenever there was a change over from one nanofluid to another. This was done in three steps: cleaning with trichloroethane and methanol followed by vacuum drying. Before proceeding to test the nanofluid system, baseline experiments were conducted with pure water. Once the quality and repeatability of the baseline data were established, three water-based nanofluids were used as the working fluid. The experiments showed that about 30-40 min was needed for the system to reach the steady-state condition.

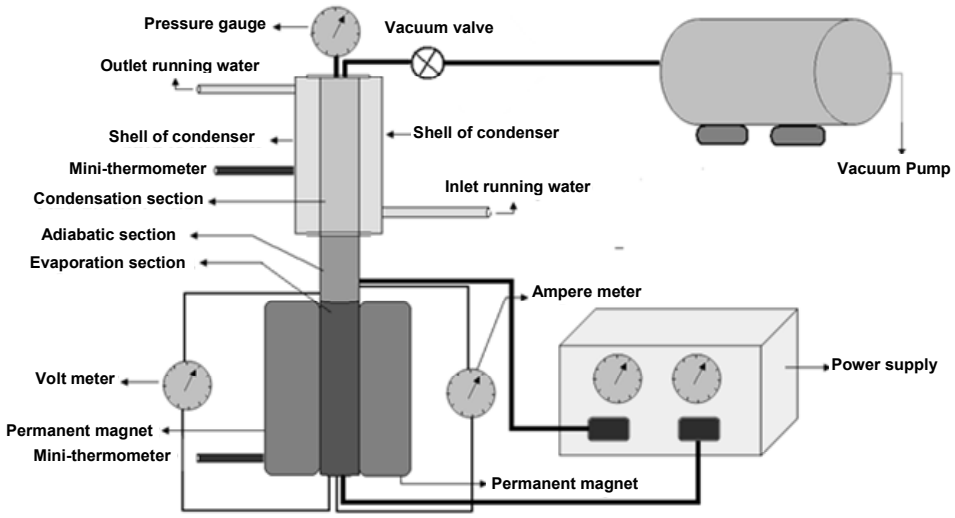

Figure 1: Experimental setup

Table 1: Characteristics of the thermosyphon

Thermosyphon parts	Characteristics
long copper tube	400 mm l, 20 mm ID, 22 mm OD
evaporator sections	0.25m Total length, can be varied by varying the length of the electrical resistance
adiabatic sections	0.25m total length
condenser section	0.15 m long, 45 mm OD

Nanofluid and Magnetic Field Interaction

The nanoparticles used in these experiments were 10nm silver nanoparticles. The base working fluid was pure water. Ag nanoparticles were prepared first using a catalytic chemical vapor deposition method, then added to pure water. No surfactant was used in the Ag/water nanofluid suspensions. To prevent any possible change of thermal properties of the nanofluid due to the presence of additive, the mixture was created using an ultrasonic homogenizer (Parsonic 3600S). Nanofluid concentrations of 20, 40, 60, and 80 mg/l (ppm) were used in this study.

After measuring the volume equivalent to the required mass of nanoparticle powder, they were mixed with distilled water in a flask and subjected to ultrasonication for about 4 h. No precipitation/ settlement of nanoparticles was observed after 24 h at rest. The stability of the particles was examined by putting the nanofluid in front of sunlight. It was seen that the mixtures were completely uniform and homogenous. Also, the density of the nanofluids did not show appreciable change after 24h.

Ag nanoparticles are paramagnetic, although it seems that, in the case of suspending them in water, a net negative electrical charge is created around nanoparticles. Then, because the nanoparticles have a surface electric charge when suspended in water and have motion in the thermosyphon, a Lorentz force is exerted on them. If the direction of the velocity vector is upward and the magnetic field is perpendicular to the tube, according to Right Hand Rule, the force exerted on the particles causes circulation of particles in the fluid (in the absence of the magnetic field, there is no such effect).

As shown in Figure 1, the magnetic field is applied by permanent magnets, with two permanent magnets located in a parallel mode on the tube. The evaporator section and the magnets had a same length. Our goal was to detect the effects of a magnetic field on the boiling heat transfer of the silver nanofluid, so the magnets were located on the evaporator section of the thermosyphon so as to cover this section completely. Values of the applied magnetic fields were 0, 0.12, 0.35 and 1.2 Tesla. It is clear that, when the distance from the magnets increases, the magnetic field intensity will diminish. Because our goal did not include measuring the magnetic field intensity, we did not measure it. The length of the magnets was 20cm and the magnetic field only applied to the evaporator section perpendicular to the thermosyphon. In the case of nanoparticles moving upward through the thermosyphon nanofluid medium, the magnetic field effects caused them to rotate in the fluid and also leads to entropy accretion of the particles in the presence of the magnetic field. The rotation of the nanoparticles in water created small wakes inside

the fluid and, as a result, the heat transfer was enhanced. It is clear that the fluid flow regime in the thermosyphon is still laminar and application of these values of magnetic field cannot change this regime.

One of the important factors evaluated in this thermosyphon study is its efficiency. There are many parameters affecting thermosyphon efficiency and, among them, boiling heat transfer has a direct effect; thus, increasing boiling heat transfer increased thermosyphon efficiency.

Uncertainty in the experimental data may have resulted from measuring errors of parameters such as current, voltage, inlet and outlet temperatures of the cooling water, mass flow rate. The thermocouples used have a maximum precision of 0.1°C. Flow rates were measured directly from the time taken to fill a glass vessel of known volume, with 5.0% uncertainty in measurement. The maximum precisions of the ammeter and voltmeter were 0.1 V and 1 A, respectively. The maximum uncertainty of the thermal resistance calculated taking into account the above considerations is 4.5%.

Figure 2 shows the thermal resistance of the thermosyphon versus magnetic field for various concentrations of nanofluid at an input power of 12W. It was recognized that the resistance decreases as the strength of the magnetic field increases. Also, thermal resistance decreases as the concentration of the silver nanoparticles increases. For example, at 0.35T, the resistance of the thermosyphon decreased 3.22% using 20ppm of silver/water nanofluid compared to pure water , but at 1.2T and the same concentration of Ag/water nanofluid, the resistance decrease was 5.37%.

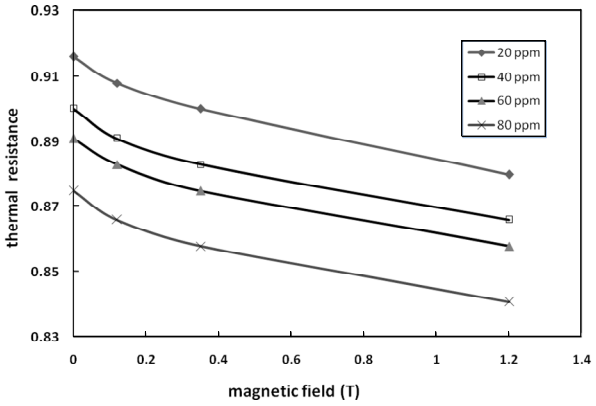

Figure 2: Effect of magnetic field on the thermal resistance of the thermosyphon at different Ag/water nanofluid concentrations.

The magnetic field applied on the thermosyphon nanofluid released the bubbles that accumulated on the inner wall of the evaporator, reduced the temperature fluctuation in the evaporation section, reduced the temperature difference between the fluid and wall and increased the vapor movement due to the Lorentz force and, consequently, enhanced the thermal efficiency of thermosyphon.

MODELING OF THE CLOSED THERMALLY-ENHANCED THERMOSYPHON

MLP Networks

The feed-forward neural networks are the most popular architectures due to their structural flexibility, good representational capabilities and the availability of a large number of training algorithms (Haykin, 1999). This network consists of neurons arranged in layers in which every neuron is connected to all neurons of the next layer (a fully connected network). MLP (Multi layer Perceptron) networks are a kind of feed-forward neural network with different transfer functions. The MLP network, sometimes called Back Propagation (BP) network, is probably the most popular ANN in engineering problems in the case of non-linear mapping and is called an ‘‘Universal Approximation”. It consists of an input layer, a hidden layer and an output layer (Figure 3). The input nodes receive the data values and pass them on to the first hidden layer nodes. Each one collects the input from all input nodes after multiplying each input value by a weight, attaches a bias to this sum, and passes on the results through a non-linear transformation like the sigmoidal transfer function. This forms the input either for the second hidden layer or the output layer that operates identically to the hidden layer. The resulting transformed output from each output node is the network output. The network needs to be trained using a training algorithm such as back propagation, cascade correlation and conjugate gradient.

Basically, the objective of training patterns is to reduce the global error. The goal of every training algorithm is to reduce this global error by adjusting the weights and biases. An output of a three-layer MLP network is defined by Eq. (1). (Haykin, 1999; Khandekar, 2008), Figure 3:

$$\cdot \psi_k^2 = f^2\left(\sum_{j=1}^{S^1} w_{jk}^2 f^1\left(\sum_{i=1}^{R} w_{ij}^1 p_i + b_j^1\right) + b_k^2\right), k = 1 \quad \text{to} \quad S^2 \cdot \quad (1)$$

where superscript 1 denotes the hidden layer and superscript 2 denotes the output layer. R, S^1 and S^2 illustrate the numbers of the input, hidden and output units, respectively. Also f , w_{ij} and b represent the transfer function, synaptic weight parameter and bias, respectively. The cross-validation data's errors are measured by the mean square-error (MSE), as defined in Eq. (2):

$$MSE = \frac{\sum_{i=1}^{n}(O_i - T_i)^2}{n} \quad (2)$$

where O_i is the desired output for the training data or cross-validation data i, T_i is the network output for the training data or cross-validation data i, and n is the number of data in the training data set or the cross validation data set; beside using MSE, we use NMSE (Normalized Mean Squared Error) and MAE (Mean Absolute Error) as the other parameters for prediction of error in our modeling. The basic element of a Multi-layer Perceptron neural network is the artificial neuron shown in Figure 4, which performs a simple mathematical operation on its inputs.

The input of the neuron consists of the variables $x_1, x_2, ...x_p$ and a threshold (or bias) term. Each of the input values is multiplied by a weight w_i, after which the results are added with the bias term. A known activation function φ performs a pre-specified (nonlinear) mathematical operation on the result. Various activation functions are summarized in Table 2.

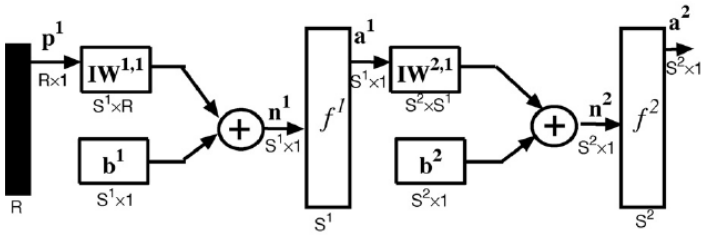

Figure 3: Multi layer perceptron

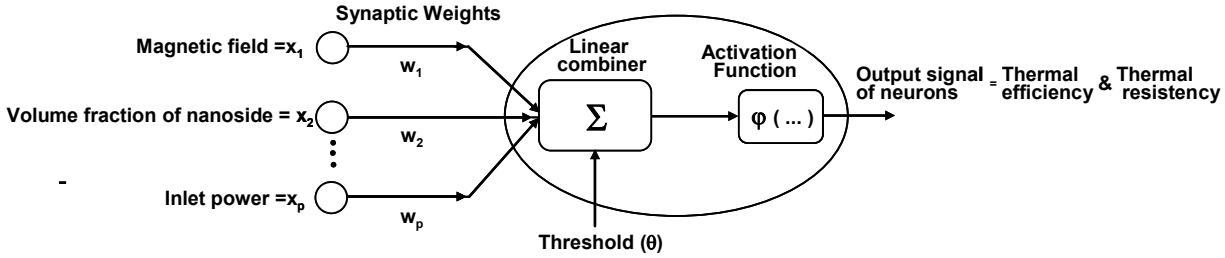

Figure 4: Schematic representation of the Multi-layer Perceptron.

Table 2: Basis functions used in MLP

Description	Transformation	Equation
Gaussian	$z^2 = \sum_{i=1}^{p} \frac{(x_i - t_i)^2}{\sigma_i^2}$	$\varphi(z) = \exp(-z^2)$
Threshold	$z = \sum_{j=1}^{p} \beta_j x_j + \beta_o$	$\varphi(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$
Sigmoid	$z = \sum_{j=1}^{p} \beta_j x_j + \beta_o$	$\varphi(z) = \frac{1}{1+\exp(-az)}$
Hyperbolic tangent	$z = \sum_{j=1}^{p} \beta_j x_j + \beta_o$	$\varphi(z) = \tanh\left(\frac{z}{2}\right) = \frac{1-\exp(-z)}{1+\exp(-az)}$

In order to approximate the function $\psi(x_1, x_2, ...x_p)$, where $(x_1, x_2, ...x_p)$ are p independent input variables, a three-layer perceptron network with p input neurons, S^1 hidden neurons coupled by a sigmoid transfer function and one output neuron coupled by a linear transfer function are selected. So, we can write Eq. (1) as follows:

$$\psi(x_1, x_2, ...) \approx \sum_{j=1}^{S^1} w_{jk}^2 f^1 \left(\sum_{i=1}^{p} w_{ij}^1 x_i + b_j^1 \right) + b_k^2 \qquad (3)$$

where $\psi(x_1, x_2, ...)$ is the approximated output function and b and w are the unknown coefficients.

Genetic Algorithm

Holland (1975) introduced an optimization procedure that mimics the process observed in natural evolution called genetic algorithms (GA). GA is a stochastic search algorithm inspired by the mechanics of natural evolution, including survival of the fittest, reproduction, cross-over and mutation. GA is based on a Darwinian survival of the fittest strategy and works with a population of individuals, each of which represents a potential solution to a given problem. Each individual or candidate solution in the population is generally represented in the GA as a linear string analogous to chromosomes. The basic algorithms in GA are selection, cross-over and mutation operators (called genetic operators). A complete description of GA can be found in Shopova and Vaklieva-Bancheva (2006). Instead of using the trial and error procedure, in several applications different GA approaches have been applied to optimize both the topology and parameters of neural networks (Ferentinos, 2005; Liu et al. 2007, Sarimveis et al. 2004).

In mathematical terms, a multi-objective problem consists of optimizing several objectives simultaneously, with a number of inequality or equality constraints. The problem can be formally written as follows:

$$\text{Find} : x = (x_i) \forall i = 1, 2, ...N_{param} \quad \text{such as}$$

$$f_k(x) \text{ is an optimum } \forall k = 1, 2, ...N_{obj} \text{ subject to:}$$

$$g_j(x) = 0 \quad \forall j = 1, 2, ..., M,$$

$$h_i(x) \le 0 \quad \forall i = 1, 2, ..., p,$$

where x is a vector containing the N_{param} design parameters, $(f_k)_{k=1-N_{obj}}$ the objective functions and N_{obj} the number of objectives. The objective function $(f_k)_{k=1-N_{obj}}$ returns a vector containing the set of N_{obj} values associated with the elementary objectives to be optimized simultaneously. The main principle of GA is to consider a so-called population of N individuals, i.e., a set of individuals covering the search domain, and to let it evolve along generations (or iterations) so that the best individuals survive and have offsprings, i.e., are taken into account and allowed to find better and better configurations.

Genetic Optimization of ANN

Many investigators (Hegazy et al., 1994) have therefore addressed these requirements by applying GAs in optimizing ANN 's parameters. In particular, GA is found to be quite useful and efficient when the exploration space of the ANN is extensive. The researches by Van Rooij et al. (1996) and Vonk et al.(1997) have proposed using evolutionary computations, such as Gas, in the field of ANN to generate both the ANN architecture and its weights. Those (Miller et al.,1989; Marshall and Harrison, 1991; Bornholdt and Graudenz, 1992) who supported the proposal were in favor of optimizing the connection weights and the architecture of ANNs using GAs. In the present work, a GA application process for optimizing the parameters (the number of neurons in the hidden layer, the coefficient of the learning rate and the momentum) of ANNs is shown in Figure 5 (Saemi et al. 2007). Chromosome population size and number of generations influence training time because the fitness value must be evaluated for every chromosome in every generation. Population sizes of 50–100 are commonly used in GA researches (Van Rooij et al. ,1996; Miller et al.,1989; Bornholdt and Graudenz, 1992; Saemi et al., 2007). Once the population size is chosen, the initial population is randomly generated. Hence, in this study, the initial population pool value is set to 60 chromosomes, which had at least one different value for the arrayed ANNs parameters. Every chromosome in a population evolved into new chromosomes for 100 generations.

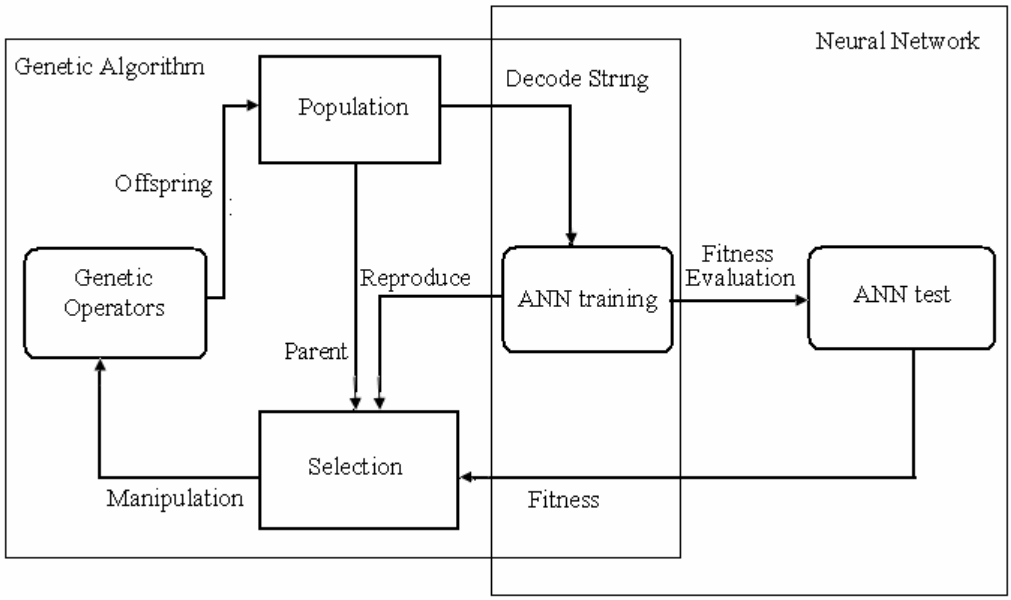

Figure 5: Combination of genetic algorithm and neural network

Genetic operators modify individuals within a population to produce a new individual for testing and evaluation. The cross-over operator takes two chromosomes to produce two new chromosomes by swapping some bits, thereby enlarging the search space and accelerating the process of reaching the optimal solution. In this study, a chromosome is represented by a binary string as in feature selection. Hence, crossover can be performed by arbitrarily choosing a point called the cross-over point, at which two chromosomes exchange their parts to create two new chromosomes. After the cross-over and mutation operations, a new population is generated and evaluated using the fitness function. In this section, the new hybrid approach is introduced, which is implemented as below:

1. It is begun with a first generation of N randomly selected chromosome strings. Each of them is divided into several sections, depending on the layers and nodes that are chosen for the ANN. Each chromosome string is represented as a set of the connection weights for the ANN and each section of the chromosome represented one weight. The training data are then submitted to the network.

2. A fitness measure for each string is established with the training data, as mentioned above, the lifetime utility function. A string's probability of being selected for reproduction is proportional to its fitness value. Mutation and crossover help to avoid premature convergence to local minima.

3. The matings, crossover and mutation create offspring that constitute the new generation. Decoding these new chromosomes, we gain a new set of weights and then submit them to the network, comparing the output with our target. If the training error meets the demand, then stop.

4. Go back to step 3 unless the total training amounts achieve the max time set in GA. Finally, the chromosome string with the smallest error for the training sample is selected to provide the final network connection weights. After each run, a new set of weights is obtained and replaces the old set of weights. Finally, one can get a best set of "prediction principles" or a best set of weights, and obtain a well-trained ANN without premature convergence to local minima.

RESULTS AND DISCUSSION

Genetic Optimization of the Multilayer Perceptron Method

The total number of data acquired at the time of this study added up to 87, all collected during one year by performing 87 different experiments. The database to be introduced to the neural network was broken down randomly into three groups: training, cross-validation, and verification. The network was trained using the training set data. The actual output of the training set data was used to develop the weights in the network. The test set was used to evaluate the predictive ability of the network. Training continued as long as the computed error

between the actual and predicted outputs for the test set was decreased. Typically, 80% of the data is used for training and validation purposes. The other 20% of the data is categorized as verification. The verification set is used to evaluate the accuracy of the newly trained network by providing the network with a set of data it has never seen. There is the possibility of using the current network weights or using the best network weights saved during a genetic training trial run. If a cross-validation set is used during the training, the best network weights are the ones that give the minimum cross-validation error. Otherwise, the best network weights are the ones that give the minimum training error. During the testing, the learning is turned off and the chosen data set is fed through the network. The network output is collected and a report is then generated showing the test results.

The number of hidden layers was defined as 2 layers and the numbers of input and output neurons in the GA-MLP model were defined as 3neurons (Magnetic field values, Inlet Power value, Volume fraction of nanosize) and 2 neurons (Thermal efficiency value, Thermal resistance value), respectively. By optimizing the ANN parameters in this study, the best GA- MLP model is achieved with a two hidden layer 3-9-6-2 neuron architecture.

The back-propagation training module shown in Figure 5 was used to evaluate all chromosomes, that is, after the parameter values for each chromosome were translated into the predefined ANN, the network was trained on the training data set and the cross-validation data set was used to prevent over-fitting and to test whether the stopping criteria were satisfied. In this study, the training process of the BPN (Back Propagation Neural network) stopped after a maximum of 6000 iterations or until there was no improvement of the MSE (because of memorizing and reaching local or global minima) for 100 iterations with the cross-validation data set. The fitness of every chromosome was evaluated by measuring the testing MSE, estimated from the results on a cross-validation data set. A better network has a lower training error, but requires a higher fitness value. The cross-validation data were not used to train the ANN model, but were used to test the ANN model in the training stage.

This process of phenotypic fitness measurement, selection, cross-over recombination, and mutation was iterated through 100 generations; the network with the lowest error in the 45th generation was designated as the optimal evolved network. The value of the average and minimum MSE of the test data are shown in Table 3, where the average MSE is the average value of the MSE for the whole set of chromosomes for each generation and the minimum MSE (best value) is the minimum ones in the whole population. The best result is the record of the minimum value for one particular GA run. The corresponding plots resulting from this table are shown in Figures 6 and 7 across all generations.

Table 3: Training and cross-validation error obtained from the trained GA-MLP network

Optimization Summary	Best Fitness	Average Fitness
Generation #	45	35
Minimum MSE	0.00070	0.00074

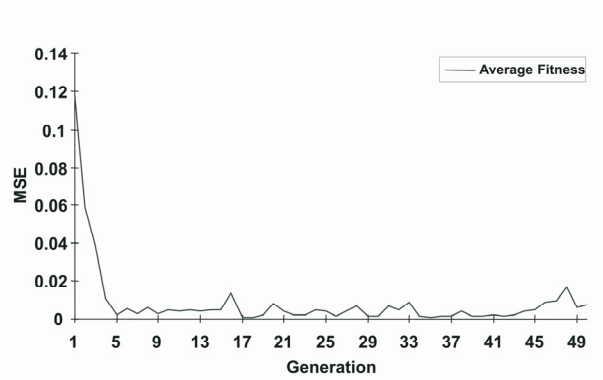

Figure 6: Average Fitness (MSE) versus Generation in GA-MLP.

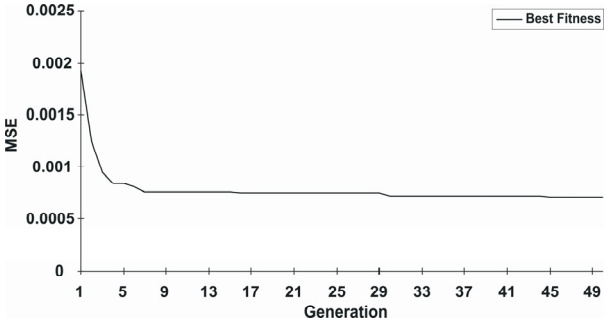

Figure 7: Best Fitness (MSE) versus Generation in GA-MLP.

In Figure 6 the average fitness achieved during each generation of the optimization is illustrated. The average fitness is the average of the minimum MSE (cross validation MSE) taken across all of the networks within the corresponding generation. The fitness function is an important factor for the convergence and the stability of a genetic algorithm. Therefore, the smallest fitness value was used to evaluate the convergence behavior of the GA. Figure 7 demonstrates the best fitness value versus the number of the generation. In this case, to evaluate the hybrid model generalization on the chosen data set or test data points, the performance of each model is reported in Table 4.

Table 4: Performance of the GA-MLP model for the test data set

Performance	Thermal efficiency	Thermal resistance
MSE	3.15E-05	0.001
NMSE	0.02	0.016
MAE	0.004	0.031
Min Abs Error	8.82E-06	0.009
Max Abs Error	0.01	0.054
r	0.98	0.999

Figure 8 shows the plot of the network output and the desired network output for each test data set. In this figure, each desired output (Thermal efficiency and Thermal resistance) is plotted as a solid color and the corresponding network output as a dashed line.

Figure 8 shows the actual efficiency and resistance that were measured in the experiment (and not presented to the network during the genetic training) in comparison with the network's estimation/ prediction for each sample. Results presented in Table 4 reveal that the correlation coefficient between the desired parameters (efficiency and resistance) and the GA-MLP output are 0.98 and 0.99. Accordingly, the genetically trained network is able to predict/estimate thermal efficiency and thermal resistance values comparable to those of the actual experimental measurements. On the other hand, the capabilities of the optimized neural networks in pattern recognition were established.

In Figures 9 and 10, the desired values (Thermal efficiency and resistance) versus GA-MLP predictions for the verification data points are shown. These figures reveal that an acceptable agreement (linear correlation coefficients are 0.98 for Thermal efficiency and 0.99 for Thermal resistance) between the predicted and operational data can be achieved.

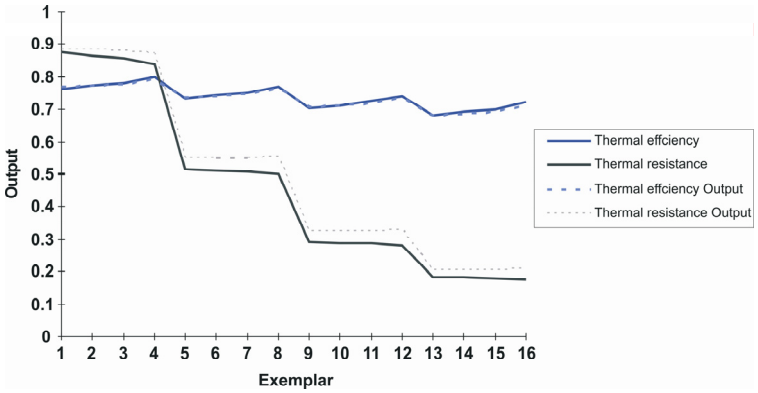

Figure 8: Desired output and actual GA-MLP network output

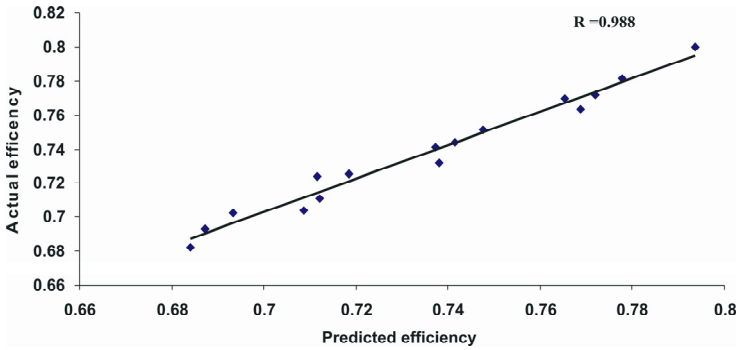

Figure 9: Thermal efficiency measurements versus GA-MLP network predictions for the test data.

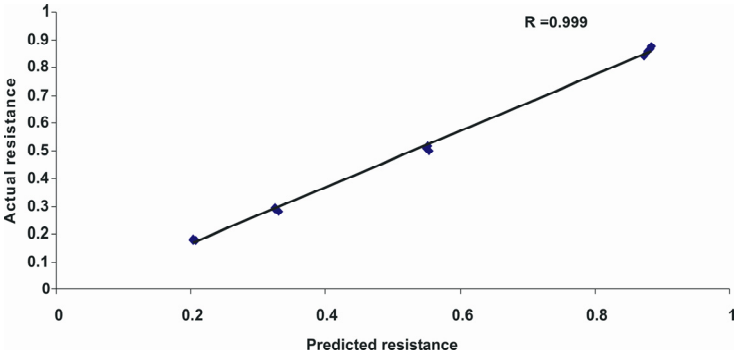

Figure 10: Thermal resistance measurements versus network predictions for the test data

CONCLUSION

In this study, a two-phase closed thermosyphon with an Ag/water nanofluid as working fluid and applying a magnetic field has been studied and a complete framework is presented for the development of a model for it. The experimental results show that the thermal resistance of the thermosyphon decreases with nanoparticle concentration as well as magnetic field strength. The GA-MLP method was programmed separately in the Matlab 7.6 environment and in NeuroSolutions Release 5.0 software and the results compared, being in good agreement. The execution time was less than 15 min with a Pentium IV 1.86 MHz processor, which shows that the method is not computationally demanding, considering that the algorithm needs to run only once. The performance of the nets with respect to the predictions made on the test sets showed that the neural network model incorporating a GA was able to estimate the thermally enhanced closed thermosyphon with high correlation coefficients. The correlation coefficients between the desired parameters (efficiency and resistance) and the GA-MLP output are 0.98 and 0.99 with respect to experimental data. Accordingly, the genetically trained network is able to predict/estimate thermal efficiency and thermal resistance values comparable to those of the actual experimental measurements.

ACKNOWLEDGMENT

The authors would like to thank Nano Research Center of Iran for financially support of this project.

NOMENCLATURE

b	Bias
B	Magnetic field strength (Tesla)
F	Transfer function
$(f_k)_{k=1-N_{obj}}$	Objective functions
N	The number of data in the training data set or the cross validation data set
N	Population of individuals
N_{obj}	Number of objectives
N_{param}	Design parameters
O_i	Desired output for the training data or cross-validation data i
P	Input vector
p	Number of input neurons
S^1	Number of hidden neurons
S^2	Number of output neurons
T_i	The network output for the training data or cross-validation data i
w_i	Individual weight constant
w	Synaptic weight parameter
$x_1, x_2, \ldots x_p$	Input variables
φ	Activation function
$\psi(x_1, x_2, ...x_p)$	Approximated function

REFERENCES

Anoop, K.B., Sundararajan, T., Sarit, K. D., Effect of particle size on the convective heat transfer in nanofluid in the developing region. Int. J. Heat Mass Transfer 52, p. 2189 (2009).

Bornholdt, S., Graudenz, D., General asymmetric neural networks and structure design by genetic algorithms. Neural Netw., 5, p. 327 (1992).

Braithwaite, D., Beaugnon, E., Tournier, R., Magnetically controlled convection in a paramagnetic fluid. Nature, 354, p. 134 (1991).

Das, S. K., Putra, N., Thiesen, P., Roetzel, W., Temperature dependence of thermal conductivity enhancement for nanofluids. J. Heat Transfer, 125, p. 567 (2003).

Eastman, J. A., Choi, S. U. S., Li, S., Yu, W., Thompson, L. J., Anomalously increased effective thermal conductivities of ethylene glycol-based nanofluids containing copper nanoparticles. Appl. Phys. Lett., 78, p. 718 (2001).

Faraday, M., On the diamagnetic conditions of flame and gases. Philos. Mag., 31, p. 401 (1847).

Ferentinos, K., Biological engineering applications of feed forward neural networks designed and parameterized by genetic algorithms. Neural Networks, 18, p. 934 (2005).

Hagan, M. T., Menhaj, M. B., Training feed forward neural network with the Marquardt algorithm. IEEE Trans. Neural Netw., 6, p. 5 (1994).

Hamilton, R. L., Crosser, O. K., Thermal conductivity of heterogeneous two component systems. Ind. Eng. Chem. Fundam., 1, p. 182 (1962).

Haykin, S., Neural Networks: A Comprehensive Foundation. 2nd ed., Prentice-Hall, New York, (1999).

Hegazy, T., Moselhi, O., Fazio, P., Developing practical neural network applications using back-propagation. Microcomput, 9, p. 145 (1994).

Holland, J. H., Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press (1975).

Jeyadevan, B., Koganezawa, H., Nakatsuka, K., Performance evaluation of citric ion stabilized, Magn. Magn. Mater., 85, p. 207 (1990).

Kang, S., Wei, W., Tsai, S., Huang, C., Experimental investigation of nanofluids on sintered heat pipe thermal performance, J. Appl. Thermal Eng., 29, p. 973 (2009).

Khandekar, S., Joshi, Y. M., Mehta, B., Thermal performance of closed two-phase thermosyphon using nanofluids. Int. J. of Thermal Sci., 47, p. 659 (2008).

Liu, X., Cheng, X., Wu, W., Peng, G., A neural network for predicting moisture content of grain drying process using genetic algorithm. Food Control, 18, p. 928 (2007).

Maki, S., Tagawa, T., Ozoe, H., Enhanced convection or quasi-conduction states measured in a super-conducting magnet for air in a vertical cylindrical enclosure heated from below and cooled from above in a gravity field. J. Heat Transfer, 124, p. 667 (2002).

Marshall, S. J., Harrison, R. F., Optimization and training of feed forward neural networks by GAs. Proceeding of IEEE Second International Conference on Artificial Neural Networks, p. 39 (1991).

Miller, G., Toddv P., Hedge, S., Designing neural networks using genetic algorithms. Proceeding of the Third International Joint Conference on Genetic Algorithms, p. 379 (1989).

Nakatsuka, K., Hama, Y., Takahashi, J., Heat transfer in temperature-sensitive magnetic fluids. J. Magn. Magn. Mater.. 85, p. 207 (1990).

Noie, S. H., Zeinali Heris, S., Kahani, M., Noie, S. M., Heat transfer enhancement using Al_2O_3/water nanofluid in a two-phase closed thermosyphon. Int. J. Heat Fluid Flow, 30, p. 700 (2009).

Pauling, L., Wood, R. E., Sturdivant, J. H., An instrument for determining the partial pressure of oxygen in a gas, J. Am. Chem. Soc., 68, p. 795 (1946).

Saemi, M., Ahmadi, M., Varjani, A.Y., Design of neural networks using genetic algorithm for the permeability estimation of the reservoir. J. Petrol. Sci. Eng., 59, p. 97 (2007).

Sarimveis, H., Alexandridis, A., Mazarakis, S., Bafas, G., A new algorithm for developing dynamic radial basis function neural network models based on genetic algorithms. Comput. and Chem. Eng., 28, 9. 209 (2004).

Schmidt, J. F., Effects of a magnetic field on the conduction heat transfer at the stagnation point of a partially ionized argon gas. Nasa TN D-3251, (1966).

Shopova, E. G., Vaklieva-Bancheva N. G., BASIC—A genetic algorithm for engineering problems solution. Comput. and Chem. Eng., 30, p. 1293 (2006).

Silveston, P. L., Wa¨medurchgang in waagerechten Flussigkeitsschichten, Part 1, Forsch, Ing. 24, p. 29 and p. 59 (1958).

Van Rooij, A. J. F., Jain, L. C., Johnson, R. P., Neural network training using genetic algorithms. World Scientific Publishing Co. Pvt. Ltd, Singapore (1996).

Vonk, E., Jain, L. C., Johnson R. P., Automatic generation of neural network architecture using evolutionary computation. World Scientific Publishing Co. Pvt. Ltd, Singapore (1997).

Wang, X. Q., Mujumdar, A. S., A review on nanofluids – part I: theoretical and numerical investigations. Braz. J. Chem. Eng. 25 (4), p. 613-630 (2008).

Wang, X. Q., Mujumdar A. S., A review on nanofluids - part II: experiments and applications, Braz. J. Chem. Eng. 25 (4) p. 631-648 (2008).

Wasp, F. J., Solid–Liquid Flow Slurry Pipeline Transportation. Trans. Tech. Pub., Berlin (1977).

Xuan Y., Li Q., Heat transfer enhancement of nanofluids. Int. J. Heat Fluid Flow 21, p.58 (2000).

Yu, W., Choi, S. U. S., The role of interfacial layers in the enhanced thermal conductivity of nanofluids: A renovated Maxwell model. Journal of Nanoparticle Research 5, p. 167 (2003).