An intelligent swarm based-wavelet neural network for affective mobile phone design

S.H. Ling[1], P.P. San[1], K.Y. Chan[2], F.H.F. Leung[3] and Y. Liu[4]

[1]*Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia*

[2]*Department of Electrical and Computer Engineering, Curtin University of Technology, WA, Australia*

[3]*Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong*

[4]*Institutes of Image and Graphics, School of Computer Science and Engineering, Sichuan University, China*

Abstract

In this paper, an intelligent swarm based-wavelet neural network for affective mobile designed is presented. The contribution on this paper is to develop a new intelligent particle swarm optimization (iPSO), where a fuzzy logic system developed based on human knowledge is proposed to determine the inertia weight for the swarm movement of PSO and the control parameter of a newly introduced cross-mutated operation. This iPSO will be used to optimize the parameters of wavelet neural network. Application on affective design of mobile phones is used to test the performance of the proposed iPSO and found that it is significantly better than that of the existing hybrid PSO methods in a statistical sense.

Keywords:
Affect mobile design, Fuzzy reasoning model, Particle swarm optimization, and Wavelet neural network.

1. Introduction

Recent research demonstrates that particle swarm optimization (PSO) is a more effective optimization method to optimize the parameters (weights) of neural network models [1, 2, 3, 4, 5, 6, 7, 8] and industrial applications

[9, 10]. Although reasonable solutions can generally be obtained by the PSO within a reasonable computational time, enhancement of the operations and the mechanisms of the PSO is essentially required in order to obtain better solutions. A commonly used enhancement approach is to integrate other optimization operations such as gradient descent, crossover and mutation operations into the PSO.

Juang [11] proposed a hybrid PSO algorithm namely HGAPSO which use evolutionary operations including crossover, mutation and reproduction to control swarm movement, and also a hybrid PSO namely HPSOM was proposed [12] by integrating the PSO with mutation operation. Both HGAPSO and HPSOM inject random components into particles using mutation, but the mutating space used in both approaches is fixed throughout the search. Although premature convergence is more likely to be avoided, the approach can be further improved by varying the mutating space with respect to the searching progress of the PSO. A hybrid PSO with wavelet mutation operation (HPSOWM) was proposed in [13], of which the mutating space varied based on the wavelet theory. By introducing the wavelet mutation, the performance in terms of solution quality and stability can be improved. Although the approaches provide a balance between the global exploration and local exploitation, they are not appropriate to assume the searching progresses of the PSO are linear or wavelet characteristics, and it is also impractical and almost impossible to mathematically model the searching progress of the PSO, in order to determine the appropriate inertia weight for searching the optimum.

In this paper, an intelligent PSO namely iPSO is proposed to optimize the parameter of variable translation wavelet neural network (VTWNN) [1, 3] and will be applied to affective product design. iPSO is proposed by introducing two new operations, namely fuzzy inertia weight and cross-mutated (CM) operation. The fuzzy inertia weight is determined based on a fuzzy inference system which consists of a set of linguistic rules in representing the searching characteristics of the PSO. By dynamically changing the fuzzy inertia weight, the dynamic of the swarm can be varied with respect to the searching progress of the PSO. Hence, solutions with better qualities are more likely to be searched. The CM injects momentum to the swarm when the progress of the PSO is saturating, where the amount of momentum is controlled based on the fuzzy inference system. It intends to further avoid the PSO in searching the local optima.

VTWNN has been used to model the relationships between design vari-

ables and affective responses on mobile design application. Wavelets are used as transfer functions in the hidden layer of the VTWNN. The network parameters, i.e., the translation parameters of the wavelets, are variable depending on the network inputs. Thanks to the variable translation parameters, the VTWNN becomes an adaptive network capable of handling different input patterns and exhibits a better modeling performance.

In the development of a new mobile phone, basic functions such as operations of transmission and receiver must work satisfactorily. After these basic functions have been achieved, function operations in a higher level are required to be satisfied. For example, the mobile phone should be felt comfortable when hand-held by the customer. The buttons of a mobile phone should also be pressed easily, and the voices should be clearly heard by the receiver. After satisfying all those functional operations, optimization of affective responses is essentially required [14]. It is now evident in the mobile phone market that successful can transform its products from being merely functional items to lifestyle or fashion accessories [15]. Consequently, product designers are increasingly focusing on optimizing affective responses rather than solely optimizing their functional operations.

This paper is organized as follows. Section 2 presents the affective mobile design and its morphological matrix. The modeling with VTWNN is discussed in Section 3. The details of iPSO is presented in Section 4. Experimental study and analysis will be given in Section 5 to evaluate the performance of the proposed method. Finally, a conclusion will be drawn in Section 6.

2. Affective mobile design

In the highly competitive market of mobile phones, the product designers provide the consumers with various styles for different brands and different product series of mobile phones. To capture the trend, we have selected 32 recent mobile phones of various brands, including Nokia, Sony Ericsson and Motorola. Morphological analysis shown in Fig. 1 is used to extract representative elements of mobile phones as numerical data sets, in which both the shape profiles and the product components of the mobile phones are used. As shown in Fig. 1, nine representative design elements of the affective design for mobile phones are used, namely "top shape", "bottom shape", "side shape", "function button shape", "number button style", "length width ratio", "thickness", "layout" and "Border and frame", where those represen-

tative design elements are denoted as x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 and x_9 respectively. Those representative design elements were identified from the 32 mobile phone samples. The number 1-6 of the design matrix represents the type of design elements shown in Figure 1.

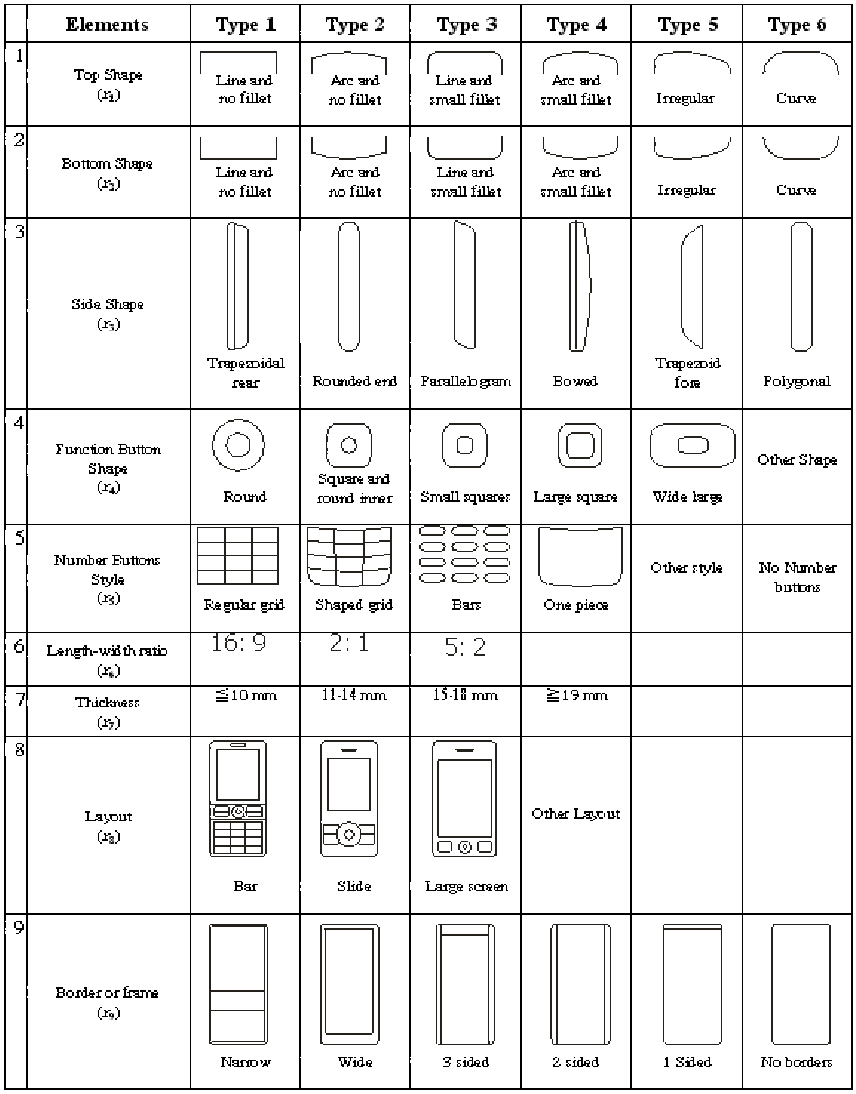


Figure 1: Morphological analysis on the 32 representative mobile phone samples.

Based on the experience of the affective designers, four most representa-

tive affective responses for mobile phone design, "simple - complex (S-C)" y_1, "unique - general (U-G)" y_2, "high-tech - classic (H-C)" y_3, and "handy - bulky (H-B)" y_4 [16], are collected from 14 image-word pairs for micro-electronic products, and they are used for evaluating the affective satisfaction of the mobile phones. A survey was conducted using an online questionnaire to study the appearance of mobile phones on y_1, y_2, y_3 and y_4, which is detailed in [17]. Fig. 2 shows the morphological matrix of the 32 mobile phones samples based on the 9 representative elements. Also, it shows the means of the affective responses S-C, U-G, H-C, and H-B with respect to 34 interviewers. Prior to optimizing the affective responses, the development of the affective models is essential in order to relate the representative design elements x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 and x_9 to one of the affective responses y_1, y_2, y_3 and y_4.

3. Modeling with variable translation wavelet neural network

A three-layer variable translation wavelet neural network [1, 3] shown in Fig. 3 is used to develop the affective models of mobile phones, as the approach is effective in modeling nonlinear relationships. In the neural network, wavelet functions are used as the transfer functions in the hidden layer. The translation parameters of wavelets are dependent on the network inputs. This wavelet neural network is tuned using the proposed iPSO. Its structure consists of an input layer in which the input vectors (consisting of nine designed elements x_1, x_2, x_3, ... x_9) are fed. The output layer generates the affective response (either y_1, y_2, y_3 or y_4), and a hidden layer is between the input layer and the output layer of the wavelet neural network. As wavelet functions are linking between the input and output layers, complex, interactive and nonlinear characteristics of affective response can be modelled effectively.

The first affective response, namely simple-complex y_1, governed by the wavelet neural network [3] is given as:

$$y_1(t) = \sum_{j=1}^{n_h} \psi_{j,b_j}(S_j) \cdot w_{1j} \tag{1}$$

where

$$S_j = \sum_{i=1}^{n_{in}} z_i(t) v_{ji} \tag{2}$$

Phone no.	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	S-C	U-G	H-C	H-B
1	3	3	1	3	2	2	3	1	1	1.85	3.62	2.97	2.56
2	3	3	2	2	1	2	2	1	2	2.59	3.44	3.15	2.79
3	6	6	1	1	5	2	4	1	2	2.88	2.76	3.21	3.32
4	4	4	3	1	6	2	2	2	2	2.41	2.65	2.88	2.59
5	3	4	3	4	6	2	2	2	2	2.06	2.85	2.53	2.47
6	3	3	1	5	6	1	3	2	1	2.71	2.41	2.15	3.18
7	1	1	2	4	6	3	4	2	6	3.26	2.53	2.47	3.18
8	1	1	1	2	6	2	2	2	6	2.79	2.74	2.50	2.71
9	3	4	6	1	6	2	3	2	2	2.91	2.65	2.85	3.12
10	4	4	3	6	4	3	2	1	2	2.65	2.82	3.00	2.15
11	2	2	6	5	6	2	4	2	2	2.76	2.62	2.47	3.18
12	2	2	6	3	6	2	3	2	4	2.71	2.56	2.41	3.38
13	6	6	6	4	6	2	3	2	2	2.09	2.76	2.85	2.71
14	4	4	2	6	6	1	2	3	3	2.21	2.09	2.09	1.94
15	4	3	6	1	6	2	3	2	2	2.44	2.82	2.71	3.09
16	3	3	6	5	6	2	3	2	1	2.62	2.15	2.35	2.94
17	3	3	2	6	6	1	2	3	1	2.12	2.53	2.35	3.03
18	2	4	6	5	2	3	1	1	5	2.50	3.38	2.97	2.59
19	3	3	1	4	5	3	3	1	1	2.41	3.00	3.00	3.03
20	4	4	6	5	1	3	2	1	1	2.68	3.68	3.53	3.06
21	4	4	1	1	2	3	2	1	2	2.88	3.35	3.29	3.12
22	6	4	3	1	4	3	2	1	2	2.88	2.94	2.97	2.97
23	3	3	6	2	3	2	3	1	6	3.12	3.38	3.15	3.56
24	5	5	1	4	3	3	2	1	1	2.50	2.85	3.24	2.62
25	4	4	6	1	6	2	3	2	2	2.44	3.21	3.06	3.09
26	3	6	5	1	6	3	3	2	2	2.68	2.97	2.85	3.32
27	1	1	5	1	6	2	2	2	4	2.65	2.79	2.79	2.91
28	3	3	4	1	6	1	2	3	2	2.00	1.91	1.91	2.53
29	4	4	2	1	6	1	2	2	2	2.41	2.47	2.21	2.56
30	4	4	4	5	2	2	3	1	2	3.26	3.15	2.82	3.03
31	3	3	1	6	6	2	3	4	2	3.38	2.79	2.76	3.18
32	3	3	1	1	6	2	3	2	1	2.32	2.62	2.56	3.50

Figure 2: Design matrix of 32 mobile phone samples.

and

$$\psi_{j,b_j}\left(S_j\right) = \frac{1}{\sqrt{j}} e^{\frac{-\left(\frac{S_j - b_j}{j}\right)^2}{2}} \left(1 - \left(\frac{S_j - f^j\left(S_j\right)}{j}\right)^2\right) \tag{3}$$

where

$$f^j(S_j) = 4 \times j \left(\frac{2}{1 + e^{-\kappa^j \times S_j}} - 1\right) \tag{4}$$

v_{ji}, i=1, 2, ..., n_{in}; j=1, 2, ..., n_h denotes the weight of the link between the i-th input node and the j-th hidden node, n_{in} and n_h represent the number of inputs and the number of hidden nodes respectively. Here, $n_{in} = 9$ and $n_h = 5$ are used. w_j denotes the weight of the link between the j-th hidden node and the output. The parameters of the wavelet neural network (v_{ji}, w_j and κ^j) are required to be optimized. Similarly, the other three affective responses, y_2, y_3 and y_4, can be governed by the formulation represented by (1)-(4). Here 55 parameters are required to be determined in the wavelet neural network.

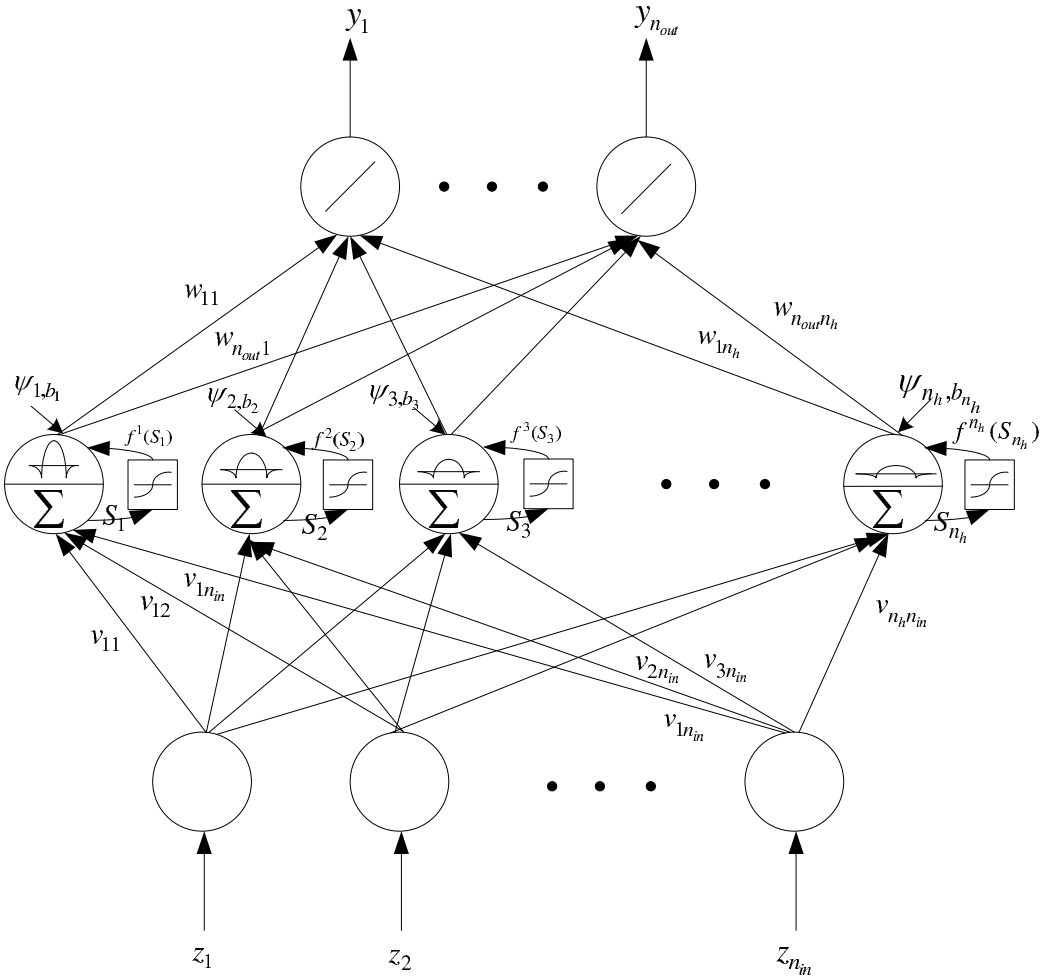


Figure 3: Structure of variable translation wavelet neural network.

4. Intelligent particle swarm optimization (iPSO)

Particle swarm optimization (PSO) models the social behavior of a swarm like bird flocking and fish schooling. The swarm is composed of a number of particles. Every particle traverses a search space for the best fitness value. PSO with inertia weight [18] and PSO with constriction factor [19] were reported to show improved searching ability over the standard PSO [20]. In [18], the inertia weight $\omega(t)$ provides a balance between the global exploration and local exploitation of the swarm. When $\omega(t)$ is linearly related to iteration time t, if the value of t/T (where T is total number of iteration) is smaller, more global exploration is done; if it is larger, more fine-tuning (local exploitation) is realized. However, a linear relation between $\omega(t)$ and t may not be so appropriate because the search progress of the swarm is not a linear movement. Thus, we propose a nonlinear inertia weight $\tilde{\omega}(t)$ to enhance the searching performance. The value of $\tilde{\omega}(t)$ is evaluated by a 2-input fuzzy inference system as one of its 2 outputs. (The other output is the control parameter $\beta(t)$ of the CM operation that will be discussed in the later sub-section.)

The pseudo code for iPSO is given in Algorithm 4.1. A fuzzy inertia weight $\tilde{\omega}(t)$ is first proposed to improve the searching quality. A cross-mutated (CM) operation is also added to tackle the limitation of standard PSO [18, 19, 20] being easy to trap in some local minima.

Under Algorithm 4.1, $X(t)$ denotes a swarm at the t-th iteration. Each particle $\mathbf{x}^i(t) \in X(t)$ contains κ elements $x_j^i(t) \in \mathbf{x}^i(t)$, where $i = 1, 2, \ldots, \gamma$ and $j = 1, 2, \ldots, \kappa$; γ denotes the number of particles in the swarm and κ is the dimension of a particle. At the beginning, the swarm particles are initialized and then evaluated by a defined fitness function $f(\mathbf{x}^i(t))$. The current generation number t is initialized to 0. The job of iPSO is to minimize the fitness value through an iterative process.

The evolution realised by iPSO is governed by the velocity (flight speed) of the particles in the search space. The velocity $v_j^i(t)$ and the position $x_j^i(t)$ of the j-th element of the i-th particle at the t-th generation is given by the following formulae:

$$v_j^i(t) = k \cdot \{\tilde{\omega}(t) \cdot v_j^i(t-1) + \varphi_1 \cdot r_1 \cdot (p_j^i - x_j^i(t-1)) + \varphi_2 \cdot r_2 \cdot (g_j - x_j^i(t-1))\} \quad (5)$$

and

$$x_j^i(t) = x_j^i(t-1) + v_j^i(t) \quad (6)$$

where
$p^i = \left[\ p_1^i \quad p_2^i \quad ,\ldots \quad p_\kappa^i\ \right]$ is the best position of the particle i, and $g = \left[\ g_1 \quad g_2 \quad ,\ldots \quad g_\kappa\ \right]$ is the best particle among all the particles; r_1 and r_2 are random numbers in the range of [0,1]. $\tilde{\omega}\,(t)$ is the fuzzy inertia weight factor; φ_1 and φ_2 are acceleration constants; k is the constriction factor derived from the stability analysis of (5) for assuring the system to converge but not prematurely [21]. In this paper, k is related to φ_1 and φ_2 as follows:

$$k = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \tag{7}$$

where $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$.

The particle velocity is limited by a maximum value $v_{\max}$ in (5). This parameter $v_{\max}$ determines the resolution of the searched regions between the present position and the target position. The value of this limit governs the local exploitation of the problem space. Practically, it emulates the incremental changes of human learning. If the value of $v_{\max}$ is too large, the particles might fly past good solutions. If the value of $v_{\max}$ is too small, the particles may not sufficiently explore beyond the local solutions. Based on our experiments, it is suggested $v_{\max}$ can be assigned with a value of 10% to 20% of the dynamic range of each element. After updating the velocity of all particles, we get a new swarm $X(t)$ based on (6). To ensure every particle element x_j^i in $X(t)$ falls within the range $\left[\ \rho_{min_j},\ \ \rho_{max_j}\ \right]$, we add the following conditions. If $x_j^i(t) > \rho_{\max_j}$, the updated $x_j^i(t)$ should be equal to $\rho_{\max_j}$. Similarly, if $x_j^i(t) < \rho_{\min_j}$, the updated $x_j^i(t)$ should be equal to $\rho_{\min_j}$. Here, $\rho_{\mathbf{min}} = \left[\ \rho_{\min_1} \quad \rho_{\min_2} \quad \cdots \quad \rho_{\min_\kappa}\ \right]$ and $\rho_{\mathbf{max}} = \left[\ \rho_{\max_1} \quad \rho_{\max_2} \quad \cdots \quad \rho_{\max_\kappa}\ \right]$; $\rho_{\min_j}$ and $\rho_{\max_j}$ are the minimum and maximum values of $x_j^i(t)$ respectively, and $j = 1, 2, ..., \kappa$.

Algorithm 4.1: PSEUDO CODE FOR iPSO($X(t)$)

$t \leftarrow 0$
Initialize $X(t)$
Define the probability of CM operation p_{cm}
output ($f(X(t))$)
while <not termination condition>
do
- $t \leftarrow t + 1$
- **output** (t/T)
- **output** ($||\varsigma(t)||$ based on (8) and (9))
- Find the inertia weight $\tilde{\omega}_k(t)$ by using fuzzy inference system based on (10) − (12).
- Update velocity $v(t)$ based on (5).
- Find the control parameter $\beta(t)$ by using fuzzy inference system based on (15) − (16).
- Generate a random number R_{cm}
- **if** $R_{cm} > p_{cm}$ **then** Perform cross-mutated operation based on (13) − (14).
- **if** $v(t) > v_{\max}$ **then** $v(t) = v_{\max}$
- **if** $v(t) < -v_{\max}$ **then** $v(t) = -v_{\max}$
- Generate a new swarm $X(t)$ based on (6).
- **if** $x_j^i(t) > \rho_{max_j}$ **then** $x_j^i(t) = \rho_{max_j}$
- **if** $x_j^i(t) < \rho_{min_j}$ **then** $x_j^i(t) = \rho_{min_j}$
- **output** ($f(X(t))$)

return (g)
comment: g is the best particle among all particles (solution)

4.1. fuzzy inertia weight

In (5), a nonlinear inertia weight $\tilde{\omega}(t)$ is proposed to enhance the searching performance. The value of $\tilde{\omega}(t)$ is evaluated by a 2-input fuzzy inference system as one of its 2 outputs. (The other output is the control parameter

$\beta(t)$ of the CM operation that will be discussed in the later sub-section.) The inputs of the fuzzy inference system are $||\varsigma(t)||$ and t/T. $||\varsigma(t)||$ is the normalized standard deviation of fitness values among all the particles, of which a larger value implies the particles being far apart from one another. The term $||\varsigma(t)||$ is given by:

$$||\varsigma(t)|| = \sqrt{\frac{1}{\gamma}\sum_{i=1}^{\gamma}\left(||f\left(\mathbf{x}^i\left(t\right)\right)|| - ||\bar{f}\left(\mathbf{x}^i\left(t\right)\right)||\right)^2} \tag{8}$$

where

$$||\bar{f}\left(\mathbf{x}^i\left(t\right)\right)|| = \frac{1}{\gamma}\sum_{i=1}^{\gamma}||f\left(\mathbf{x}^i\left(t\right)\right)|| \tag{9}$$

and $\|\cdot\|$ denotes the l_2 vector norm.

The following fuzzy rules govern the fuzzy inertia weight $\tilde{\omega}\left(t\right)$:

$$\text{Rule } j: \text{IF } ||\varsigma(t)|| \text{ is } \mathrm{N}_1^j, \text{ AND } t/T \text{ is } \mathrm{N}_2^j, \text{ THEN } \tilde{\omega}\left(t\right) = \sigma_j,$$

$$j = \ 1, \ \ 2, \ \ \ldots, \ \ \varepsilon \tag{10}$$

where N_1^j and N_2^j are fuzzy terms of rule j, ε is the number of rules, $\sigma_j \in \left[\begin{array}{cc} \omega_{min} & \omega_{max} \end{array}\right]$ is a singleton to be determined, with ω_{min} and ω_{max} being set at 0.1 and 1.1 respectively [13, 22]. The final value of $\tilde{\omega}\left(t\right)$ is given by:

$$\tilde{\omega}\left(t\right) = \sum_{j=1}^{\varepsilon} m_j\left(t\right)\sigma_j \tag{11}$$

where

$$m_j\left(t\right) = \frac{\mu_{\mathrm{N}_1^j}\left(||\varsigma(t)||\right) \times \mu_{\mathrm{N}_2^j}\left(t/T\right)}{\sum_{j=1}^{\varepsilon}\left(\mu_{\mathrm{N}_1^j}\left(||\varsigma(t)||\right) \times \mu_{\mathrm{N}_2^j}\left(t/T\right)\right)} \tag{12}$$

$\mu_{\mathrm{N}_1^j}\left(||\varsigma(t)||\right)$ and $\mu_{\mathrm{N}_2^j}\left(t/T\right)$ are the membership function values corresponding to N_1^j and N_2^j respectively.

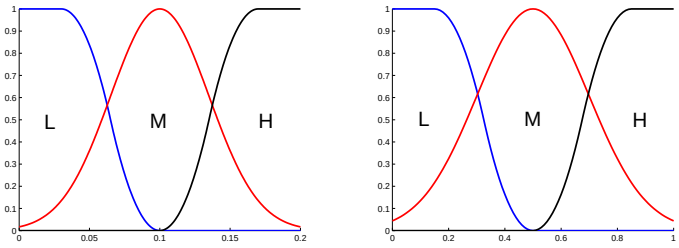
As shown in Fig. 4, the fuzzy inference system uses three membership functions to model each input: L (Low), M (Medium), and H (High).

The output singletons use five terms, namely VL (Very Low), L (Low), M (Medium), H (High), and VH (Very High). The threshold values for these five terms are set at 0.1, 0.35, 0.6, 0.85, 1.1 respectively. The values are determined based on the values of ω_{min} and ω_{max}. For example, the output term is Very High, the threshold value is equal to ω_{max} (=1.1). If the output term is Medium, then the threshold value is equal to $(\omega_{max}+\omega_{min})/2$ (=0.6). With $||\varsigma(t)||$ and t/T as inputs, the 9 linguistic IF-THEN fuzzy rules for determining $\tilde{\omega}(t)$ are given as follows:

Rule 1: IF $||\varsigma(t)||$ is "L" AND t/T is "L", THEN $\tilde{\omega}(t)$ is "VH" (= 1.1)
Rule 2: IF $||\varsigma(t)||$ is "M" AND t/T is "L", THEN $\tilde{\omega}(t)$ is "H" (= 0.85)
Rule 3: IF $||\varsigma(t)||$ is "H" AND t/T is "L", THEN $\tilde{\omega}(t)$ is "VH" (= 1.1)
Rule 4: IF $||\varsigma(t)||$ is "L" AND t/T is "M", THEN $\tilde{\omega}(t)$ is "M" (= 0.6)
Rule 5: IF $||\varsigma(t)||$ is "M" AND t/T is "M", THEN $\tilde{\omega}(t)$ is "M" (= 0.6)
Rule 6: IF $||\varsigma(t)||$ is "H" AND t/T is "M", THEN $\tilde{\omega}(t)$ is "H" (= 0.85)
Rule 7: IF $||\varsigma(t)||$ is "L" AND t/T is "H", THEN $\tilde{\omega}(t)$ is "VL" (= 0.1)
Rule 8: IF $||\varsigma(t)||$ is "M" AND t/T is "H", THEN $\tilde{\omega}(t)$ is "VL" (= 0.1)
Rule 9: IF $||\varsigma(t)||$ is "H" AND t/T is "H", THEN $\tilde{\omega}(t)$ is "L" (= 0.35)

The rationale of the fuzzy rules for determining $\tilde{\omega}(t)$ is given as follows. The value of t/T represents the evolution stage (a small t/T represents an early stage.) The value of $\tilde{\omega}(t)$ is set higher when the value of t/T is smaller (in early stage) so that a larger value of the particle velocity is given for global searching. Similarly, a larger value of t/T implies a smaller value of the particle velocity for local searching and fine-tuning. Thus, $\tilde{\omega}(t)$ of the fuzzy rules 1, 2, and 3 (t/T is "L") has a larger value than that of the rules 4, 5, and 6 (t/T is "M"). As $||\varsigma(t)||$ is the normalized standard deviation of fitness values among all the particles, a large value of $||\varsigma(t)||$ implies that the particle locations are far away from one another. In rules 1 to 3, the searching process is in its early stage (t/T is "L"). When $||\varsigma(t)||$ is "H", the wide-spread particle locations implies a larger value of $\tilde{\omega}(t)$ should be used for global exploration. When $||\varsigma(t)||$ is "L" in the early stage, the value of $\tilde{\omega}(t)$ is also set large as the chance of the solution being trapped in a local optimum is high. In rule 2, the value of $||\varsigma(t)||$ is "M", and we set the value of $\tilde{\omega}(t)$ to be slightly smaller than that in rules 1 and 3 in the early stage (t/T is "L"). In rule 4 to rule 6, the searching process is in its middle stage (t/T is "M"). The rationale for suggesting the value of $\tilde{\omega}(t)$ is similar to that for rules 1-3. However, when $||\varsigma(t)||$ is "L", the value of $\tilde{\omega}(t)$ is smaller than that when $||\varsigma(t)||$ is "H". It is because the optimal solution may have

been found in the middle stage when a smaller value of $\tilde{\omega}(t)$ is given. In rule 7 to rule 9, the searching process is in its late stage (t/T is "H"). Hence, the searching process is undergoing a fine-tuning process (local exploitation) to reach the optimal solution. As a result, when the value of $||\varsigma(t)||$ is "L", the locations of particles are close to one another and near the optimal solution and the smallest value of $\tilde{\omega}(t)$ is used.



(a) fuzzy input 1: $||\varsigma(t)||$ (b) fuzzy input 2: t/T

Figure 4: Membership functions (a) x-axis: $||\varsigma(t)||$,y-axis: $\mu_{N_1}(||\varsigma(t)||)$ (b) x-axis: t/T ,y-axis: $\mu_{N_2}(t/T)$.

4.2. Cross-mutated operation

The proposed cross-mutated (CM) operation merges the ideas of crossover and mutation operations of the genetic algorithm [23] in order to help the particles escaping from some local optima. By injecting random components into particles, the CM operation improves the iPSO performance, particularly when it is used to tackle multimodal optimization problems with many local minima.

With the CM operation, the velocity of every particle element will have a chance to undergo CM operation governed by a probability of CM operation, $p_{cm} \in \left[\begin{array}{cc} 0 & 1 \end{array}\right]$, which is defined by the user. A random number R_{cm} between 0 and 1 will be generated for each particle element such that if it is less than or equal to p_{cm}, the CM operation will take place on that element. The value of the p_{cm} affects the solution quality, and its sensitivity analysis with experimental results will be discussed later.

After taking the CM operation, the resulting velocity of a particle element

is given by:

$$\bar{v}_j^i(t) = \begin{cases} (1-\beta(t))\, v_j^i(t) + \beta(t)\tilde{v}_j^i(t), r_3 > 0.5 \\ (1-\beta(t))\, v_j^i(t) - \beta(t)\tilde{v}_j^i(t), r_3 \leq 0.5 \end{cases} \tag{13}$$

where

$$\tilde{v}_j^i(t) = 0.25\left\{r_4 \cdot \left(\rho_{\max_j} - \rho_{\min_j}\right) + \rho_{\min_j}\right\} \tag{14}$$

$r_3, r_4 \in \left[\begin{array}{cc} 0 & 1 \end{array}\right]$ is a random number, $v_j^i(t)$ is determined by (5), $\tilde{v}_j^i(t)$ is a random velocity of particle element and its value is bounded within 0.25 of the range of the particle element value; a control parameter $\beta(t)$ is introduced into the CM operation, which is governed by some fuzzy rules based on human knowledge. The maximum velocity and minimum velocity are therefore 0.25 of the range of particle element value. The value of 0.25 is chosen by trial and error through experiments. If this value is too large or too small, the searching performance might be degraded. In (13), the resulting velocity of particle element $\bar{v}_j^i(t)$ combines the information of $v_j^i(t)$ and $\tilde{v}_j^i(t)$, exhibiting the characteristic of the crossover operation. However, in (13), $\bar{v}_j^i(t)$ is changed individually the mutation operation. Therefore, it is called the cross-mutated (CM) operation.

In (13), the control parameter $\beta(t)$ provides a balance to control the resulting velocity $\bar{v}_j^i(t)$ converging toward $v_j^i(t)$ or $\tilde{v}_j^i(t)$. This control parameter is in the range of 0.1 to 0.5. If $\beta(t)$ is approaching 0, $\bar{v}_j^i(t)$ will approach $v_j^i(t)$. Conversely, when $\beta(t)$ is approaching 1, $\bar{v}_j^i(t)$ will approach $\tilde{v}_j^i(t)$. Hence, $\tilde{v}_j^i(t)$ in (14) provides a means for the particle element to escape from a local optimum through a random movement governed by $\beta(t)$, of which the value is generated by the following fuzzy rules:

$$\text{Rule } j: \text{IF } ||\varsigma(t)|| \text{ is } \mathrm{N}_1^j \text{ AND } t/T \text{ is } \mathrm{N}_2^j, \text{ THEN } \beta(t) = \chi_j,$$

$$j = \ 1, \ \ 2, \ \ \ldots, \ \ \varepsilon \tag{15}$$

where χ_j is a singleton to be determined. The final value of $\beta(t)$ is given by:

$$\beta(t) = \sum_{j=1}^{\varepsilon} m_j(t)\chi_j \tag{16}$$

where $m_j(t)$ is given by (12)

The output singletons use five terms, namely VL (Very Low), L (Low), M (Medium), H (High), and VH (Very High). As the control parameter of CM is in the range of 0.1 to 0.5. Thus, the threshold values for these five terms are set at 0.1, 0.2, 0.3, 0.4, 0.5 respectively. Here, 9 linguistic IF-THEN fuzzy rules for determining $\beta(t)$ are used and listed as follows:

Rule 1: IF $||\varsigma(t)||$ is "L" AND t/T is "L", THEN $\beta(t)$ is "VH" $(= 0.5)$
Rule 2: IF $||\varsigma(t)||$ is "M" AND t/T is "L", THEN $\beta(t)$ is "H" $(= 0.4)$
Rule 3: IF $||\varsigma(t)||$ is "H" AND t/T is "L", THEN $\beta(t)$ is "VH" $(= 0.5)$
Rule 4: IF $||\varsigma(t)||$ is "L" AND t/T is "M", THEN $\beta(t)$ is "H" $(= 0.4)$
Rule 5: IF $||\varsigma(t)||$ is "M" AND t/T is "M", THEN $\beta(t)$ is "M" $(= 0.3)$
Rule 6: IF $||\varsigma(t)||$ is "H" AND t/T is "M", THEN $\beta(t)$ is "H" $(= 0.4)$
Rule 7: IF $||\varsigma(t)||$ is "L" AND t/T is "H", THEN $\beta(t)$ is "VL" $(= 0.1)$
Rule 8: IF $||\varsigma(t)||$ is "M" AND t/T is "H", THEN $\beta(t)$ is "L" $(= 0.2)$
Rule 9: IF $||\varsigma(t)||$ is "H" AND t/T is "H", THEN $\beta(t)$ is "L" $(= 0.2)$

The rationale for formulating the fuzzy rules is similar to that for formulating the rules governing the fuzzy inertia weight in section 4.1. As mentioned before, rules 1-3 with t/T being "L" correspond to the early searching process, and rules 7-9 correspond to the late searching process. The values of $\beta(t)$ in rules 1-3 are larger than those in rules 7-9. A more significant random velocity (higher value of $\beta(t)$ in (13)) provides more global exploration in the early stage. Conversely, the effect of the random velocity should be reduced in the late stage for more fine-tuning (local exploitation).

In the early stage, when $||\varsigma(t)||$ is "L", the locations of particles are close to one another. Hence, we have to set the value of $\beta(t)$ to be larger than that when $||\varsigma(t)||$ is "M" as the chance of trapping in a local optimum is high. Conversely, when the value of $||\varsigma(t)||$ is "L", a small $\beta(t)$ is used to fine-tune the solutions in the late stage.

5. Results and discussions

To develop the wavelet neural network, 32 pieces of survey data, which represent affective responses with different design elements are used. The training patterns consist of the input vectors regarding the design elements and their corresponding expected outputs regarding the affective responses.

In order to test the learning ability of the wavelet neural network trained by the proposed iPSO, a cross-validation was used. We split the 32 pieces of survey data into 8 subsets randomly and each subsets contains 4 piece of survey data. 6 subsets are used for training and 2 subsets are used for testing. By using cross-validation, each subset is used six times as the training set and twice as the testing set. The training and testing results are then averaged over the splits.

Apart from the the proposed iPSO, other PSO methods (HPSOWM [13], HPSOM [12], and HGAPSO [11]) are employed to tune the wavelet neural network. We aim to minimize the mean absolute error (MAE) of the wavelet neural network by optimizing the following objective function f_{obj}:

$$f_{obj_i} = \frac{1}{M} \sum_{t=1}^{M} |y_i^d(t) - y_i(t)| \tag{17}$$

where $i = 1, 2, ..., 4$. For all PSO methods, the swarm size is set at 50, the number of runs is 50. The acceleration constants φ_1 and φ_1 are set at 2.02 and the maximum velocity v_{max} is equal to 0.2. The number of iteration is 1000. The values of v_{ji} and w_j are bounded between -5 and 5. The values of κ^j is bounded between 0.3 and 1.5. The probability of mutation operation for HPSOWM, HPSOM and HGAPSO are set at 0.1. The probability of CM operation for iPSO is set at 0.01 in this application. The shape parameters of the wavelet mutation and the parameter g of the wavelet mutation for HPSOWM are set at 2 and 10000 respectivity. For HGAPSO, the probability of crossover operation is 0.8. The training results for the four affective responses are tabulated in Table 1.

The result in the table shows that the average training performance of iPSO is better than those obtained by the other PSO methods. In terms of p-value, all values for four affective responses are less than 0.05 ($p < 0.05$) which means that the iPSO method is significantly better than the other PSO methods with a 95% confidence level. To validate the modeling performance, the average testing results are shown in the same table. In this table, we can see that the testing results obtained by iPSO for all affective responses are better than those by other PSO methods. For example, for the S-C (Simple-complex) response, the wavelet neural network obtained by iPSO gives smaller MAE ($5.82 \times 10^{-2} \pm 1.07 \times 10^{-2}$) than the second best wavelet neural network by HPSOM ($6.57 \times 10^{-2} \pm 1.58 \times 10^{-2}$). In terms of p-value for testing, the p values with respect to iPSO for S-C and H-B are

less than 0.05. Hence, the performance of iPSO is significantly better than the other PSO methods with a 95% confidence level. For H-C, the p values for testing are mostly less than 0.05, except that of HPSOM ($p = 0.053$). Generally, $p = 0.053$ can be also considered to be statistically significant. For the response U-G, the iPSO can be considered as slightly statistically significant as the p values are between 0.0667 to 0.1687. The values in this range are acceptable. iPSO is generally the best method among the tabled PSO methods for this affective product design problem.

			iPSO	HPSOWM	HPSOM	HGAPSO
S-C	Training	Mean ($\times 10^{-2}$)	3.61	4.28	4.23	4.25
		Std Dev ($\times 10^{-2}$)	0.81	0.84	0.77	0.80
	Testing	Mean ($\times 10^{-2}$)	5.82	6.74	6.57	6.58
		Std Dev ($\times 10^{-2}$)	1.07	1.59	1.58	1.23
		p-value for training	–	0.0047	0.0059	0.0054
		p-value for testing	–	0.0156	0.0433	0.0186
U-G	Training	Mean ($\times 10^{-2}$)	3.11	3.51	3.42	3.55
		Std Dev ($\times 10^{-2}$)	0.63	0.72	0.66	0.63
	Testing	Mean ($\times 10^{-2}$)	4.77	5.42	5.44	5.19
		Std Dev ($\times 10^{-2}$)	1.01	1.73	1.61	1.22
		p-value for training	–	0.0334	0.0827	0.0141
		p-value for testing	–	0.0891	0.0667	0.1687
H-C	Training	Mean ($\times 10^{-2}$)	2.59	3.18	3.07	3.05
		Std Dev ($\times 10^{-2}$)	0.67	0.71	0.71	0.61
	Testing	Mean ($\times 10^{-2}$)	4.60	5.26	5.08	4.99
		Std Dev ($\times 10^{-2}$)	0.86	1.28	0.95	0.86
		p-value for training	–	0.0032	0.0128	0.0123
		p-value for testing	–	0.0303	0.0530	0.0979
H-B	Training	Mean ($\times 10^{-2}$)	3.38	3.92	3.91	3.95
		Std Dev ($\times 10^{-2}$)	0.99	0.85	0.78	0.80
	Testing	Mean ($\times 10^{-2}$)	5.31	5.96	6.02	6.04
		Std Dev ($\times 10^{-2}$)	1.14	1.14	1.22	1.30
		p-value for training	–	0.0355	0.0323	0.0242
		p-value for testing	–	0.0381	0.0289	0.0294

Table 1: Comparison between different PSO methods for affective product design of mobile phones. All results are averaged ones over 30 runs (p-value illustrates the significant difference between iPSO and the other methods).

6. Conclusion

In this paper, we have proposed an intelligent particle swarm optimization (iPSO) that incorporates an adaptive inertia weight and a new cross-mutated operation to optimize the parameters of variable translation wavelet neural network. In iPSO, the control parameters of cross-mutated operation are determined by a set of fuzzy rules. By introducing the fuzzy system, the solution quality obtained by the iPSO is improved. On applying to optimize the neural network parameters for modeling affective product design of mobile phones, iPSO is found to be successful and outperforms the other PSO methods, including HPSOWM, HPSOM and HGAPSO. In this study, we have two limitations, the first limitation is that choosing the suitable parameter values for the PSO is quite difficult. Most parameter values are determined by trial and error through experiments. The second limitation is that the total iteration number needs to be determined for doing the optimization.

7. Acknowledgment

The work described in this paper was partially supported by a grant from University of Technology Sydney (Activity code: 2032019), and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Account Code G-Y563)

References

[1] P. P. San, S. H. Ling and H. T. Nguyen, "Hybrid PSO-based variable translation wavelet neural network and its application to hypoglycemia detection system", *Neural Computing and Applications*, vol. 23, nos. 7-8, 2013, pp. 2177-2184.

[2] P. P. San, S. H. Ling and H. T. Nguyen, "Industrial application of evolvable block-based neural network to hypoglycemia monitoring system", *IEEE Trans. Industrial Electronics*, vol. 60, no. 12, 2013, pp. 5892-5901.

[3] S. H. Ling, H. H. C. Iu, F. H. F. Leung, and K. Y. Chan, "Improved hybrid PSO-based wavelet neural network for modelling the development of fluid dispensing for electronic packaging", *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, 2008, pp. 3447-3460.

[4] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems", *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, 2005, pp. 1478-1489.

[5] M. Han, J. Fan, and J. Wang, "A dynamic feedforward neural network based on gaussian particle swarm optimization and its application for predictive control", *IEEE Trans. Neural Networks*, vol. 22, no. 9, 2011, pp. 1457-1468.

[6] D. Yi and X. Ge, "An improved PSO-based ANN with simulated annealing technique", *Neurocomputing*, vol. 63, 2005, pp. 527-533.

[7] J. Yu, S. Wang and L. Xi, "Evolving artificial neural networks using an improved PSO and DPSO", *Neurocomputing*, vol. 71, nos. 4-6, 2008, pp. 1054-1060.

[8] G. Y. Lian, K. L. Huang, J. H. Chen, and F. Q. Gao, "Training algorithm for radial basis function neural network based on quantum-behaved particle swarm optimization", *International Journal of Computer Mathematics*, vol. 87, no. 3, 2010, pp. 629-641.

[9] S.H. Ling, F. Jiang, K.Y. Chan, and H.T. Nguyen, "Hybrid fuzzy logic-based particle swarm optimization for flow shop scheduling problem", *nt. J. Computational Intelligence and Applicat.*, vol. 10, no. 3, 2011, pp. 335-356.

[10] L.C Jain, V. Palade, D. Srinivasan, "Advances in Evolutionary Computing for System Design", Springer, 2007.

[11] C. F. Juang, "A hybrid genetic algorithm and particle swarm optimization for recurrent network design", *IEEE Trans. Syst. Man and Cybern. B*, vol. 34, no. 2, 2004, pp. 997-1006.

[12] A. A. E. Ahmed, L. T. Germano, and Z. C.Antonio, "A hybrid particle swarm optimization applied to loss power minimization", *IEEE Trans. Power Syst.*, vol. 20, no. 2, 2005, pp. 859-966.

[13] S.H. Ling, C.W. Yeung, K.Y. Chan, H.H.C. Iu, and F.H.F. Leung, "A new hybrid particle swarm optimization with wavelet mutation for neural network training", in *Proc. Congress on Evolutionary Computation (CEC2007), Singapore*, Sep 2007, pp. 1977-1984.

[14] J. Kuang and P. Jiang, "Analysis of the impact of slight changes in product formal attributes on user's emotions and configuration of an emotional space for successful design", *Journal of Engineering Design*, vol. 21, no. 6, 2010, pp. 693-705.

[15] M. A. Artacho, A. Ballester and E. Alcantara, "Product platform design for a product family based on Kansei engineering", *Journal of Engineering Design*, vol. 20, no. 6, 2009, pp. 589-607.

[16] H. H. Lai, Y. C. Lin and C. H. Yeh, "Form design of product image using grey relational analysis and neural network models", *Computers & Operations Research*, vol. 32, no. 10, 2004, pp. 2689-2711.

[17] K. Y. Chan, C. K. Kwong, T. S. Dillon and K. Y. Fung, "An intelligent fuzzy regression approach for affective product design that captures nonlinearity and fuzziness", *Journal of Engineering Design*, vol. 22, no. 3, 2010, pp. 523-542.

[18] Y. Shi, "Empirical study of particle swarm optimization", *Proc. of the 1999 Congress on Evolutionary Computation*, Washington, US, 1999, pp. 1945-1950.

[19] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. Evol Comput.*, vol. 6, no. 1, 2002, pp. 58-73.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization", *Proc. 30th IEEE Conf. Decision and Control*, vol. 4, 1995, pp. 1942-1948.

[21] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *Proc. IEEE Congress on Evolutionary Computing*, vol. 1, 2000, pp. 84-88.

[22] N. Mo, Z. Y. Zou, K. W. Chan, and T. Y. G. Pong, "Transient stability constrained optimal power flow using particle swarm optimization", *IET Proceedings - Generation, Transmission and Distribution*, vol. 1, no. 3, 2007, pp. 476-483.

[23] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, USA, 1996.