Master's Thesis

Determination of Near-global Optimal Initial Weights of Artificial Neural Networks Using Harmony Search Algorithm: Application to Breakwater Armor Stones

Harmony Search Algorithm 을 이용한
인공신경망 근사 전역 최적 초기 가중치 결정:
방파제 피복석에의 적용

February 2016

Graduate School of Civil and Environmental Engineering
Seoul National University

Anzy Lee

Determination of Near-global Optimal Initial Weights of Artificial Neural Networks Using Harmony Search Algorithm: Application to Breakwater Armor Stones

Supervisor: Kyung-Duck Suh

Submitting a master's thesis

December 2015

Graduate School of Civil and Environmental Engineering
Seoul National University

Anzy Lee

Confirming the master's thesis written by Anzy Lee

December 2015

Chair _____(Seal)

Vice Chair _____(Seal)

Examiner _____(Seal)

Abstract

Artificial neural networks (ANNs) have become a popular tool as the efficient model for prediction and forecast in various areas. Despite a great number of application in numerous researches, ANNs are hardly recognized as a generalized tool because of its characteristics. The back propagation (BP) algorithm helps to find the optimal values of the weights and biases of the neural networks that correspond to the minimum value of a performance function usually defined as the root-mean squared error between output variable and target variable. However, the BP is based on the gradient descent method which can give the local minimum value of a specified function and which is sensitive to the initial values of the weights and biases. To search for the global minimum of the performance function, the Monte-Carlo simulation generating a number of ANNs having different initial weights and biases has been suggested to search the global minimum of the performance function. However, it is not efficient and it takes a long time.

In this study, an ANN model is developed to predict the stability number of breakwater armor stones based on the experimental data reported by Van der Meer in 1988. To resolve the fundamental problems in neural networks due to local minimization, the harmony search (HS) algorithm is used. Firstly, the HS algorithm would find the weights which have the near-global minimum of the performance function. The optimized weights found by HS are then used as the initial weights for the ANNs and further modified by the BP algorithm. The BP training based on the gradient descent method would allow fine adjustment of the weights.

To assess the reliability of the ANN model with BP training and the ANN-HS model, both models were run 50 times and the statistical analysis was conducted for the model results. Each of harmony memory considering rate (HMCR) and pitch adjustment rate (PAR) of HS has five different values varying from 0.1 to 0.9 at an interval of 0.2. The correlation coefficient (r) and index of agreement (I_a) between model output values and target values in the validation data were used to evaluate the performance of the models. It was shown that the ANN-HS models with HMCR=0.9 and PAR=0.1 and HMCR=0.7 and PAR=0.5 give more accurate and consistent prediction ability than the general ANN model trained by BP algorithm.

keywords: armor stone, artificial neural network model, harmony search algorithm, stability number

Student Number: 2014-20546

Table of Contents

List of Table

List of Figures

List of Symbols

Latin Uppercase

D_{n50}	Nominal size of the armor unit
H_s	Significant wave height in front of the structure
I_a	Index of agreement
K_D	Stability coefficient
N_s	Stability number
N_w	Number of waves
P	Permeability of core
S	Damage level
SS	Spectral shape
T_m	Average period
T_p	Peak period

Latin Lowercase

a_1^o	Output value of output layer after passing transfer function
$\mathbf{a}_1^o$	Output vector of output layer after passing transfer function
a_k^h	Output value of *k*th neuron in the hidden layer after passing transfer function

$\mathbf{b}$	Bias vector
b_1^o	Bias of a neuron in the output layer
b_k^h	Bias of kth neuron in the hidden layer
e_i	Estimated stability number
$\overline{e}$	Average of estimated stability number
f^h	Activation function of hidden layer
f^o	Activation function of output layer
h	Water depth
h / H_s	Dimensionless depth
i	Number of input variable in general ANN structure
m	Number of principal components
m_i	Measured stability number
$\overline{m}$	Average of measured stability number
n	Number of neural networks in ensemble modeling
n_1^o	Output value of output layer before passing transfer function
n_k^h	Output value of kth neuron in the hidden layer before passing transfer function
$\mathbf{p}$	Input vector
p_j	jth input variable
r	Correlation coefficient
$\mathbf{w}$	Weight vector
$\mathbf{w}^p$	Weight matrix in the principal component analysis

w_{1j}^{o} Weight connecting a neuron in the output layer and *j*th neuron in the hidden layer

w_{kj}^{h} Weight connecting *k*th neuron and *j*th input variable in the hidden layer

w_{kj}^{p} Weight converting *k*th input variable and *j*th experimental parameter

$\mathbf{x}$ Experimental parameter in group 2

$\mathbf{x}'$ Experimental parameter in group 1 subtracted by its mean

x_k *k*th experimental parameter in group 2

x_j' *j*th experimental parameter in group 1 subtracted by its mean

y_i *i*th output value

y_i^k *i*th output value obtained from the *k*th members

Greek Uppercase

Δ Relative mass density

Greek Lowercase

α Angle of structure slope

ε Error function

μ Damping parameter

θ Factor for damping parameter

ρ_w Density of water

ρ_s Density of armor unit

$\boldsymbol{\tau}$	Target value vector
ξ_m	Surf similarity parameter
ξ_c	Critical surf similarity parameter

CHPTER 1. INTRODUCTION

1.1 Background

Artificial neural networks (ANNs) have become a popular tool as the efficient model for prediction and forecast in various areas, including finance, medicine, power generation, water resources and environmental science. Although the basic concept of artificial neurons was first proposed in 1943 (McCulloch and Pitts, 1943), applications of ANNs have blossomed after the introduction of the backpropagation training algorithm for feedforward ANNs in 1986, and the improvement of calculation ability accelerated the employment of ANNs (Rumelhart et al., 1986).

Since the 1990s, ANNs have been utilized in coastal and nearshore researches (Mase et al., 1995; Tsai and Lee, 1999; Cox et al., 2002; Kim and Park, 2005; Van Gent et al., 2007; Browne et al., 2007; Yoon et al., 2013). In the early years (1992~1998), ANNs were believed as a novel model which is applicable to various kinds of problems, and researches were conducted to examine their practicability as an alternative model. Despite a great number of applications in numerous researches, ANNs are hardly recognized as a mature modeling approach as other numerical and statistical modeling methods because of its characteristics. The back propagation (BP) algorithm helps to find the optimal values of the weights and biases of the neural networks that correspond to the minimum value of a

performance function usually defined as root-mean squared error. However, the BP with the gradient descent method can give the local minimum values of a specified function and it is sensitive to the initial values of weight and bias. To search for the global minimum of the performance function, Monte-Carlo simulation generating a number of ANNs having different initial weights and biases is suggested, but it is not efficient and it takes a long time. Moreover, best fitting ANNs from the simulation could not or would not be accepted to have a global minimum of the performance function. Even if we find the global optimal weights of ANNs by training, they cannot be reproduced by the general users of the ANN model. Accordingly, the development of robust ANN model is needed to alleviate the local minimization problem of the BP training algorithm.

1.2 Previous Studies

1.2.1 Generalization of ANN model

A proper neural network model should provide high prediction accuracy for the test and validation data set as well as for the training data. However, the performance of the general ANN model is very sensitive to the initial weights, thereby not giving the stable results in the test data. The erratic output obtained from ANN model forces the modelers to choose the best model result without any explanation of setting the optimal initial weights. Consequently, it is difficult to use the ANN model as a robust and generalized prediction model.

Many researchers have shown that the BP algorithm is greatly dependent on the initial weights, and they implemented various methods to solve this problem. Kolen and Pollack (1990) demonstrated that the BP training algorithm has a large dependence on the initial weights by performing a Monte-Carlo simulation. On the other hand, Yam and Chow (1995) proposed an algorithm based on least-squares methods to determine the optimal initial weights. The algorithm showed that the model's dependence on the initial weights can be reduced. In addition, genetic algorithm (GA) has been applied to search for the optimal initial weights of ANNs and had improved the model accuracy (Venkatesan et al., 2009; Chang et al., 2012; Mulia et al., 2013). Furthermore, ensemble methods have been implemented to enhance the accuracy of the model. They also overcome the dependence of the ANN model not only on the initial weights but also on training algorithms and data structure (Krogh and Velelsby, 1995; Boucher et al., 2009; Zamani et al., 2009; Kim 2014).

The problem of BP can be resolved by applying the methods mentioned above. In particular, the genetic algorithm finds near-global optimal initial weights of ANN (Montana, 2005). This method is relatively efficient and effective in making the result of ANN stable. In this study, the harmony search algorithm, a music-inspired metaheuristic optimization algorithm, is used to find the near-global optimal initial weights of ANNs, thereby enhancing the accuracy and stability of ANN model. The structure of the HS algorithm is much simpler than other metaheuristic algorithms. In addition, the intensification procedure conducted by HS algorithm encourages to speed up the convergence by exploiting the history and experience in the search process. Thus, the HS is expected to find the near-global optimal initial weights of the ANN.

1.2.2 Stability number of armor stones

To determine the optimum weight of armor stones for sloping revetments and breakwaters, the Hudson (1959) formula has been widely used probably because of its simplicity. However, it has been found to have a lot of shortcomings. It does not include, for example, the influence of wave period and does not take into account random waves. To solve the shortcomings in the Hudson formula, Van der Meer (1987, 1988) proposed a new design formula, based on a series of more than 250 model tests of Van der Meer (1988), which includes the influence of wave period, number of waves, armor grading, wave spectrum shape, groupiness of waves and the permeability of the core.

The model test data of Van der Meer (1988) have been used by many researchers in different types of research. Yoo et al. (2001) used the data to develop a new

design formula, which is simpler than and comparable in accuracy to the Van der Meer formula (see Suh and Yoo 2003). The data have also been used in the development of artificial neural network (ANN) model (Mase et al. 1995; Kim and Park 2005; Balas et al. 2010), fuzzy model (Erdik 2009), or M5' model tree (Etemad-Shahidi and Bonakdar 2009) to predict the damage level or stability number of rock armors.

The stability number is a dimensionless number which measures the stability of the armor layer of a rubble mound structure. It is defined as

$$N_s \equiv \frac{H_s}{\Delta D_{n50}} \tag{1.1}$$

where H_s is the significant wave height in front of the structure, $\Delta = \rho_s / \rho_w - 1$ is the relative mass density, ρ_s and ρ_w are the mass densities of armor unit and water, respectively, and D_{n50} is the nominal size of the armor unit (Van der Meer, 1988). Fig. 1.1 shows the cross section of typical emerged rubble mound breakwater.

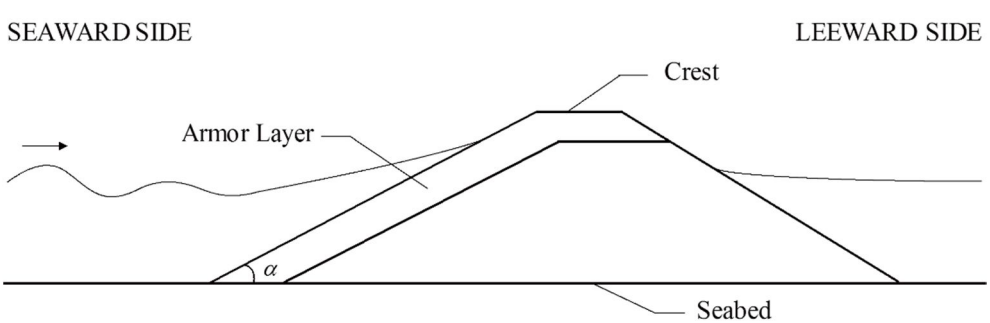

Figure 1.1 Cross section of emerged rubble mound breakwater

To estimate the stability number, it is required to determine the relationship between the stability number and other variables which would describe the characteristics of waves and structure. Unfortunately, the physical mechanism of displacement of armor units due to waves is so complicated that the analytic solution is hardly found. For this reason, plenty of experiments which include various physical factors of waves and structure were conducted to propose empirical formulas explaining the relationship. Hudson (1959) suggested an empirical formula:

$$N_s = (K_D \cot\alpha)^{1/3} \tag{1.2}$$

where α is the angle structure slope shown in Fig. 1.1 and K_D is the stability coefficient which depends on the shape of the armor unit, placement method, the location at the structure (i.e. trunk or head), and whether the structure is subject to breaking wave or non-breaking wave. Even though it is very simple, the Hudson formula has been found to have a lot of shortcomings.

To solve the main shortcomings of the Hudson formula, Van der Meer (1988) conducted an extensive series of tests including the parameters which are considered to have significant effects on armor stability, and the empirical formula based on the experimental data was proposed by Van der Meer (1987, 1988) as follows.

$$N_s = \frac{1}{\sqrt{\xi_m}}\left[6.2P^{0.18}\left(\frac{S}{\sqrt{N_w}}\right)^{0.2}\right] \quad \text{for } \xi_m < \xi_c \tag{1.3a}$$

$$N_s = 1.0P^{-0.13}\left(\frac{S}{\sqrt{N_w}}\right)^{0.2}\sqrt{\cot\alpha}\,\xi_m^{\,P} \quad \text{for } \xi_m \geq \xi_c \tag{1.3b}$$

where $\xi_m = \tan\alpha / \sqrt{2\pi H_s / gT_m^{\,2}}$ is the surf similarity parameter based on the average wave period T_m, and $\xi_c = \left(6.2P^{0.31}\sqrt{\tan\alpha}\right)^{1/(P+0.5)}$ is the critical surf similarity parameter indicating the transition from plunging waves to surging waves.

On the other hand, with the recent developments in computational intelligence, particularly in the area of machine learning, various data-driven models have been developed, based on the extensive experimental data of Van der Meer (1988), as described in the Introduction. A brief summary is given here only for the ANN models. Mase et al. (1995) constructed an ANN by the randomly selected 100 experimental data set of Van der Meer (1988) and by 5,000 epoch, which means the number of modification of the weights and biases. They used 579 experimental data excluding the data of low-crested structures. They employed six input variables: P, N_w, S, ξ_m, h / H_s, and the spectral parameter, where h is the water depth in front of the structure. Kim and Park (2005) followed the Mase et al.'s (1995) approach, but they used 641 data including low-crested structures. Believing that the predictability of a neural network increases as the input dimension increases, they split the surf similarity parameter into wave steepness and structure slope, and further the wave steepness into wave height and period.

They showed that the ANN gives better performance as the input dimension increases. It is known that in general the bias error and variance error decreases and increases, respectively, with the increase of input dimension. If the decreasing rate of bias error is greater than the increasing rate of variance error, the overall error decreases, and vice versa (Geman et al. 1992). It seems that the former is true for the Van der Meer's (1988) data. On the other hand, Balas et al. (2010) developed hybrid ANN models with PCA based on 554 data of Van der Meer (1988). They developed four different models by systematically reducing the data from 554 to 166 by using PCA or by using the PCs as the input variables of the ANN. Table 1.2 shows the correlation coefficients of different studies, which will be compared with that of the present study later.

Table 1.1 Correlation coefficients of different empirical formula or ANN models.

Author	Correlation coefficient	Remarks
Van der Meer (1987)	0.92	Empirical formula, Eq. (1.3)
Mase et al. (1995)	0.91	Including data of Smith et al. (1992)
Kim and Park (2005)	0.902 to 0.952	Including data of low-crested structures
Balas et al. (2010)	0.906 to 0.936	ANN-PCA hybrid models

1.3 Research Objective and Thesis Overview

The methodological issues on the proper selection and preprocessing of the input and output variables and the choice of the structure of the ANNs have been evaluated traditionally. Especially, this study focuses on the problem of the choice of an optimal ANN initial weights to resolve the issues on the generalization of ANN model by implementing the harmony search (HS) algorithm.

In this study, an ANN model is developed to predict the stability number of breakwater armor stones based on the experimental data reported by Van der Meer in 1988. To resolve the fundamental problems in neural networks due to local minimization, HS algorithm is used. Firstly, the HS algorithm would find the weights which have the near-global minimum value of the performance function. The optimized weights found by HS are then used as the initial weights for the ANNs and further modified by the BP algorithm. The BP training based on the gradient descent method would allow fine adjustment of the weights.

To assess the reliability of the ANN model with BP training and the ANN-HS model, both models are run 50 times and the statistical analysis is conducted for the model results. The correlation coefficient(r) and index of agreement(I_a) between model output values and target values in the validation data are used to evaluate the performance of the models.

CHAPTER 2. THEORETICAL BACKGROUNDS

2.1 Artificial Neural Networks (ANNs)

An ANN model is a powerful data-driven model aiming to mimic the systematic relationship between input and output data by training the network based on a large amount of data. Training is to modify the weights that connect the input and output variables so that the output values calculated by the model are as close as possible to the target (observed) values. An ANN model is composed of the information-processing units called neurons, which are fully connected with different weights indicating the strength of the relationships between input and output data.

2.1.1 Structure of ANNs

An ANN contains supervised learning such as pattern recognition and multi-layer perceptron and unsupervised learning such as self-organized map. Especially the multi-layer perceptron is considered as an alternative of empirical formulas because it takes into account the nonlinear relationship among input variables in a more effective and simpler way than other nonlinear regression methods by training or revising the weight matrix that links the input and output data. In this study, a two-layer perceptron model with the input layer and one hidden layer is developed as shown in Figure 2.1.

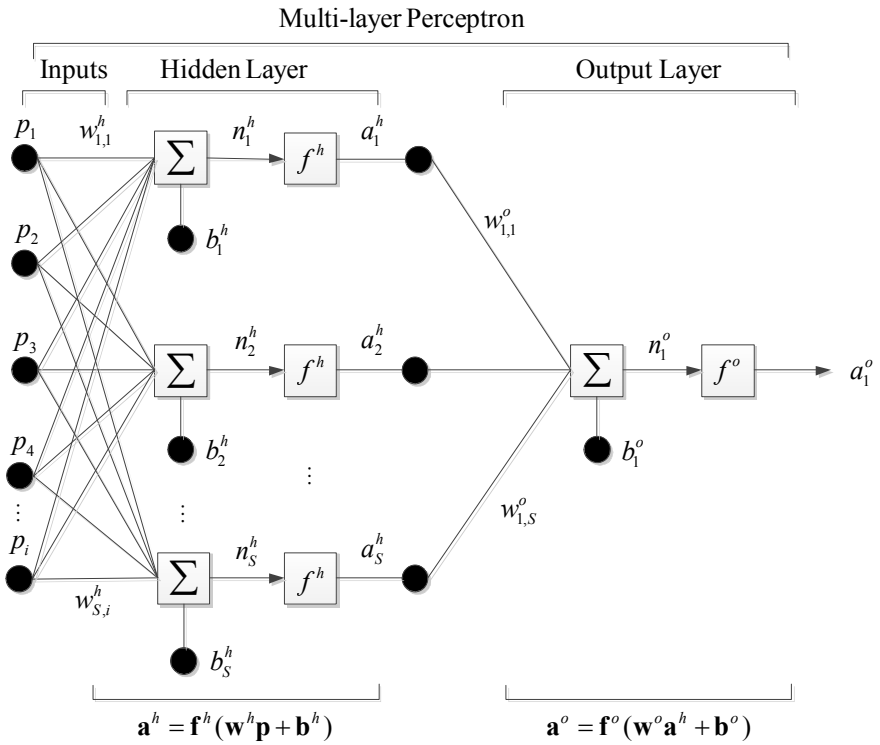

Figure 2.1 General structure of ANN

Here, i is the number of input variables. Firstly, for each of the input and output variables, the data are normalized so that the whole data are distributed in the range of $[\min, \max] = [-1, 1]$. This can be done by subtracting the average from the data values and rescaling the resulting values in such a way that the smallest and largest values become -1 and 1, respectively. Secondly, the initial weights in the hidden layer are set to have random values between -1 and 1, and the initial biases are all set to zero. The next step is to multiply the weight matrix by the input data, $\mathbf{p}$, and add the bias so that

$$n_k^h = \sum_{j=1}^{R} w_{kj}^h p_j + b_k^h, \quad k = 1 \text{ to } S \tag{2.1}$$

where R and S are the numbers of input variables and hidden neurons, respectively, and $\mathbf{p}$, $\mathbf{b}^h$, and $\mathbf{w}^h$ are the input variable, and bias and weight in the hidden layer, respectively. The subscripts of the weight w_{kj}^h are written in such a manner that the first subscript denotes the neuron in question and the second one indicates the input variable to which the weight refers. The n_k^h calculated by Eq. (2.1) is fed into an activation function, f^h, to calculate a_k^h. Hyperbolic tangent sigmoid function is used as the activation function so that

$$a_k^h = \frac{e^{n_k^h} - e^{-n_k^h}}{e^{n_k^h} + e^{-n_k^h}} \tag{2.2}$$

In the output layer, the same procedure as that in the hidden layer is used except that only one neuron is used so that

$$n_1^o = \sum_{j=1}^{S} w_{1j}^o a_j^h + b_1^o \tag{2.3}$$

and the linear activation function is used to calculate a_1^o so that

$$a_1^o = n_1^o \tag{2.4}$$

2.1.2 Training algorithm

With the randomly selected initial weights and biases, the neural network cannot

accurately estimate the required output. The weights and biases are continuously modified by the so-called training so that the difference between the model output and target (observed) value becomes small. To train the network, the error function is defined as the sum of square of the difference, i.e.

$$\varepsilon = \left\| \boldsymbol{\tau} - \mathbf{a}_1^o \right\|^2 \tag{2.5}$$

where $\| \ \|$ indicates a norm, and $\boldsymbol{\tau}$ is the target value vector to be sought.

In neural networks system, many kinds of gradient-based methods are frequently used: Steepest descent gradient algorithm, Newton's method, Conjugate gradient algorithm, Quasi-Newton algorithm and Levenberg-Marquardt algorithm. The gradient descent method means that it always heads towards a solution by lowering the error of the network – it has a direction. Considering the fact that the final goal of back-propagation training in ANN is to minimize the error function, we can say that the procedure of finding out the optimal weights and biases of ANNs is exactly the same as solving an optimization problem whose objective function is the error function. The subject of the optimization is the weights and biases of ANNs.

Generally, Levenberg-Marquardt algorithm, one of the gradient descent methods is widely used to train the networks. However it has a problem of local minimization. Figure2.2 illustrates the possible situation of local minimization. The value of the error function varies with different weights and biases. If the initial weights and biases are fortunately close enough to the values that have global minimum in the error function, the gradient method would reach the global

minimum by following the slope of the error function. On the other hand, as expected in most cases, if they are chosen to be far from the optimal values as shown by the green dot in the figure, their final destination would be the local minimum which is indicated as the red dot. As a consequence of local minimization, most of ANN models provide erroneous result. This unstable characteristic of local minimization prevents ANNs to be utilized as prediction model. In order to find a global minimum, optimal points from a number of ANN models having different initial weight and bias are calculated and the point which has the minimal error function among them would be designated as the global minimum optimal point. In this study, the Harmony Search algorithm is used to find the global minimum point easily.

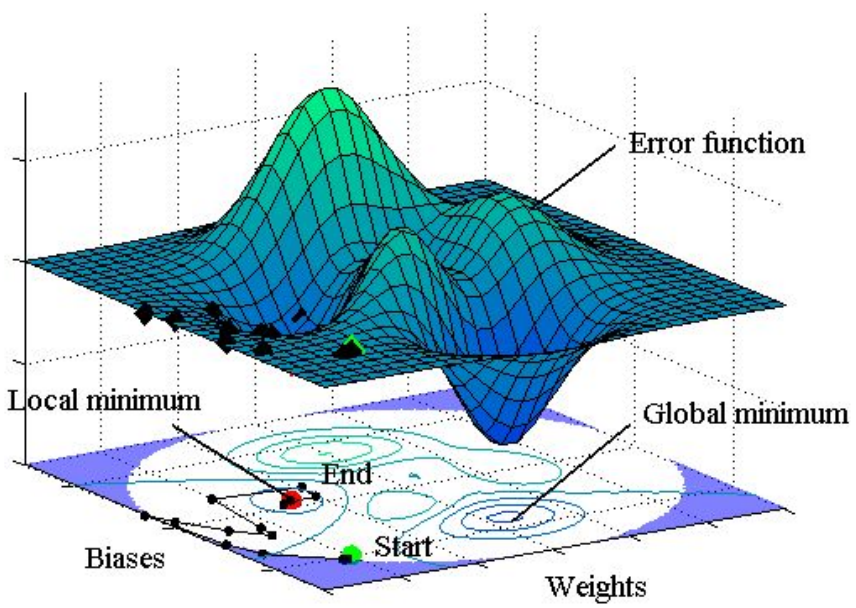

Figure 2.2 Local minimization problem

2.2 Harmony Search (HS) Algorithm

The HS algorithm is a music-based metaheuristic optimization algorithm (Geem et al., 2001). It was first developed by Geem et al. in 2001 and has been vigorously applied to many different optimization problems such as function optimization, design of water distribution networks, engineering optimization, groundwater modeling, model parameter calibration, etc. The objective of this algorithm is to find a perfect state of harmony, and the effort to search the harmony in music is similar to finding the optimality in an optimization process. In other words, both of them are looking for an optimal state. The HS algorithm consists of five steps as follows (Lee and Geem, 2004).

Step 1. Initialization of the algorithm parameters

Generally, the problem of global optimization can be written as

$$\begin{aligned} &\text{Minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad x_i \in \mathbf{X}_i, \quad i = 1, 2, ..., N \end{aligned} \tag{2.6}$$

where $f(\mathbf{x})$ is an objective function; $\mathbf{x}$ is the set of each decision variable; and $\mathbf{X}_i$ is the set of possible range of values for each decision variable which can be denoted as $\mathbf{X}_i = \{x_i(1), x_2(2), ..., x_i(K)\}$ for discrete decision variables satisfying $x_i(1) < x_2(2) < \cdots < x_i(K)$ or ${}_L x_i \le \mathbf{X}_i \le {}_U x_i$ for continuous decision variables. In addition, N is the number of decision variables and K is the number of possible values for the discrete variables. Also, there exists HS algorithm parameters which are required to solve the optimization problems: harmony

memory size (HMS, number of solution vectors), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) and termination criterion (maximum number of improvisation). HMCR and PAR are the parameters used to improve the solution vector.

Step 2. Initialization of harmony memory

As shown in Eq. 2.7, the harmony memory (HM) matrix is composed with as many randomly generated solution vectors as the size of the HM. And they are stored with the values of the objective function, $f(\mathbf{x})$, ascendingly.

$$\mathrm{HM} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^{HMS} \end{bmatrix} \tag{2.7}$$

Step 3. Improvise a new harmony from the HM

A new harmony vector, $\mathbf{x}' = \left(x_1', x_2', ..., x_N'\right)$, is created from the HM based on assigned HMCR, PAR, and randomization. For example, the value of the first decision variable $\left(x_1'\right)$ for the new vector can be selected from any value in the designated HM range, $x_1^1 \sim x_1^{\mathrm{HMS}}$. In the same way, the values of other decision variables can be selected. The HMCR parameter, which varies between 0 and 1, is a possibility that the new value is selected from HM as follows:

$$x_i' \leftarrow \begin{cases} x_i' \in \left\{x_i^1, x_i^2, ..., x_i^{\mathrm{HMS}}\right\} & \text{w.p. HMCR} \\ x_i' \in \mathbf{X}_i & \text{w.p. } (1-\mathrm{HMCR}) \end{cases} \tag{2.8}$$

The HMCR is the probability of selecting one value from the historic values stored in the HM, and (1-HMCR) is the probability of randomly taking one value from the possible range of values. This procedure is analogous to the mutation operator in genetic algorithms. For instance, if a HMCR is 0.95, the HS algorithm would pick the decision variable value from the HM including historically stored values with a 95% of probability. Otherwise, with a 5% of probability, it takes the value from the entire possible range. A low memory considering rate selects only few best harmonies and it may converge too slowly. If this rate is near 1, most of the pitches in the harmony memory are used, and other ones are not exploited well, not leading into good solutions. Therefore, typically $\text{HMCR} = 0.7 \sim 0.95$ are recommended.

On the other hand, the HS algorithm would examine every component of the new harmony vector, $\mathbf{x}' = (x_1', x_2', ..., x_N')$, to decide whether it has to be pitch-adjusted or not. In this procedure, the PAR parameter which sets the probability of adjustment for the pitch from the HM is used as follows:

$$\text{Pitch adjusting decision for } x_i' \leftarrow \begin{cases} \text{Yes} & \text{w.p. PAR} \\ \text{No} & \text{w.p. (1 - PAR)} \end{cases} \tag{2.9}$$

The pitch adjusting procedure is conducted only after a value is selected from the HM. The value (1-PAR) is the probability of doing nothing. To be specific, if the value of PAR is 0.1, the algorithm will take a neighboring value with $0.1 \times \text{HMCR}$ probability. For example, if the decision for x_i' in the pitch adjustment process is Yes, and x_i' is considered to be $x_i(k)$, then the kth element

in $\mathbf{X}_i$, or the pitch-adjusted value of $x_i(k)$ is changed into

$$\begin{aligned} x_i' &\leftarrow x_i(k+m) \quad \text{for discrete decision variables} \\ x_i' &\leftarrow x_i' + \alpha \qquad \text{for continuous decision variables} \end{aligned} \tag{2.10}$$

where m is the neighboring index, $m \in \{..., -2, -1, 1, 2, ...\}$; α is the value of $bw \times u(-1,1)$; bw is an arbitrary chosen distance bandwidth or fret width for the continuous variable; and $u(-1, 1)$ is a random number from uniform distribution with the range of $[-1, 1]$. If this rate is very low, because of the limitation in the exploration of a small subspace of the whole search space, it slows down the convergence of HS. If a pitch-adjusting rate is very high, it may cause the solution to scatter around some potential optima. Therefore, $\mathrm{PAR} = 0.1 \sim 0.5$ are used in most applications. The parameters HMCR and PAR help the HS algorithm to find globally and locally to improve the solution, respectively.

Step 4. Update the HM

If the new harmony vector gives better performance than the worst harmony in the HM, evaluated in terms of the value of objective function, the new harmony would be included in the harmony memory and the existing worst harmony is eliminated from the harmony memory.

Step 5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

The iterations are terminated if the stop criterion is satisfied. If not, Steps 3 and 4 would be repeated.

The HS algorithm can be summarized as the pseudo code shown in Figure 2.3. The presented pseudo code can be also expressed as the flow chart in Fig. 2.4.

Harmony Search

begin
Define objective function f(x), $x = (x_1, x_2, \ldots, x_d)^T$
Define harmony memory accepting rate (r_{accept})
Define pitch adjusting rate (r_{pa}) *and other parameters*
Generate Harmony Memory with random harmonies
while *(t<max number of iterations)*
while *(i<=number of variables)*
if *(rand<HMCR), Choose a value from HM for the variable i*
if *(rand<PAR), Adjust the value by adding certain amount*
end if
else *Choose a random value*
end if
end while
Accept the new harmony (solution) if better
end while
Find the current best solution
End

Figure 2.3 Pseudo code of the Harmony Search algorithm (Geem, 2009)

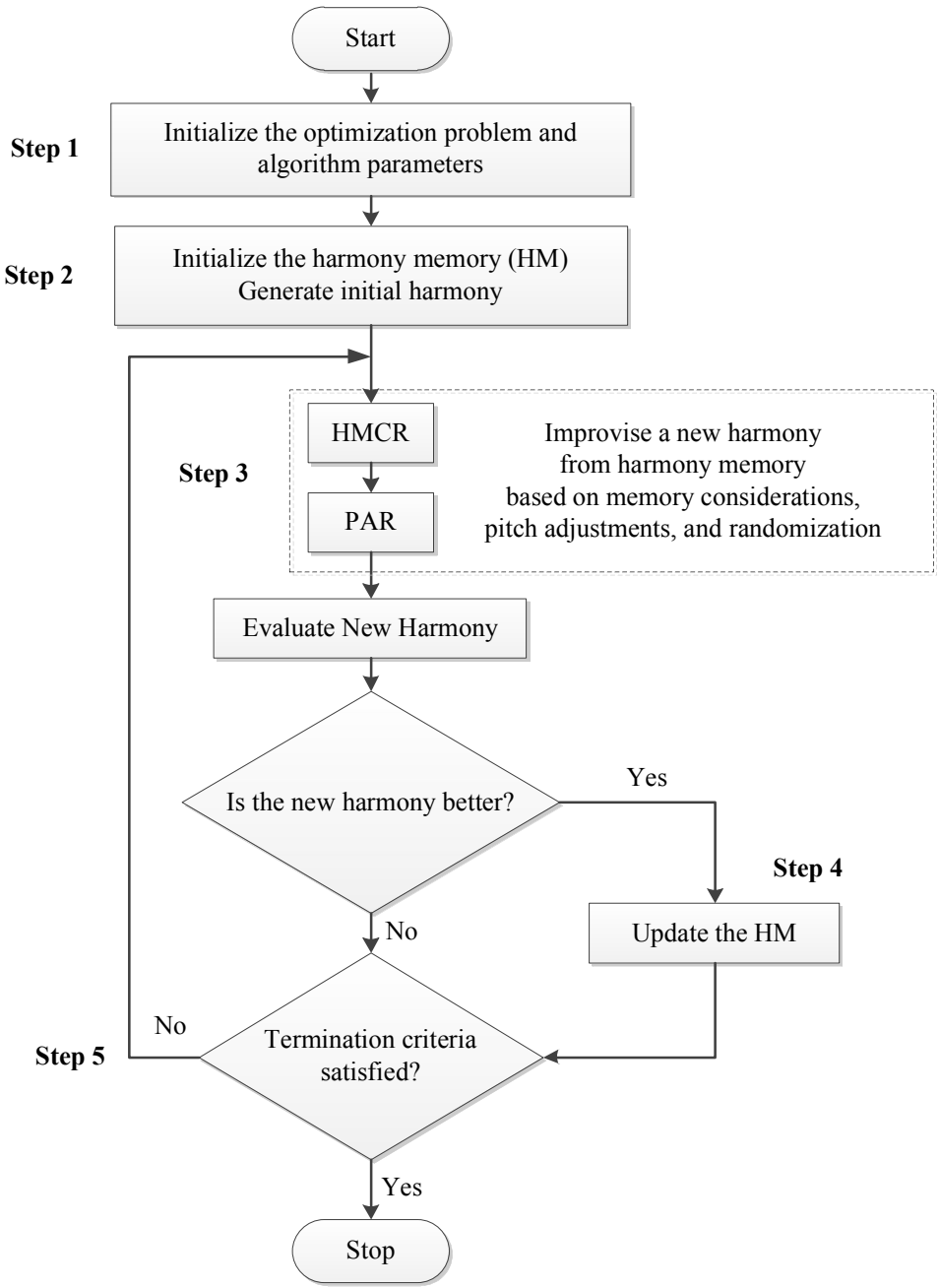

Figure 2.4 Flow chart of original HS algorithm (Lee and Geem, 2004)

CHAPTER 3. METHODOLOGY

3.1 Sampling for Training Data

Since training a network model is aiming to fit the system of ANNs to the training dataset, the whole dataset should be divided into two parts: one for learning and the other for validation, or test. The selected, or sampled dataset for training is important and has to include the characteristic of population since the training stage determines the feature of the networks. Otherwise, data-driven model cannot appropriately predict the data that had not been used in the learning process. For example, if the training data consists of only relatively small values, ANNs would estimate the small values well but not the large values because they do not learn the distinctive systematic relationship made in larger values and vice versa. To avoid this problem, the stratified sampling method can be used which makes the probability mass/density function (PMF/PDF) of sampled dataset be similar to that of population dataset. Unfortunately, because each case of the experimental data is dependently related with conditions which are the variables of dataset, the training data have to be chosen manually. Then, data for training and test are statistically analyzed to evaluate whether they have same PMF or not.

There are two kinds of tests to conduct the PMF analysis: parametric statistical test and nonparametric statistical test. In this study, the PMF of each variable in the experimental data does not follow the normal distribution, so one of the

nonparametric statistical test, Chi-square test is applied. In addition, as done by Mase et al. (1995), randomly selected 100 experimental data set was used to train the network, and the remaining 479 data were used to test the model.

The chi-square (χ^2) test is fundamentally based on the error between the assumed and observed PMF/PDF of the distribution (Haldar and Mahadevan, 2000). In the χ^2 goodness-of-fit test, the range of the *n* observed data is separated into m intervals. Also, the number of frequencies (n_i) of the random variable in the *i*th interval is counted (*i* = 1 to *m*). Further, the observed frequencies $n_1, n_2, ..., n_m$ and the corresponding theoretical frequencies $e_1, e_2, ..., e_m$ of an assumed distribution are compared. As the total sample points n tends to ∞ , it can be shown (Hoel, 1962) that the quantity

$$\sum_{i=1}^{m} \frac{\left(n_i - e_i\right)^2}{e_i} \tag{3.1}$$

approaches the χ^2 distribution with $f = m - 1 - k$ degree of freedom where k is the number of parameters in the assumed distribution from the data and m is the number of intervals. The degree of freedom f is a parameter of the χ^2 distribution. Significance levels, α , between 1% and 10% are commonly used. The assumed distribution is acceptable at significance level α if the value calculated from Eq. 3.1 is less than this value. Thus, the observed distribution can be considered to follow the assumed distribution with the significance level α if

$$\sum_{i=1}^{m} \frac{(n_i - e_i)^2}{e_i} < c_{1-\alpha,f} \tag{3.2}$$

Here, $c_{1-\alpha,f}$ indicates the value of the χ^2 distribution with f degree of freedom at Cumulative Density Function (CDF) of $(1-\alpha)$. In this study, 5% of significance level is chosen. Input and output variables in the ANN shown in Table 3. 1 are tested separately using χ^2 test.

Table 3.1 Input parameters of neural network models

Input parameters	Target parameters
P, N_w, S, $\cot\alpha$, H_s, T_p, h/H_s, SS	N_s

Previously, Kim and Park (2005) found that the stability number is more accurately predicted when using the observation parameters themselves rather than using the parameters composed with multiplication of several parameter as input variable. To be specific, it is more suitable to use significant wave height (H_s), mean wave period (T_m), and $\cot\alpha$ than using the surf similarity parameter, $\xi_m = \tan\alpha / \sqrt{2\pi H_s / gT_m^2}$ as input variable. Here, peak period, T_p, was used instead of T_m because it contains information about spectral shape as well as the mean wave period.

The neural network can deal with qualitative data by assigning the values to them. The permeability coefficients of impermeable core, permeable core, and homogeneous structure are assigned to 0.1, 0.5, and 0.6, respectively, as done by Van der Meer (1987). On the other hand, the spectral shapes of narrowband,

medium-band (i.e. Pierson-Moskowitz spectrum), and wideband are assigned to 1.0, 0.5, and 0, respectively, as done by Mase et al. (1995).

To perform the chi-square test, firstly, ranges of each variable are defined from 1 to 10 or 11 as shown in Table 3.2. In the training dataset, the observed frequency ($n_1, n_2, ..., n_m$) of each variable is counted as in Table 3.3.

Table 3.2 Definition of range

Range	*Ns*	*P*	*Nw*	*S*	$\cot\alpha$	*Hs*	*Tp*	*h/Hs*	*SS*
1	0.79 – 1.149	0.1 – 0.15	1000 – 1200	0 – 1.5	2 – 2.4	0 – 0.1	0 – 0.25	0 – 1	0.0045 – 0.1015
2	1.149 – 1.508	0.15 – 0.2	1200 – 1400	1.5 – 4.5	2.4 – 2.8	0.1 – 0.3	0.25 – 0.75	1 – 3	0.1015 – 0.1985
3	1.508 – 1.867	0.2 – 0.25	1400 – 1600	4.5 – 7.5	2.8 – 3.2	0.3 – 0.5	0.75 – 1.25	3 – 5	0.1985 – 0.2955
4	1.867 – 2.226	0.25 – 0.3	1600 – 1800	7.5 – 10.5	3.2 – 3.6	0.5 – 0.7	1.25 – 1.75	5 – 7	0.2955 – 0.3922
5	2.226 – 2.585	0.3 – 0.35	1800 – 2000	10.5 – 13.5	3.6 – 4	0.7 – 0.9	1.75 – 2.25	7 – 9	0.3925 – 0.4895
6	2.585 – 2.944	0.35 – 0.4	2000 – 2200	13.5 – 165	4 – 4.4	0.9 – 1.1	2.25 – 2.75	9 – 11	0.4895 – 0.5865
7	2.944 – 3.303	0.4 – 0.45	2200 – 2400	16.5 – 19.5	4.4 – 4.8	1.1 – 1.3	2.75 – 3.25	11 – 13	0.5865 – 0.6835
8	3.303 – 3.662	0.45 – 0.5	2400 – 2600	19.5 – 22.5	4.8 – 5.2	1.3 – 1.5	3.25 – 3.75	13 – 15	0.6835 – 0.7805
9	3.662 – 4.021	0.5 – 0.55	2600 – 2800	22.5 – 25.5	5.2 – 5.6	-	3.75 – 4.25	15 – 17	0.7805 – 0.8775
10	4.021 – 4.38	0.55 – 0.6	2800 – 3000	25.5 – 28.5	5.6 – 6	-	4.25 – 4.75	17 – 19	0.8775 – 0.9745
11	-	-	-	28.5 – 31.5	-	-	4.75 – 5.25	-	-

Table 3.3 Frequency analysis of sampled(training) data

Range	N_s	P	N_w	S	$cot\alpha$	H_s	T_p	h/H_s	SS
1	4	56	52	10	40	29	-	-	8
2	15	-	-	31	-	66	-	1	-
3	22	-	-	21	37	1	-	18	-
4	21	-	-	12	-	1	13	35	-
5	16	-	-	12	13	1	29	23	87
6	9	-	-	4	-	1	22	15	-
7	8	-	-	3	-	1	11	5	-
8	2	38	-	3	-	-	15	2	-
9	1	-	-	2	-	-	5	-	-
10	2	6	48	1	10	-	-	1	5
11	-	-	-	1	-	-	5	-	-
Sum	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Next, the observed frequencies are compared with the corresponding theoretical frequencies $e_1, e_2, ..., e_m$ of an assumed distribution which are given in Table 3.4. Using Table 3.3 and Table 3.4, we can obtain the residual chart calculated based on Eq. (3.1) as in Table 3.5.

Table 3.4 Frequency analysis for population data

Range	N_s	P	N_w	S	$cot\alpha$	H_s	T_p	h/H_s	SS
1	3.45	59.93	50.78	12.26	37.82	30.40	-	-	6.56
2	15.20	-	-	30.05	-	65.98	-	3.45	-
3	22.11	-	-	20.55	37.31	0.35	-	17.79	-
4	20.90	-	-	11.74	-	0.69	16.75	32.64	-
5	16.93	-	-	10.88	15.72	1.38	28.84	24.35	86.87
6	10.02	-	-	4.84	-	0.86	21.42	14.51	-
7	7.43	-	-	3.63	-	0.35	11.05	4.49	-
8	2.42	34.54	-	2.59	-	-	14.51	2.07	-
9	0.69	-	-	1.04	-	-	3.80	-	-
10	0.86	5.53	49.22	0.52	9.15	-	-	0.69	6.56
11	-	-	-	1.90	-	-	3.63	-	-
Sum	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 3.5 Residual chart

Range	N_s	P	N_w	S	$cot\alpha$	H_s	T_p	h/H_s	SS
1	0.09	0.26	0.03	0.42	0.13	0.06	-	-	0.31
2	0.00	-	-	0.03	-	0.00	-	1.74	-
3	0.00	-	-	0.01	0.00	1.24	-	0.00	-
4	0.00	-	-	0.01	-	0.14	0.84	0.17	-
5	0.05	-	-	0.12	0.47	0.11	0.00	0.08	0.00
6	0.10	-	-	0.14	-	0.02	0.02	0.02	-
7	0.04	-	-	0.11	-	1.24	0.00	0.06	-
8	0.07	0.35	-	0.06	-	-	0.02	0.00	-
9	0.14	-	-	0.90	-	-	0.38	-	-
10	1.50	0.04	0.03	0.45	0.08	-	-	0.14	0.37
11	-	-	-	0.43	-	-	0.52	-	-
Sum	1.99	0.64	0.06	2.67	0.68	2.81	1.77	2.20	0.69
f	9	2	1	10	3	6	6	7	2
5%	**16.82**	**5.991**	**3.841**	**18.31**	**7.86**	**12.59**	**12.59**	**14.07**	**5.99**

'f' and '5%' denote degree of freedom and $c_{0.95,f}$, the value of the χ^2 distribution with f degree of freedom at CDF of 0.95. Previously, f was defined as $m-1-k$. Here, m is the number of intervals. Only the intervals that have the frequency of at least one are counted. k is set to 0 for nonnormal distribution. As a result, Eq. (3.2) is satisfied for all the variables. Therefore, we can conclude that the PMF of training dataset is acceptable within a 5% significance level.

Based on the frequency analysis in Table 3.3 and Table 3.4, the PMFs of each variable in sampled and population data can be illustrated as in Figure 3.1 to Figure 3.9. Comparing the two PMF graphs in each figure, we can visually confirm that their distributions are very similar.

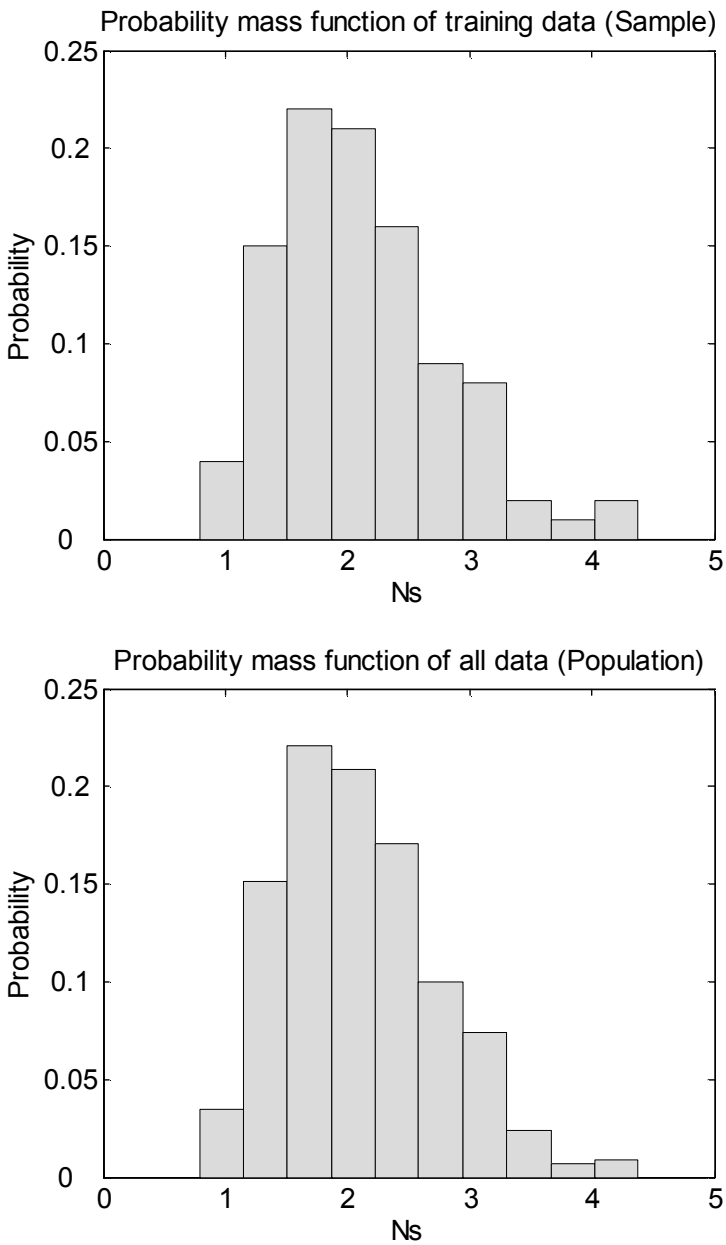

Figure 3.1 Probability mass function of N_s

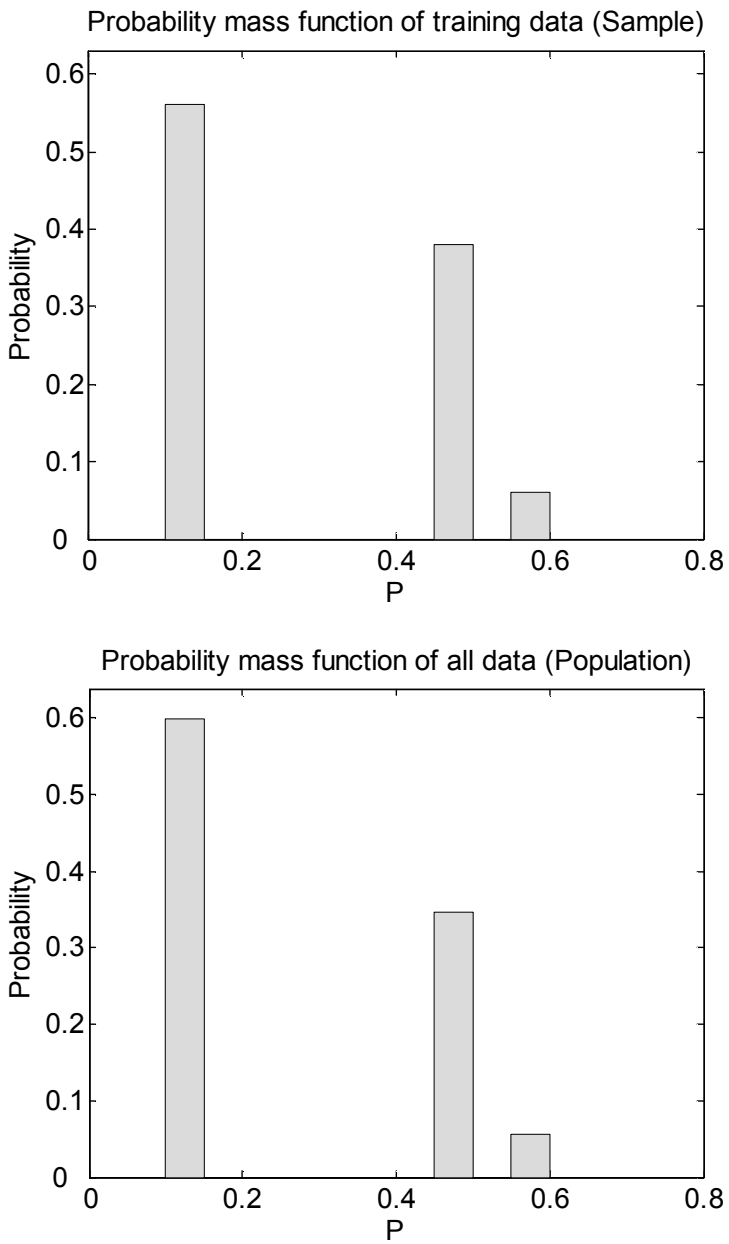

Figure 3.2 Probability mass function of *P*

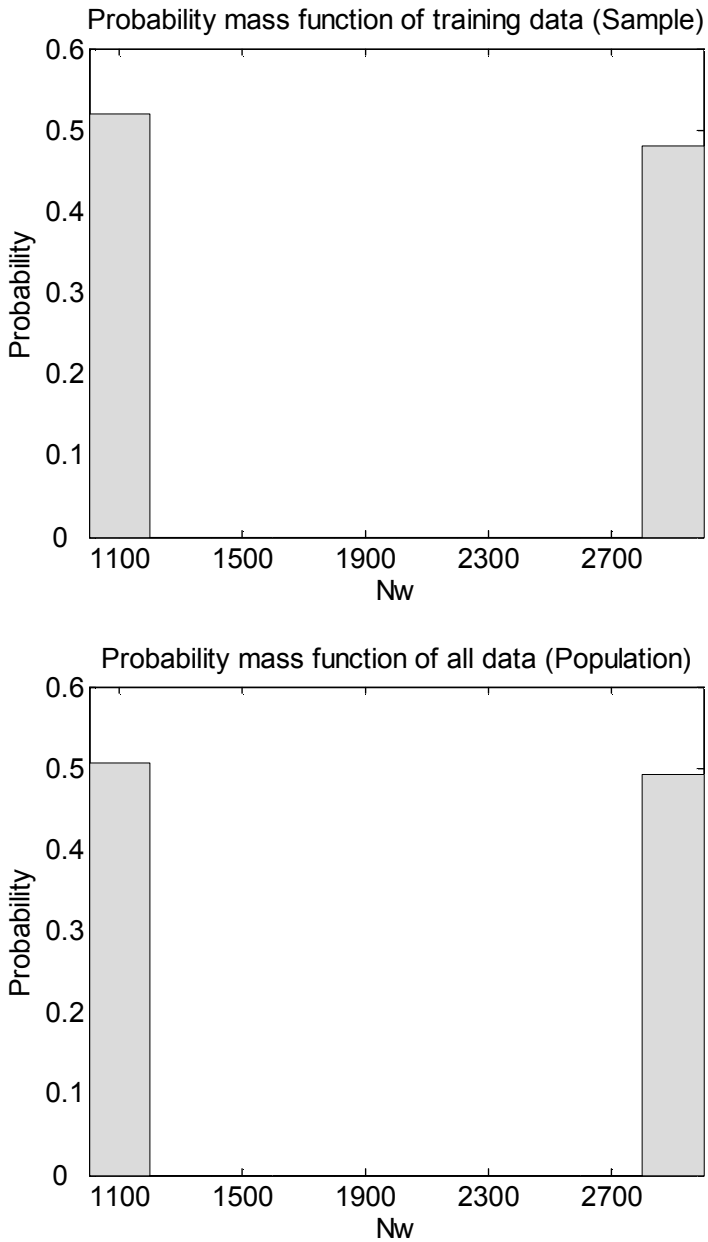

Figure 3.3 Probability mass function of N_w

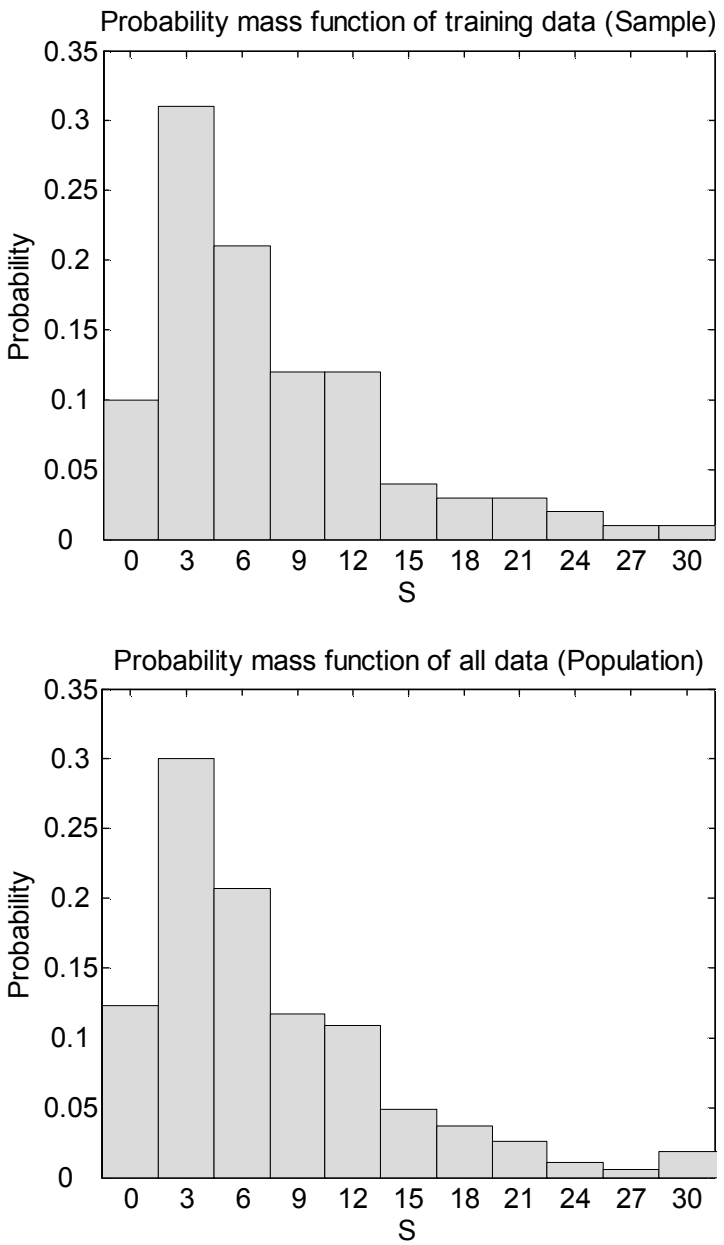

Figure 3.4 Probability mass function of *S*

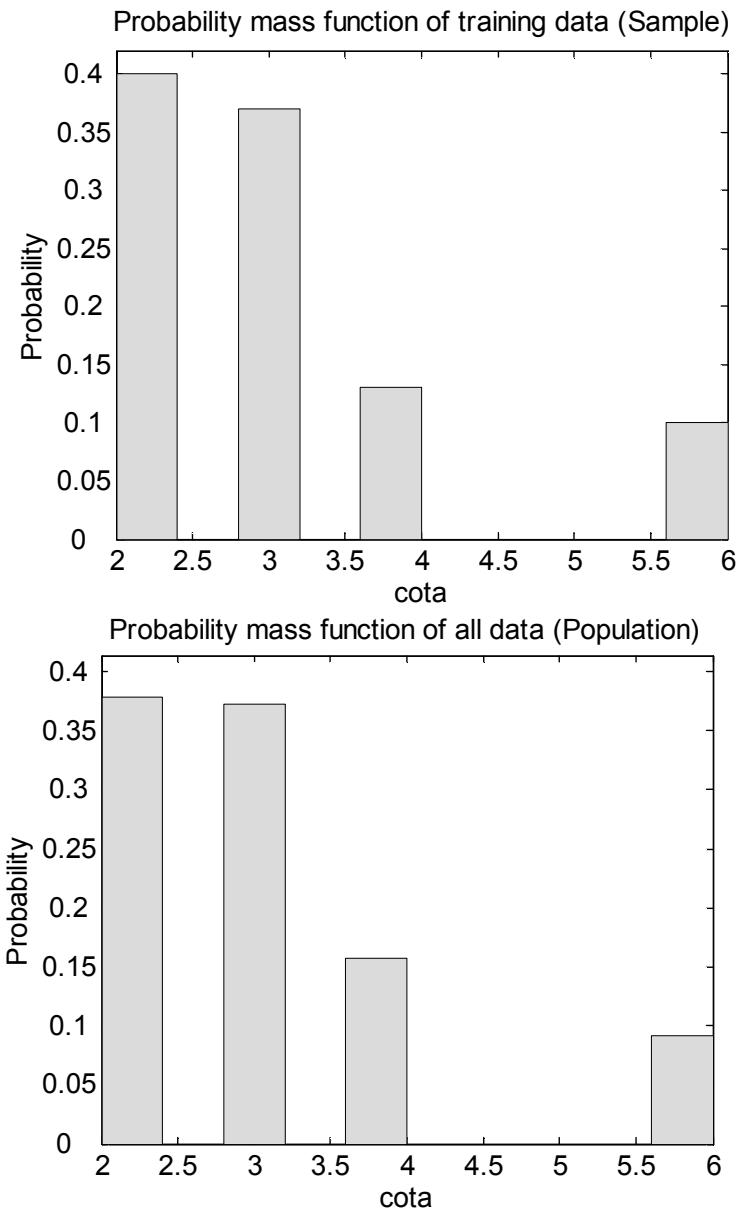

Figure 3.5 Probability mass function of cot α

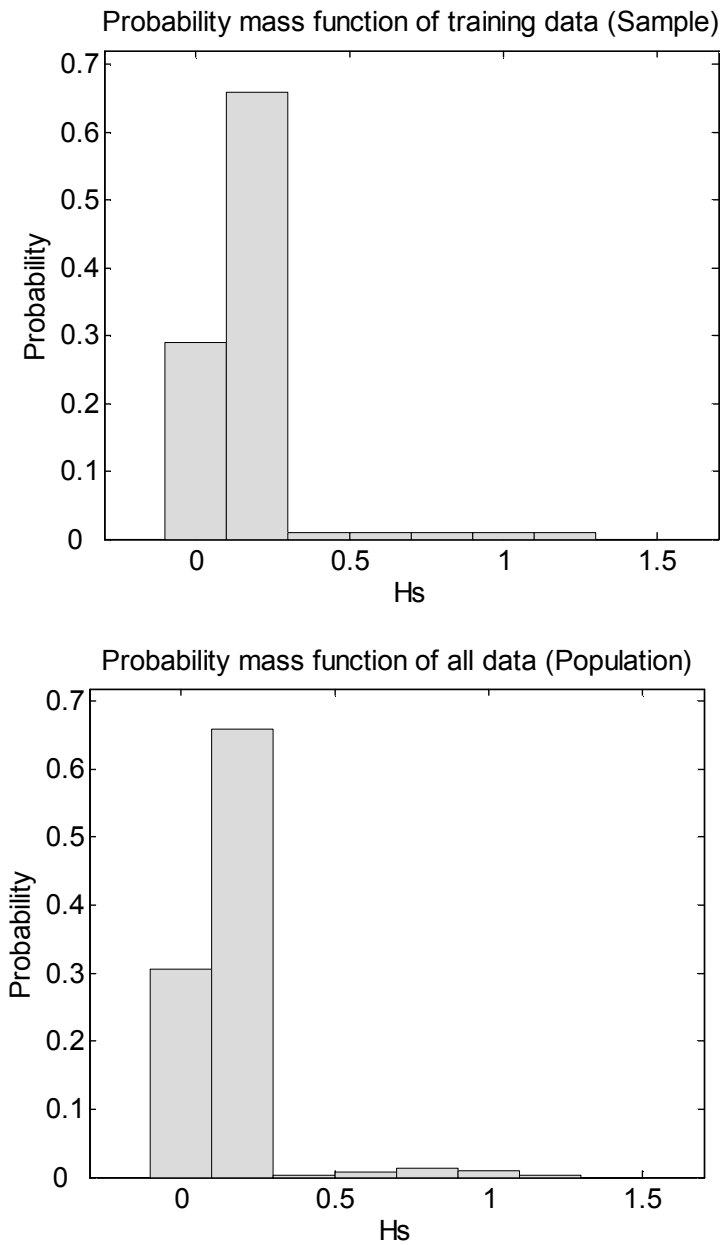

Figure 3.6 Probability mass function of H_s

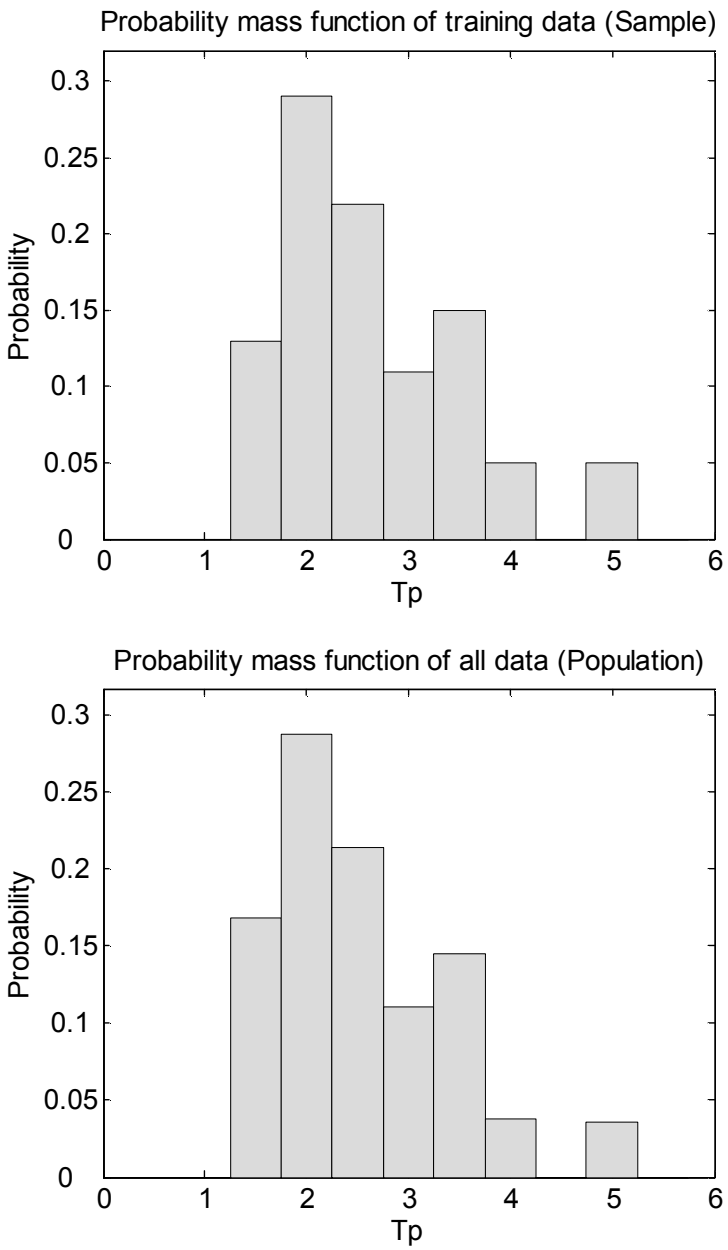

Figure 3.7 Probability mass function of T_p

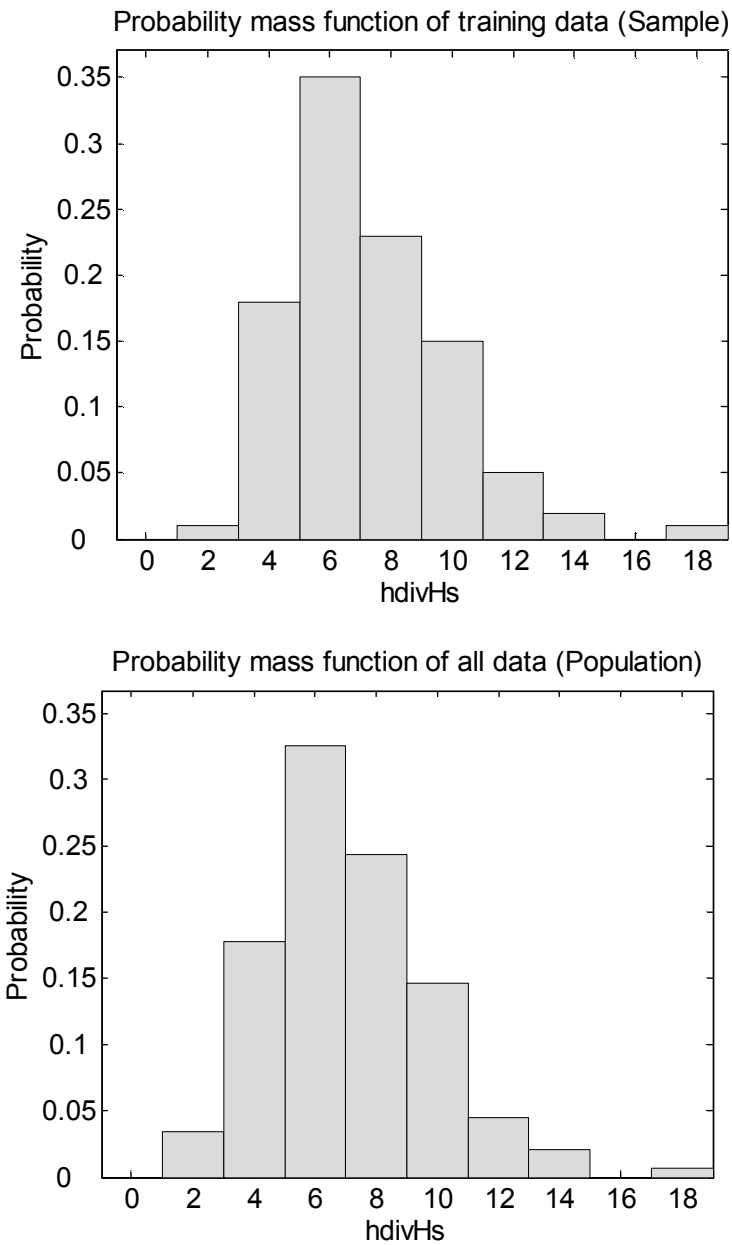

Figure 3.8 Probability mass function of h/H_s

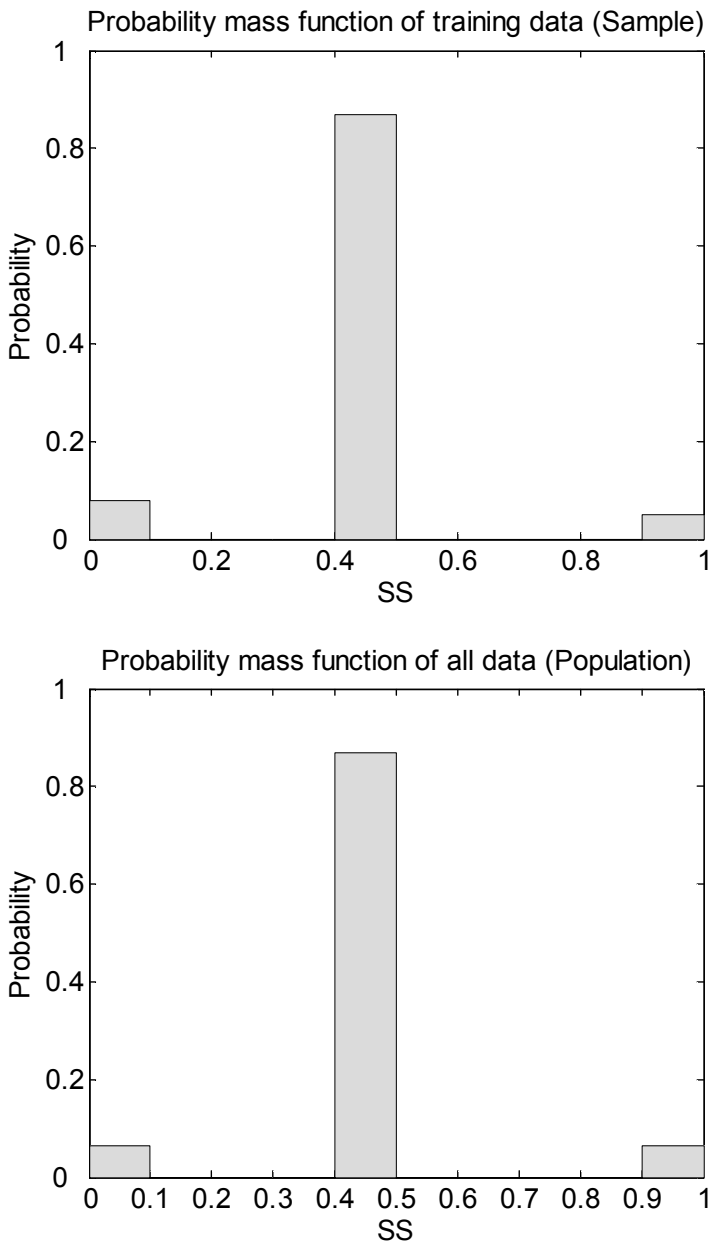

Figure 3.9 Probability mass function of *SS*

3.2 Model design

On the basis of the selected experimental data, an ANN model and ANN-HS model were constructed. In this research, the condition and parameter of ANNs in both models were identical. The following sections explain these settings.

3.2.1 ANN model

Considering the efficiency of the model, the number of neurons is fixed to have four. To minimize the error function, the Levenberg-Marquardt algorithm is used, which is the standard algorithm of nonlinear least-squares problems and the detailed description of which can be found in Press et al. (1992). Like other numeric minimization algorithms, the Levenberg-Marquardt algorithm is an iterative procedure. It necessitates a damping parameter μ, and a factor θ which is greater than one.

In this study, $\mu = 0.001$ and $\theta = 10$ were used. If the squared error increases, then the damping is increased by successive multiplication by θ until the error decreases with a new damping parameter of $\mu\theta^k$ for some k. If the error decreases, the damping parameter is divided by θ in the next step. The training was stopped when the epoch reached 50,000 or the damping parameter was too large for more training to be performed.

3.2.2 ANN-HS model

As previously mentioned, since ANN training algorithms are based on BP, they

are prone to local minima problems. On the other hand, Evolutionary Algorithms (EA) like HS are directed to stochastic. They are not as efficient as BP for training neural nets. They start search from random points, and slowly converge to a solution. To solve local optimization problems, initial weights of the networks are determined by HS as long as the stop condition evaluated by the fitness function (i.e. Root Mean Squared Error) is not satisfied. If the stop condition is satisfied, the detailed optimization process using back propagation to tune the parameters (i.e. weights and biases) would start with the initial condition obtained from the HS optimization as illustrated in Figure 3.10.

First, ANNs are created with the randomly designated initial weights. The ANNs calculate the output data and evaluate its fitness function or error function. If certain stop condition is not satisfied, the initial weights would follow the HS optimization procedure. The HS finally finds new combination harmony which would be evaluated again. After going through the optimization process with HS, weights of the networks are modified and the output variables of the networks are adjusted to be more close to the target variables because its objective is to minimize their differences. Finally, the obtained weights are used to set the initial condition for the further BP training.

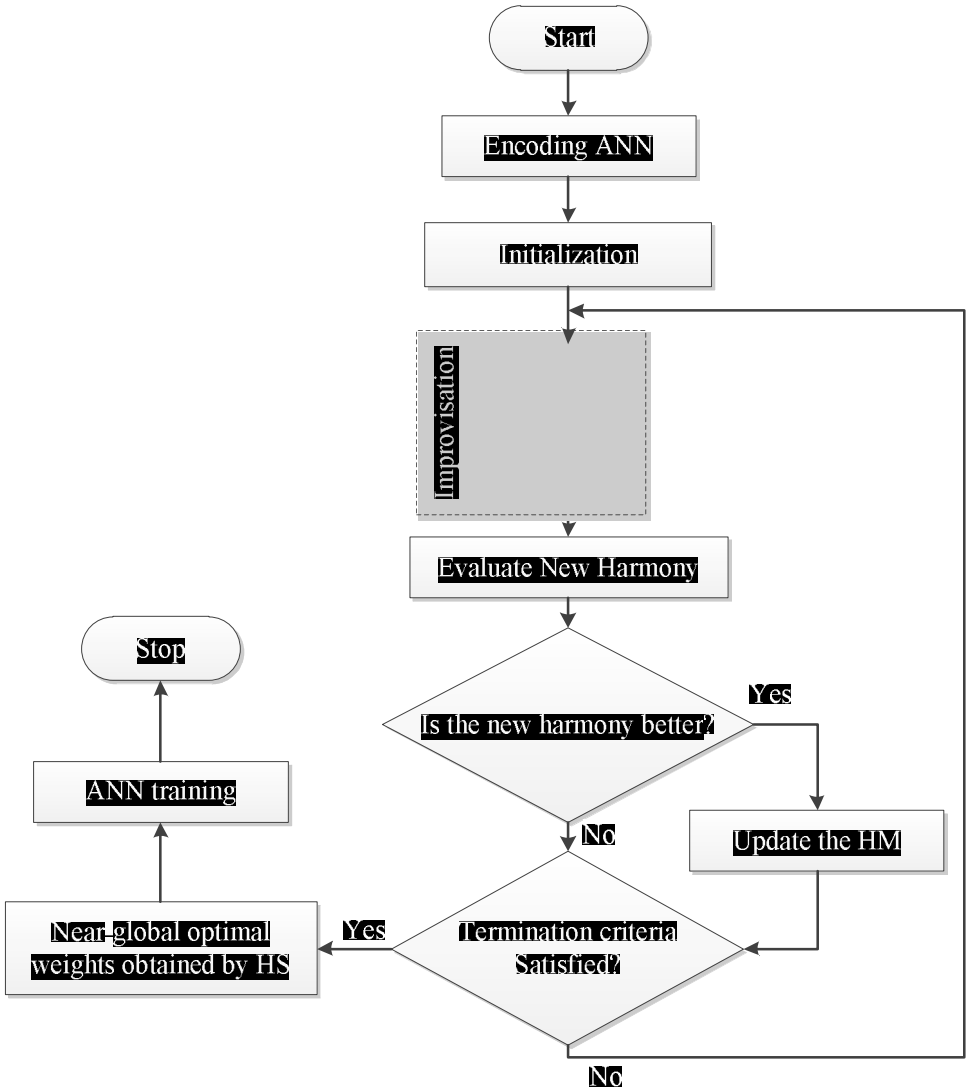

Figure 3.10 Flow chart of ANN-HS

The number of input variables is eight, hidden neurons are four and the output variable is one. Thus, the number of weights to be optimized by HS is $8 \times 4 + 4$ =36 in total. The number of HS memory was 72 because doubled value of the number of optimization variables is generally used. Biases were not included in the optimization parameters because they could deteriorate the performance of HS algorithm by extending the dimension of objective function. Further, they can be corrected at the later BP learning process. The maximum number of improvisation was set to 100,000 and the optimization range was defined between -20 and 20. The interval for searching is 0.04 which is the total range divided by 1,000. Here, the parameters which are considered to have the greatest effect on the results and the reliability of the HS were selected: Harmony Memory Considering Rate (HMCR) and PAR (Pitch Adjusting Rate). Accordingly, the ANN-HS model simulations of HMCR and PAR varying from 0.1 to 0.9 at intervals of 0.2 were conducted and compared.

CHAPTER 4. RESULT AND DISCUSSION

4.1 Assessment of Model Stability

In this chapter, the stability of the two models are compared: one is the ANN model with BP training and the other is the ANN-HS model. Both models were run 50 times and the statistical analysis was conducted for the model results. Each of HMCR and PAR of HS has five different values varying from 0.1 to 0.9, so the case of ANN-HS includes $5 \times 5 = 25$ models. The correlation coefficient(r) and index of agreement(I_a) between model output values and target values in the validation data were used to evaluate the performance of the models. The statistical indices used in the assessment is the average, standard deviation and the maximum and minimum values of r and I_a. The average indicates that the higher the value, the average predictive power of the model is excellent and vice versa. The low standard deviation denotes the stability of the model, that is, the consistency of the output results is high. Maximum value indicates the highest value of r and I_a among 50 model results. The last is the minimum value of r and I_a. If this value is high, the model has high stability. Also it can be interpreted that the model can be used as a suitable model for predicting the future event.

In particular, the high minimum value and the low standard deviation of r and I_a may signify the precision and accuracy (Fig. 4.1). The smaller the standard deviation, we expect that the model would produce the results in a range without departing significantly form the average. This means that the model is precise.

Moreover, the high minimum value suggests that all the models show considerable accuracy. This variable also exhibits the worst result of the 50 model results. When this value is very low, it also indicates that it is not possible to use the model as a predictive model.

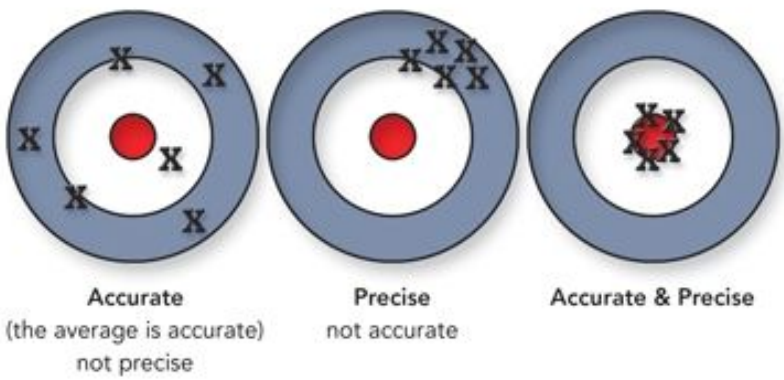

Figure 4.1 Accuracy and precision

The correlation coefficient and index of agreement between output variables of models and target variables are given in Eq. (4.1) and Eq. (4.2), respectively.

$$r = \frac{\sum_{i=1}^{n}\left(e_i - \overline{e}\right)\left(m_i - \overline{m}\right)}{\sqrt{\sum_{i=1}^{n}\left(e_i - \overline{e}\right)^2 \sum_{i=1}^{n}\left(m_i - \overline{m}\right)^2}} \tag{4.1}$$

$$I_a = 1 - \frac{\sum_{i=1}^{n}\left(e_i - m_i\right)^2}{\sum_{i=1}^{n}\left[\left|e_i - \overline{m}\right| + \left|m_i - \overline{m}\right|\right]^2} \tag{4.2}$$

where e_i and m_i denote the estimated and the measured stability number, $\overline{e}$

and $\bar{m}$ are the average of estimated and measured stability number. As r and I_a are close to one, the predicted set agree well to the measured set.

Table 4.1 and 4.2 show the statistics described above. In the tables, the highest value of average, the lowest value of standard deviation, the highest value of maximum and the highest value of minimum are colored in blue and the second best value colored in orange. In addition, the bold-faced fonts indicate the values that are relatively close to the blue-colored ones and the rank is denoted at the right upper part of its value.

For example, the minimum r value of ANN-HS with HMCR=0.9 and PAR=0.1 is 0.7976. This means that the model with such condition was run 50 times and the lowest correlation coefficient between the output data of those 50 models and target data is estimated as 0.7976 whereas the existing ANN model gives -0.0955.

Table 4.1 Statistical result analysis, Correlation coefficient (*r*)

Average of *r*						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.8423	0.8828	0.8289	0.8092	0.8585
	0.3	**0.8964**	0.869	0.8655	0.8747	0.869
	0.5	0.8709	0.8837	0.8833	0.8861	0.8873
	0.7	0.8427	0.8853	**0.9027**②	**0.8953**③	**0.8924**⑤
	0.9	**0.9032**①	**0.8939**④	0.87	0.8842	0.8475
ANN	-	0.7622				
Standard Deviation of *r*						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.1959	0.1272	0.2495	0.2465	0.1516
	0.3	0.1278	0.1664	0.1599	0.1789	0.1877
	0.5	0.1777	0.1028	0.1004	**0.0831**	0.13
	0.7	0.2020	0.1242	**0.0381**①	**0.0526**④	**0.0579**⑤
	0.9	**0.0397**②	0.1043	0.1539	**0.0496**③	0.1757
ANN	-	0.3140				
Maximum Value of *r*						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.9571	**0.9709**⑤	0.961	**0.9733**①	0.9636
	0.3	0.9590	0.9672	**0.9701**	**0.9719**③	0.96
	0.5	0.9614	0.9542	0.9606	0.9572	0.9679
	0.7	0.9676	**0.973**②	0.9594	0.9669	**0.9697**⑤
	0.9	**0.9712**④	**0.9702**	0.9717	**0.9700**	0.9595
ANN	-	0.9713				
Minimum Value of *r*						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.0216	0.2139	-0.1017	0.0313	0.1399
	0.3	0.0606	0.0178	0.0693	-0.0739	-0.0956
	0.5	-0.0900	0.3195	0.3264	0.4094	0.0649
	0.7	0.0327	0.0986	**0.7951**②	**0.7325**③	**0.6456**⑤
	0.9	**0.7976**①	0.2515	0.0698	**0.6960**④	0.1278
ANN	-	-0.0955				

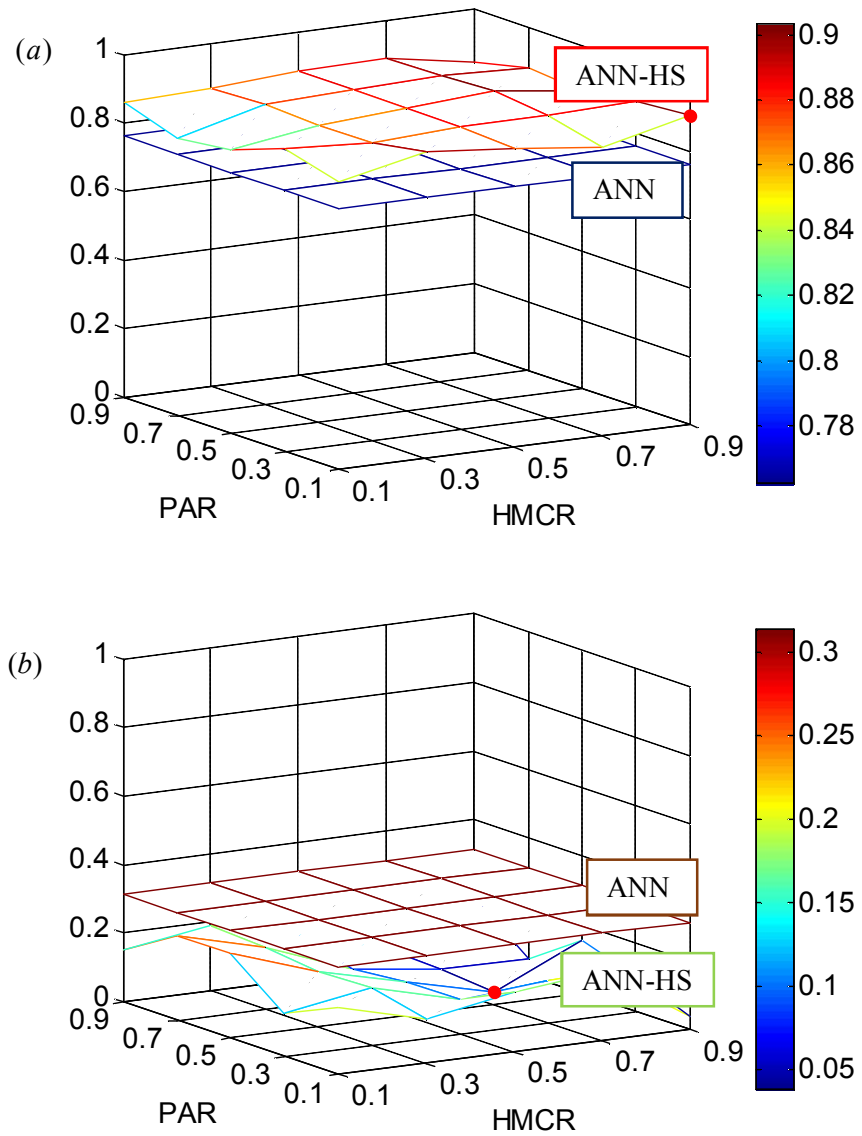

Figure 4.2 (a) Average and (b) standard deviation of correlation coefficient

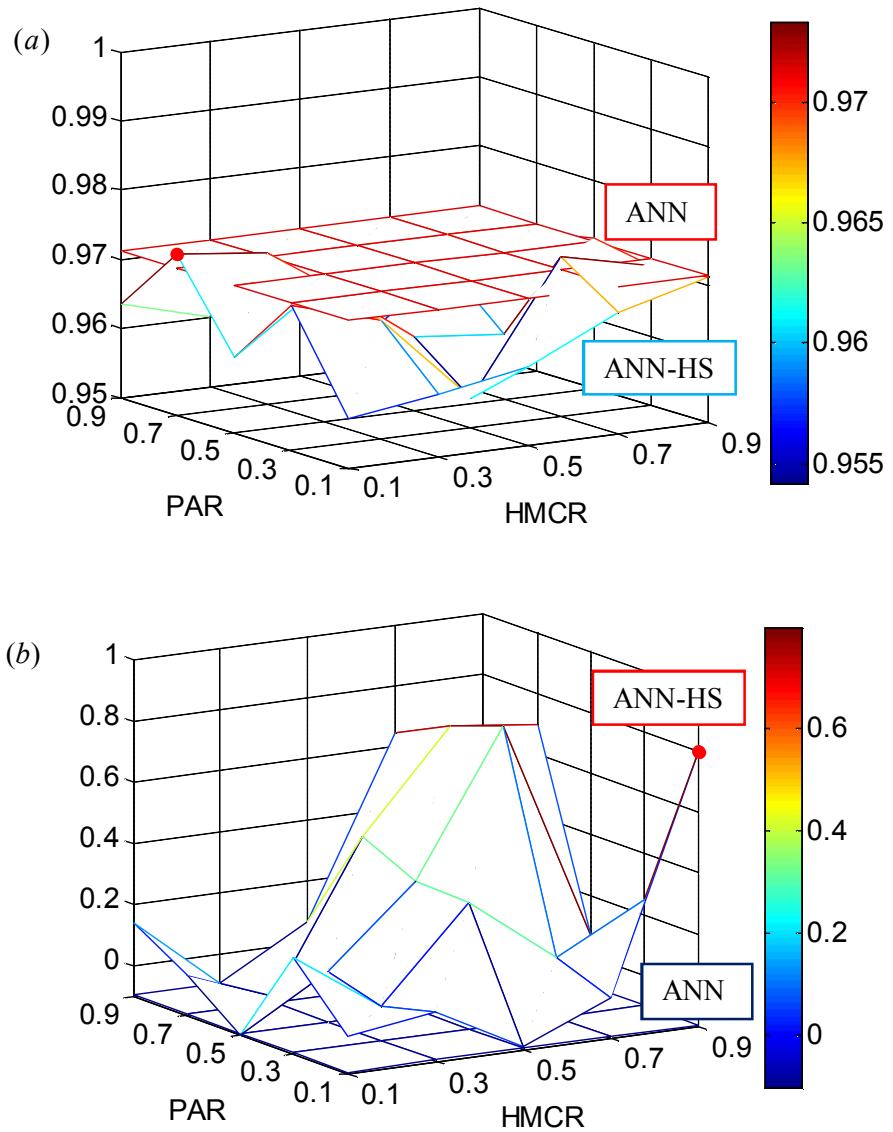

Figure 4.3 (a) Maximum and (b) minimum of correlation coefficient

Table 4.2 Statistical result analysis, Index of agreement (I_a)

Average of I_a						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.8853	0.9255	0.8724	0.8428	0.9047
	0.3	0.9343	0.9098	0.9125	0.9136	0.9119
	0.5	0.9133	0.9292	0.9292	0.9339	0.9285
	0.7	0.8813	0.929	**0.9482**①	**0.9435**③	**0.9402**④
	0.9	**0.9481**②	0.9351	0.9127	0.937⑤	0.8918
ANN	-	0.8042				
Standard Deviation of I_a						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.2053	0.1204	0.2445	0.2774	0.1583
	0.3	0.1374	0.1751	0.155	0.1832	0.1889
	0.5	0.1784	0.1006	0.1038	0.0728	0.1378
	0.7	0.2243	0.1301	**0.0213**①	**0.0305**③	**0.042**⑤
	0.9	**0.0226**②	0.1101	0.1684	**0.0305**④	0.2001
ANN	-	0.3174				
Maximum Value of I_a						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0.9781	**0.9853**	**0.98**	**0.9865**①	**0.9812**
	0.3	0.979	**0.9833**	**0.9849**	**0.9858**③	0.9794
	0.5	**0.9803**	0.9766	0.9797	0.9782	**0.9835**
	0.7	**0.9836**	**0.9864**②	0.979	**0.9831**	**0.9847**
	0.9	**0.9854**⑤	**0.9849**	**0.9857**④	**0.9848**	0.9795
ANN	-	0.9854				
Minimum Value of I_a						
ANN-HS	**HMCR\PAR**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
	0.1	0	0.2164	0	0	0.0015
	0.3	0.0017	0	0.0289	0.0036	0.0024
	0.5	0.0033	0.3186	0.2538	0.4684	0.0034
	0.7	0.0012	0.0512	**0.8889**①	**0.8524**③	0.7099⑤
	0.9	**0.8845**②	0.1955	0.0125	**0.8012**④	0.0047
ANN	-	0.0004				

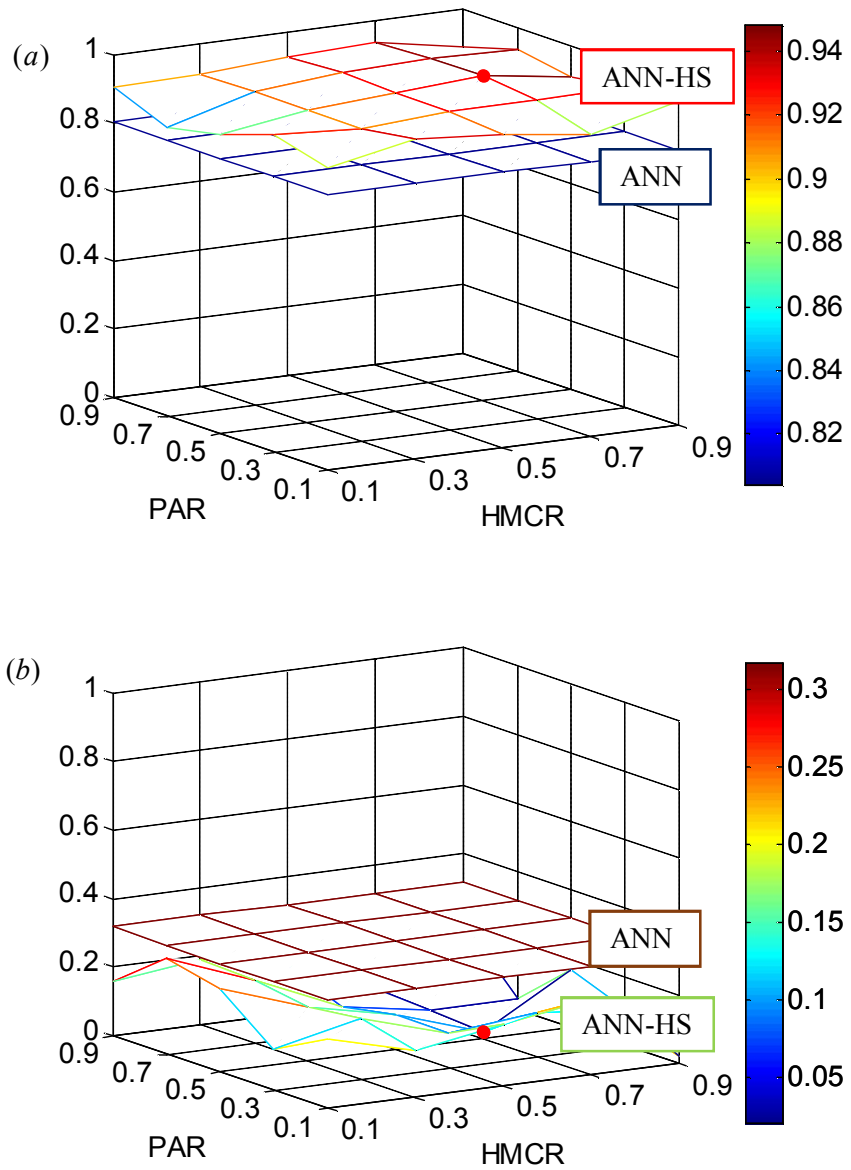

Figure 4.4 (a) Average and (b) standard deviation of index of agreement

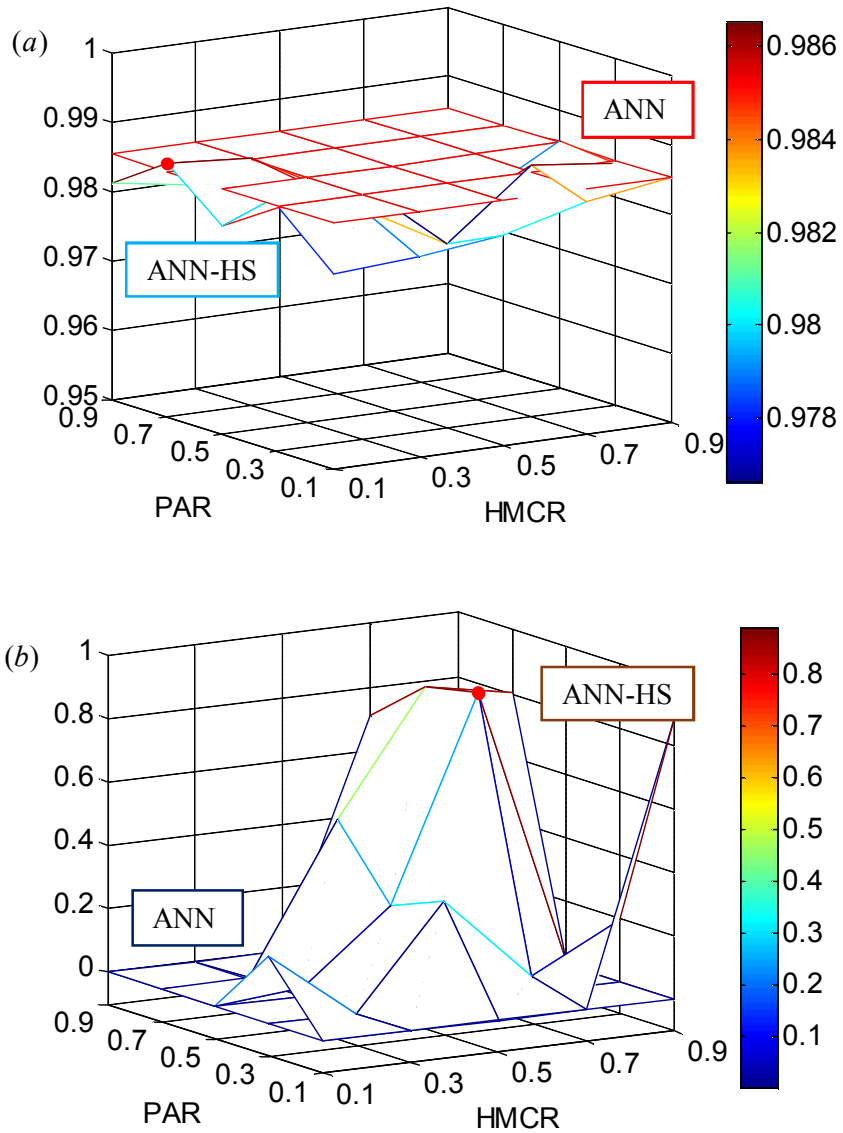

Figure 4.5 (a) Maximum and (b) minimum of index of agreement

As a result, the performance of general ANN and ANN-HS is compared by two different criteria, *r* and I_a. Although somewhat different trend is seen, both shows that ANN-HS is better than the general ANN model. In the Table 4. 1, the average value of *r* of ANN is 0.7622 while ANN-HS with HMCR=0.9 and PAR=0.1 has much higher value, 0.9032. Even though the optimal values of HMCR and PAR become different, the maximum average value of I_a of ANN-HS, 0.9482 is again much higher than that of ANNs, 0.8042. The standard deviation of *r* was 0.314 for the case of ANN, and the smallest value of *r* was 0.0381 for ANN-HS model with HMCR=0.7 and PAR=0.5. This implies that the initial value setting process of the ANN with HS increases the stability of the ANN model and finds near-global weights. In other words, HS is such a useful tool for setting the value of the initial weights for ANN. In the case of comparison of maximum values of *r* and I_a, the result of ANN-HS was only slightly higher than that of ANN. On the other hand, the minimum value of *r* and I_a of ANN-HS model were much higher ANN-HS model than that of ANN model. To be specific, ANN model gave -0.0955 of correlation coefficient and it signifies why it cannot be used as a predictive model.

From this analysis, we can propose ANN-HS model as an alternative to the general ANN model overcoming the limitation in the existing model. Furthermore, by comparing *r* and I_a of ANN-HS results having different HMCR and PAR, the optimal values of parameters could be suggested. However, the conclusions made by *r* and I_a are not exactly the same. The overall tendency is that I_a slightly overestimate than I_a values when it is higher than 0.9 or so. On the contrary, the value of *r* was somewhat overestimates than I_a if I_a is lower than the 0.1.

First, looking at the Table 4.1, all the results in the case for HMCR=0.9 and PAR=0.1 was evaluated to be good. On the other hands, in the Table 4. 2, we can see that the best combination of HMCR and PAR is 0.7 and 0.5.

The two cases which had been evaluated as outstanding in each evaluation criterion would be examined by the different criterion case. First, the case of HMCR=0.9, PAR=0.1 which received a good rating in the criterion *r* was also highlighted in the bold face in the criterion I_a, which means the prediction ability is relatively high. Especially, comparing it with the best case with HMCR=0.7 and PAR=0.5, there were only exiguous differences in average, standard deviation, maximum and minimum. Also, the case of HMCR=0.7 and PAR=0.5 which is the best case under the criterion I_a shows small dissidence with the case of HMCR=0.7 and PAR=0.5 in all aspects.

As a result, both cases of HMCR=0.9, PAR=0.1 and HMCR=0.7, PAR=0.5 were evaluated to be excellent in the optimization. The two cases always showed either best or second best performance in the criteria of average, standard deviation and minimum both in *r* and I_a. The result can be confirmed by Geem (2009) which suggests the typical range of HMCR=0.7~0.95 and PAR=0.1~0.5. Although the two cases did not always give the highest value of *r* and I_a, the lower value of maximum *r* does not mean that the ability and reliability of HS as a predictive model are low. Thus, we can conclude that the two models produce reliable, accurate and robust results.

4.2 Aspect of Transition of Weights

There are two major components in metaheuristic algorithms: diversification and intensification(Geem, 2009). These two components seem to be contradicting each other, but balancing their combination is crucial and important to the success of metaheuristic algorithm.

Appropriate diversification helps the algorithm to search for an optimal point in the solution space exploring as many locations and regions as possible in an effective way. If the diversification is too strong, it may wander many locations in a stochastic manner and the convergence of the algorithm becomes slow. On the contrary, if the diversification is too weak, the solutions are biased and trapped in local optima, or even produce meaningless solutions since the solution space to be explored is so limited.

On the other hand, proper intensification aims to exploit the history and experience obtained during search process. It intends to ensure to speed up the convergence by reducing the randomness and limiting diversification. If the intensification is too strong, it could lead the premature convergence, resulting in biased local optima or meaningless solutions since the domain of the search space is limited; if the intensification is too weak, it slows down the convergence of the algorithm

In the HS algorithm, diversification is controlled by the pitch adjustment and randomization. In addition, the intensification is represented by the harmony memory considering rate. Therefore, in this section, the results of neural networks for training dataset of four different cases would be compared and examined: (1)

low HMCR and low PAR, (2) low HMCR and high PAR, (3) high HMCR and low PAR, (4) high HMCR and high PAR. High value of HMCR means strong intensification and high value of PAR indicates strong diversification. The optimization process of HS algorithm regarding the weights of neural network for each case of parameter combination is illustrated in the following graphs (Figure 4.6 ~ 4.9). It is observed that a low memory considering rate selects only a few best harmonies and the model converges very slowly and increases the randomness of its accuracy whereas a high pitch-adjusting rate causes the solution to scatter around some potential optima with high randomness. One of the fifty ANN-HS model's outcomes demonstrated with *r* and I_a for each case and shown in Table 4.3.

Table 4.3 *r* and I_a between the predicted and target data

Case		**1**	**2**	**3**	**4**
HMCR/PAR		**0.1/0.1**	**0.1/0.9**	**0.9/0.1**	**0.9/0.9**
Initial weights	***r***	0.4662	-0.4838	0.3289	-0.0402
	I_a	0.0352	0.0123	0.0355	0.0191
After HS	***r***	-0.0898	0.1473	0.5772	0.9319
	I_a	0.2359	0.3519	0.9166	0.9581
After HS and BP	***r***	0.9719	0.9541	0.9896	0.9863
	I_a	0.9856	0.9761	0.9947	0.9930

The correlation coefficient and index of agreement of the model output calculated with randomly chosen initial weights are 0.4622 and 0.0362 as illustrated in the first graph in Figure 4.6 for the case (1). In figures 4.6 to 4.9, each scatter plot indicates the relationship between calculated output values and the observed stability number using (a) initial weights, (b) optimized weights after HS and (c) further trained weights after BP. The first graph shows that the model output ranges from -70 to 70 and are not related to the observed or target variables. The second graph is the result of neural networks after conducting HS optimization. Although the range of outcomes is reduced to -5 ~ 5, most of output values were sorted and converged to two different values. Furthermore, the second graph reveals the characteristic of global optimization; it does not make the fine adjustments. The third graph is the scatter plot of observed data. Output values after back propagation training based on initial weights based on the HS optimization results. It shows the advantage of gradient descent method, fine tuning.

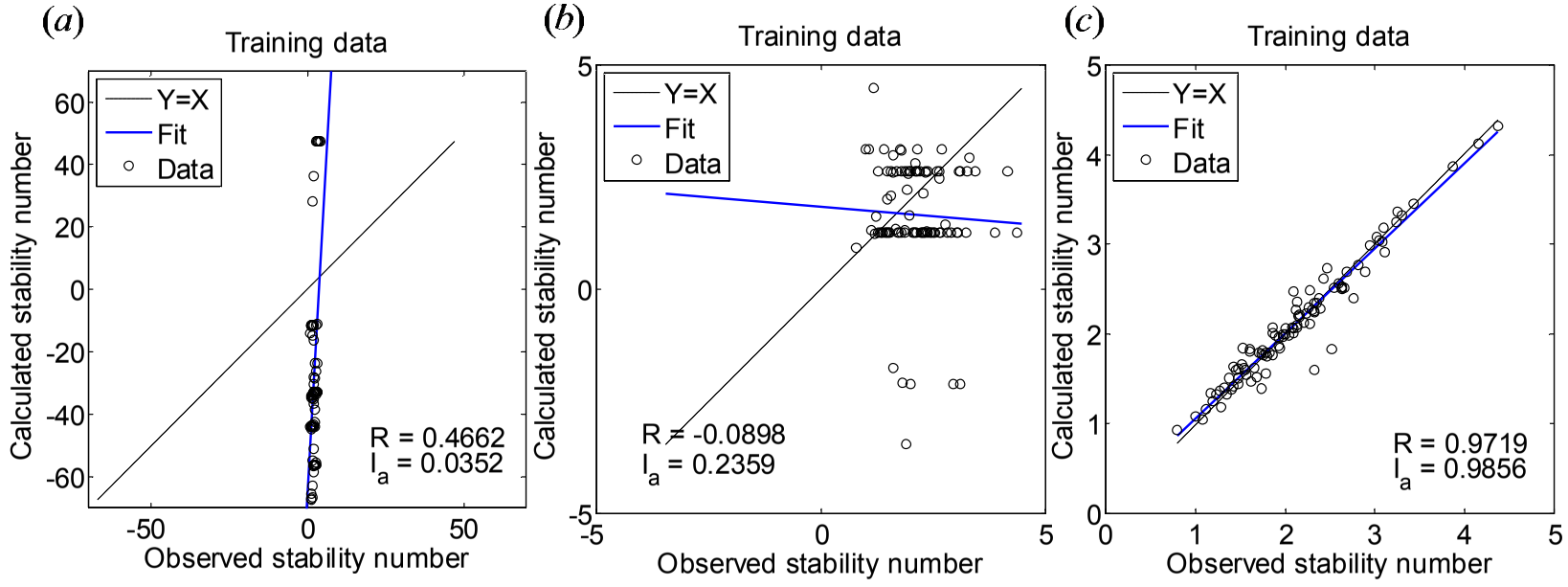

Figure 4.6 Scatter plot of case (1): observed and calculated based on (a) initial weights, (b) after HS and (c) after BP

Next, the second case is the ANN-HS model with HMCR=0.1, PAR=0.9. The diversification is very strong and the intensification is weak compared to other models; the randomness owing to those two parameter setting makes the algorithm converge slowly. The scatter plots in Figure 4.7 seem to be quite analogous to those of Figure 4.6. The second graph also illustrates the failure of fine adjustment with HS optimization converging into only two values. This phenomenon is considered to be caused by the small value of HMCR; even if it found the near-optimal harmony memories(HMs), it kept searching new HMs. Compared with the first case, a quite different value of PAR was used but the trend of scatter plot was almost the same. Therefore, this may lead us to the conclusion that HMCR has more influence on the HS optimization ability than the PAR.

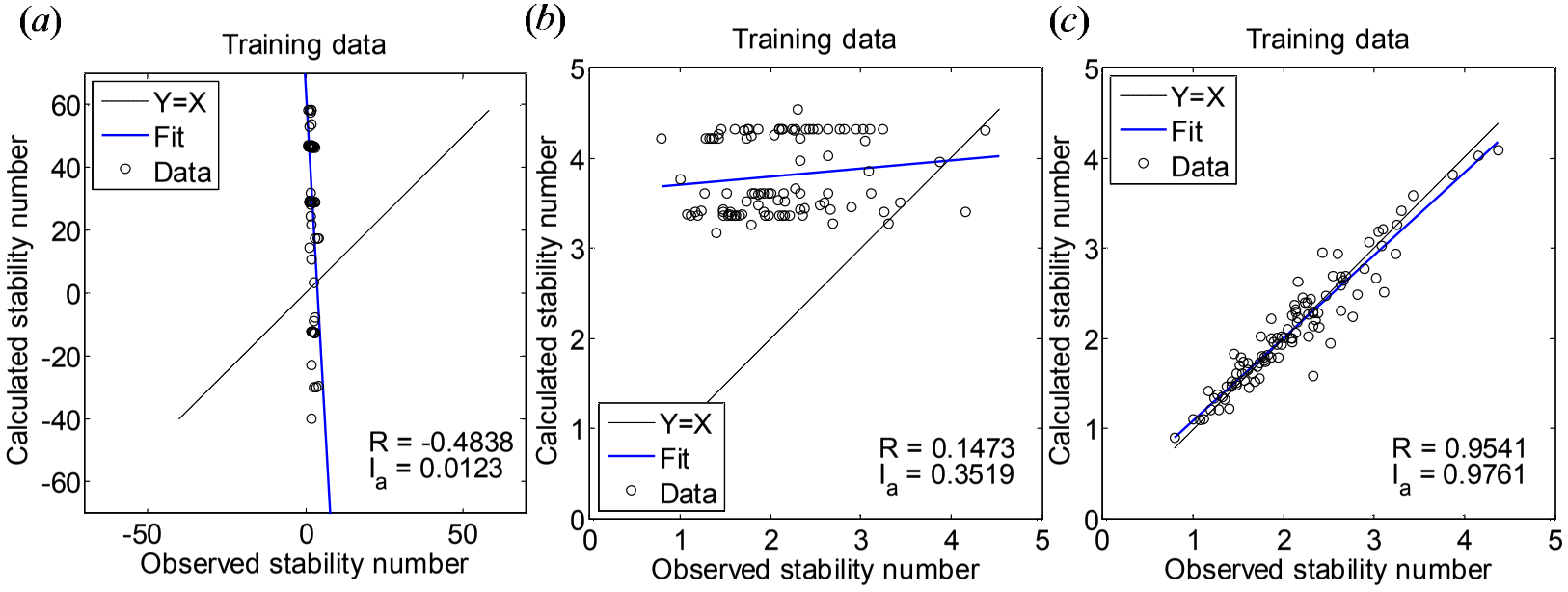

Figure 4.7 Scatter plot of case (2): observed and calculated based on (a) initial weights, (b) after HS and (c) after BP

In the third case, both parameters have high values: HMCR=0.9 and PAR=0.9. In the Figure 4.8(a), we can see that the calculated model data have no clear relationship with observed stability number. After the optimization with HS, although the index of agreement between the estimated and observed values is somewhat low as 0.5772, the correlation coefficient appears very high as 0.9319. In particular, unlike the previous two cases, the optimized results after HS become quite accurate. The only difference from the case (1) was the value of HMCR compare to the case (1). Therefore, the HMCR can be considered as an important factor for the performance of HS. Finally, this case gives 0.9863 of correlation coefficient after further training process with back propagation algorithm.

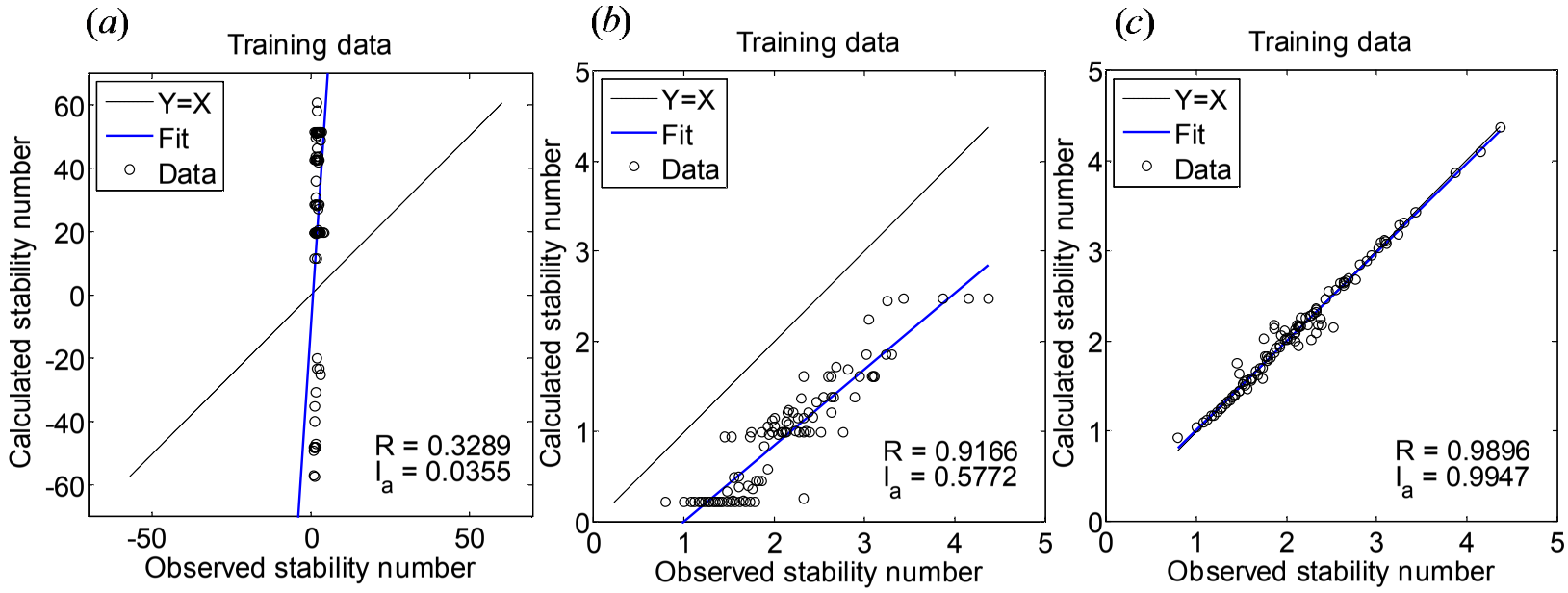

Figure 4.8 Scatter plot of case (3): observed and calculated based on (a) initial weights, (b) after HS and (c) after BP

The last case has 0.9 of HMCR and 0.1 of PAR. In this case, we can observe that the performance of the model using only optimization of HS is not so bad compared that of the model using both optimization of HS and training process of back propagation. Just like one of the conclusion drawn from the comparison of case (1) and case (2), the performance of HS optimization does not considerably differ between case (3) and case (4) which have the same value of HMCR but different values of PAR. In addition, a comparison between the first two cases and the last two cases shows that a large value of HMCR enhances the predicting ability of the model, especially when only the optimization of HS is used. However, a large value of HMCR not always guarantees the accurate result in HS algorithm. For example, we can see that the case of HMCR=0.9 and PAR=0.3 has $r_{\min}$ = 0.2515, $I_{a,\min}$ = 0.1955 and $\sigma_r = 0.1041$, $\sigma_{I_a} = 0.1101$, showing that the model gives inaccurate output. Therefore, the use of the parameters of HMCR=0.9 and PAR=0.1 is recommended.

Generally, we can find that the final result of ANN-HS was improved by HS optimization. Also, the final result of ANN-HS is dependent on the accurate HS optimization result; if the correlation coefficient of calculated and observed values after HS optimization is high, it is also high when the weights are further trained by back propagation training, vice versa.

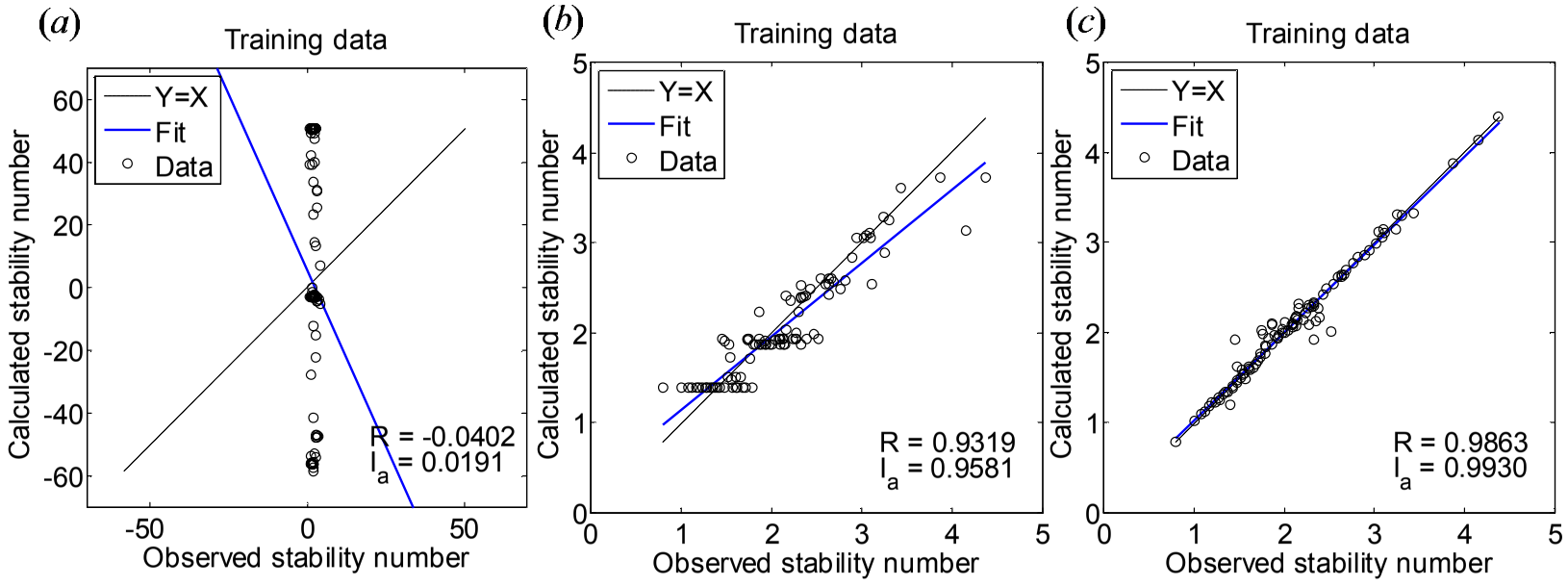

Figure 4.9 Scatter plot of case (4): observed and calculated based on (a) initial weights, (b) after HS and (c) after BP

4.3 Computation Time

Average and standard deviation of computation time of HS and BP are measured to give information to potential users(Table 4.4 ~ Table 4.7). The average computation time of HS algorithm is 285.51 s and that of BP after the HS optimization is 100.34 s. However, the average computation time of BP in general ANN model case was 68.57 s. In addition, the average of standard deviation of HS computation time is 6.82 s and that of BP is 94.96. Thus, the average computation time of ANN-HS model is 385.85 s and that of ANN model is 68.57 s; the former one was approximately 5 times to the latter one, however the absolute difference of those two may be considered to be negligible to the users.

Also, we could observe that there are no significant differences among the model results from various values of HMCR and PAR. This is because the maximum number of improvisation was set to 100,000. In the same manner, the maximum number of training, maximum epoch, of the general ANN model with only BP training algorithm was also set to 50,000. However, there were other stopping criteria such as maximum value of μ and maximum validation checks. This led to relatively large standard deviation of the average time spent on conducting ANN training.

Table 4.4 Average of HS computation time

Average of HS computation time (s)						
	HMCR\PAR	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
ANN-HS	**0.1**	284.66	285.01	285.31	285.08	284.91
	0.3	285.14	284.33	284.57	285.28	285.31
	0.5	285.59	285.67	286.16	286.21	286.04
	0.7	286.10	286.19	285.59	285.23	285.70
	0.9	285.70	285.76	285.83	286.42	285.91

Table 4.5 Standard deviation of HS computation time

Standard deviation of HS computation time (s)						
	HMCR\PAR	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
ANN-HS	**0.1**	6.28	6.03	6.60	7.04	6.38
	0.3	6.29	5.61	5.56	6.27	6.84
	0.5	7.09	7.42	6.86	7.39	8.00
	0.7	8.58	9.08	7.80	7.50	6.84
	0.9	6.50	6.66	6.00	6.27	6.54

Table 4.6 Average of BP computation time

Average of BP computation time (s)						
	HMCR\PAR	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
ANN-HS	**0.1**	91.28	93.28	94.02	95.53	92.57
	0.3	97.19	108.42	96.10	108.41	102.06
	0.5	86.97	100.08	103.57	104.79	107.55
	0.7	109.91	87.47	102.92	110.80	96.85
	0.9	102.75	105.71	107.72	103.94	98.61
ANN	**-**	68.57				

Table 4.7 Standard deviation of BP computation time

Standard deviation of BP computation time (s)						
	HMCR\PAR	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
ANN-HS	**0.1**	57.08	59.00	59.48	60.25	61.39
	0.3	55.87	54.53	58.72	55.16	56.52
	0.5	59.74	57.94	56.01	58.06	56.37
	0.7	53.36	62.72	55.17	54.43	59.60
	0.9	57.59	58.16	54.06	58.57	62.26
ANN	**-**	94.96				

CHAPTER 5. CONCLUSIONS

In this study, a ANN-HS model is developed to predict the stability number of breakwater armor stones based on the experimental data of Van der Meer (1988). It is composed of two steps of optimization. Firstly, the harmony search optimization algorithm is used to set initial weights of neural networks. Then, the back propagation training further modifies the weights and biases of the network in the direction of minimizing the errors between model output and target values.

To compare the reliability of the ANN model using only BP training and the ANN-HS model, both models were run 50 times and the statistical analysis was conducted for the model results. Each of HMCR and PAR of HS has five different values varying from 0.1 to 0.9 at intervals of 0.2. The correlation coefficient (r) and index of agreement (I_a) between model output values and target values in the validation data were used to evaluate the performance of the models.

As a result, the prediction abilities of 50 ANN models were quite different from each other. The average of r is 0.7622 and the standard deviation is 0.3140. On the contrary, the average value of r was 0.9032 and the standard deviation is 0.0391 in ANN-HS models with HMCR=0.9 and PAR=0.1. This result indicates that both the prediction accuracy and stability of ANN-HS model are much higher than those of ANN model.

In addition, analyzing the results of ANN-HS models, we could conclude that the cases of HMCR=0.9, PAR=0.1 and HMCR=0.7, PAR=0.5 showed best performance and a sense of reliability. Those two models give significantly

improved performance indicating low level of standard deviation and high level of minimum both in r and I_a . The result can be confirmed by Geem (2009) who suggests the typical range of HMCR=0.7~0.95 and PAR=0.1~0.5.

Furthermore, the results of ANN model and ANN-HS model could be assessed whether they are different or not by using t-test. Besides, the performance of ANN-HS model can be compared to that of GA-ANN model to evaluate their relative ability to find the initial weights of ANN.

Ultimately, we hope to assist engineers and researches not only engaged in coastal engineering but also in various fields of study which need a robust and reliable alternative prediction model by proposing ANN-HS model through this study.

REFERENCES

Balas, C.E., Koc, M.L., Tür, R.. (2010) Artificial neural networks based on principal component analysis, fuzzy systems and fuzzy neural networks for preliminary design of rubble mound breakwaters. *Applied Ocean Research* 32, 425-433.

Boucher, M. A., Perreault, L. and Anctil, F. (2009) Tools for the assessment of hydrological ensemble forecasts obtained by neural networks. *Journal of Hydroinformatics*, 11(3-4), 297–307.

Browne, M., Castelle, B., Strauss, D., Tomlinson, R., Blumenstein, M. and Lane, C. (2007) Near-shore swell estimation from a global wind–wave model: spectral process, linear and artificial neural network models. *Coastal Engineering* 54, 445–460.

Chang, Y.-T., Lin, J., Shieh, J.-S.,Abbod, M.F. (2012) Optimization the Initial Weights of Artificial Neural Networks via Genetic Algorithm Applied to Hip Bone Fracture Prediction. *Advances in Fuzzy Systems* 2012, 9.

Cox, D.T., Tissot, P., Michaud, P. (2002) Water level observations and short-term predictions including meteorological events for entrance of Galveston Bay, Texas. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 128, 21–29.

Erdik, T. (2009) Fuzzy logic approach to conventional rubble mound structures

design. *Expert Systems with Application* 36, 4162-4170.

Etemad-Shahidi, A., Bonakdar, L. (2009) Design of rubble-mound breakwaters using M5' machine learning method. *Applied Ocean Research* 31, 197-201.

Geem, Z.W. (2009) *Music-Inspired Harmony Search Algorithm*. Springer

Geem, Z.W., Kim, J.H., Loganathan, G.V. (2001) A new heuristic optimization algorithm: Harmony search. *Simulation* 76, 60–68

Geman, S., Bienenstock, E., Doursat, R. (1992) Neural networks and the bias/variance dilemma. *Neural Computation* 4(1), 1-58.

Haldar, A., and Mahadevan, S. (2000) *Reliability Assessment Using Stochastic Finite Element Analysis*. Wiley, New York.

Hoel, P.G. (1962) *Introduction to Mathematical Statistics* (3rd edition). Wiley, New York.

Hudson, R.Y. (1959) Laboratory investigation of rubble-mound breakwaters. *Journal of Waterways and Harbors Division* 85(WW3), 93-121.

Kim, D.H., Park, W.S. (2005) Neural network for design and reliability analysis of rubble mound breakwaters. *Ocean Engineering* 32. 1332-1349.

Kim, S.E. (2014) Improving the generalization accuracy of ANN modeling using factor analysis and cluster analysis: its application to streamflow and water quality predictions. *Ph.D. dissertation,* The Graduate School of Civil and Environmental Engineering of Seoul National University

Kolen, J.F. and Pollack, J.B. (1990) *Back propagation is sensitive to initial conditions*, pp. 860-867, Morgan Kaufmann Publishers Inc., Denver,

Colorado, United States.

Krogh, A. and Vedelsby, J. (1995) Neural network ensembles, cross validation and active learning, *Advances in Neural Information Processing System,* 7, 231–238.

Lee, K.S. and Geem Z. W. (2004) A new structural optimization method based on the harmony search algorithm. *Computers and Structures 82*, 781–798

Mase, H., Sakamoto, M., Sakai, T. (1995) Neural network for stability analysis of rubble-mound breakwaters. *J. Waterway, Port, Coastal and Ocean Eng.* 121(6), 294-299.

McCulloch, W.S. and Pitts, W. (1990) A Logical Calculus of the Ideas Immanent in Nervous Activity (Reprinted from Bulletin of Mathematical Biophysics, Vol 5, Pg 115-133, 1943). *Bulletin of Mathematical Biology* 52(1-2), 99-115.

Montana, D.J. (1995) Neural network weight selection using genetic algorithms. In: Goonatilake S, Khebbal S. editors. *Intelligent hybrid systems*. Chichester: Wiley. 85-104.

Mulia, I. E., Tay, H., Roopsekhar, K. and Tkalich, P. (2013) Hybrid ANN-GA model for predicting turbidity and chlorophyll-a concentration. *Journal of Hydroenvironmental Reasearch,* 7, 279-299.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (1992) *Numerical recipes in Fortran*, 2nd ed., Cambridge University Press, Cambridge, UK.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) Learning

representations by back-propagating Errors. *Nature* 323(6088), 533-536.

Smith, W.G., Kobayashi, N., Kaku, S. (1992) Profile changes of rock slopes by irregular waves. *Proceedings of 23rd International Conference on Coastal Engineering*. American Society of Civil Engineers, pp. 1559-1572.

Suh, K.-D., Yoo, D.H. (2003) Comparison of accuracy of stability formulas for breakwater armor stones. *Journal of Korean Society of Coastal and Ocean Engineers* 15(3), 186-189 (in Korean).

Tsai, C.P., Lee, T. (1999) Back-propagation neural network in tidal-level forecasting. *Journal of Waterway, Port, Coastal, and Ocean Engineering* 125, 195–202.

Van der Meer, J.W. (1987) Stability of breakwater armor layers − Design formulae. *Coastal Engineering* 11, 93-121.

Van der Meer, J.W. (1988) *Rock slopes and gravel beaches under wave attack*. Delft Hydraulics, Communication No. 396.

Van Gent, M.R.A., van den Boogaard, H.F.P., Pozueta, B. and Medina, J.R. (2007) Neural network modeling of wave overtopping at coastal structures. *Coastal Engineering* 54, 586–593.

Venkatesan, D., Kannan, K. and Saravanan, R. (2009) A genetic algorithm-based artificial neural network model for the optimization of machining processes, *Neural Computing and Applications,* 18 (2), 135–140.

Yam, Y.F. and Chow, T.W.S. (1995) Determining initial weights of feedforward neural networks based on least-squares method. N*eural Processing*

Letters 2(2), 13-17.

Yoo, D.H., Koo, S.K., Kim, I.H. (2001) Minimum weight of breakwater armor unit. *Proceedings of 1st Asian and Pacific Coastal Engineering Conference*, Dalian, China, pp. 605-612.

Yoon H.D., Cox D.T. and Kim M.K. (2013) Prediction of time-dependent sediment suspension in the surf zone using artificial neural network. *Coastal Engineering* 71, 78-86

Zamani, A., Azimian, A., Heemink, A. and Solomatine, D. (2009) Wave height prediction at the Caspian Sea using a data-driven model and ensemble-based data assimilation methods. *J. Hydroinf.* 11(2), 154–164.

국문초록

하모니 서치 알고리즘을 이용한 인공신경망 근사 전역 최적 초기 가중치 설정: 방파제 피복석에의 적용

서울대학교 대학원

건설환경공학부

이 안 지

인공신경망은 다양한 분야에서 예보 및 예측을 위한 유용한 도구로 자리매김하고 있다. 다양한 연구 분야에 적용된 수 많은 사례에도 불구하고, 인공신경망은 여전히 일반화된 도구로 여겨지지 못하고 있다. 역전파 알고리즘은 주로 출력 변수와 타겟 변수의 평균 제곱근 오차(Root Mean Square Error, RMSE)로 정의되는 성능함수(performance function)를 최소로 하는 신경망의 가중치(weights)와 바이어스(bias)를 찾도록 하지만, 이는 기울기 하강 방법(Gradient descent method)을 사용하기 때문에 성능함수의 값을 국소 최솟값 (local minimum)에 머무르게 하며 초기 가중치와 바이어스에 따른 높은 민감도를 갖게 한다. 이러한 문제를 해결하기 위해 서로 다른 초기 가중치를 갖는 다수의 인공 신경망을 생성하는 몬테-카를로 시뮬레이션을 통해 전역 최솟값(global minimum)을 갖는 가중치와 바이어스를 탐색하는 방법이 상당수 제안되었지만, 이는 비효율적이고 시간이 상당히 많이 소요된다는 단점을 갖는다. 또한 이 시뮬레이션을 통

해 얻은 최적합 신경망이 전역 최솟값을 갖는다는 것을 보장할 수 없으며, 만약 이 신경망이 전역 최솟값을 갖는다 하더라도 이를 추후에 재현할 수 없다는 문제점이 있다.

본 연구에서는 1988년에 수행된 Van der Meer의 실험 자료를 바탕으로 방파제 사석의 안정수를 예측하는 인공신경망 모형을 구현하였다. 역전파 알고리즘의 근본적인 문제인 국소 최솟값 탐색을 해결하기 위해 전역 최솟값 탐색 방법 중 하나인 하모니 서치 최적화 알고리즘을 이용하였다. 먼저, HS 알고리즘을 통해 성능함수의 근사 전역 최소값을 찾는다. HS를 통해 최적화된 가중치 값들은 인공신경망의 초기 가중치로 사용되고 추후 역전파 알고리즘으로 학습된다. 역전파 학습 알고리즘으로는 기울기 하강 방법을 사용하여 가중치를 더욱 미세하게 수정한다.

역전파 학습 방법으로만 최적 가중치 값을 탐색한 인공신경망 모형(ANN)과 HS 알고리즘을 이용하여 초기 가중치를 설정한 후 역전파 학습을 수행한 인공신경망 모형(ANN-HS)의 신뢰도와 안정성을 평가하기 위해서 두 모형을 각 50번씩 수행한 결과를 통계적으로 분석하였다. ANN-HS 모형은 하모니 메모리 채택 비(Harmony memory considering ratio, HMCR)와 피치 조정비(Pitch adjusting ratio, PAR)는 각각 0.1 부터 0.2 간격으로 0.9 까지 변화시켜가며 수행하였다. 인공신경망 모형의 성능을 평가하기 위해 두 모형의 출력 값과 목표 값 사이의 상관계수(Correlation coefficient, r)와 일치도(Index of agreement, I_a)를 계산하였다. 결과적으로 HMCR=0.9와 PAR=0.1 인 경우, 그리고 HMCR=0.7 와 PAR=0.5 인 경우

의 ANN-HS 모형이 역전파 알고리즘으로만 학습한 ANN 모형보다 더욱 정확하고 일관된 예측 능력을 보였다.

keywords: 피복석, 인공신경망 모형, 하모니 서치 알고리즘, 안정수

학번: 2014-20546