2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evolutionary Regional Network Modeling for Efficient Engineering Optimization

Jyh-Cheng Yu*, Zhi-Fu Liang,
Department of Mechanical and Automation Engineering,
National Kaohsiung First University of Science and Technology,
Kaohsiung 824, Taiwan, R.O.C.
jcyu@nkfust.edu.tw

Tsung-Ren Hung
Department of Mechanical Engineering
National Taiwan University of Science and Technology
Taipei 106, Taiwan, R.O.C.

***Abstract*—This study presents a soft computing based optimization methodology, the evolutionary regional neural network modeling for engineering applications with sampling constraints. Engineering optimization often involves expensive experiment costs. Intelligent optimization advocates establishing a neural network model using small training samples such as orthogonal array to set up a surrogate model for the engineering system followed by an optimum search in the model to reduce optimization cost. However, scarce training samples might compromise modeling generality for a complex problem. Empirical rules suggest reliable predictions are likely restricted to the neighboring space of training samples, and interpolating designs are more reliable than extrapolating designs. To avoid imperfection of model due to small learning samples, an evolutionary regional network model is set up to confine the search of quasi-optimum using genetic algorithm. The constrained search in the regional network model provides a reliable quasi-optimum. The verification of the optimum is added to the learning samples to retrain the regional network model while the size and the distribution of reliable space will evolve intelligently during the optimization iteration using a fuzzy inference according to the prediction accuracy. An engineering case study, the optimal die gap parison programming of extrusion blow molding process for a uniform thickness, is presented to demonstrate the robustness and efficiency of the proposed methodology.**

Keywords- Neural Network; Genetic Algorithm; Fuzzy Logics; Evolutionary Optimization; Surrogate Model, Extrusion Blow Molding

I. INTRODUCTION

Cost constraints in engineering optimization often impose a limit on the number of samples that could be actual experiments or time consuming numerical simulations. Taguchi's method [1] is well known for its efficiency and simplicity in parameter design. Inspired from statistical factorial experiments, Taguchi's method features orthogonal arrays (OAs) and analysis of mean (ANOM) to estimate the effects of design variables. Use of orthogonal arrays helps reducing the number of experiments; however, the optimum prediction is sensitive to the selection of parameter ranges and possible interaction effects. Also, the restriction of parameter values to factorial levels reduces possibilities of exploring better designs between preset levels.

An artificial neural network (ANN) model could be established from a set of training data as a surrogate model to predict system responses. Nonlinear continuous systems can be approximated by multilayer artificial neural network [2]~[4]. The optimum setting of ANN parameters in a multilayer perceptron (MLP) network trained with the back propagation algorithm can be identified using Taguchi's design of experiments [5] and genetic algorithms [6]. Instead of a direct interaction with the real system, the optimum search is applied to the surrogate model to increase the searching quality and reduce the experimental cost. The combinations of a simulated neural network and evolutionary optimization [7], such as genetic algorithms, have thus attracted much research attention [8]~[10].

The size and the distribution of training data are essential to the prediction accuracy of a simulated network model. The training samples are usually existed field data or planned experiments such as Taguchi's orthogonal arrays (OA). However, possible bias distribution of field data will decrease the prediction accuracy of simulated models. Though OA experiments provide a smaller and even sample distribution, scarce training data might result in the lack of model generality for a complex problem. Also, because ANN is mainly based on fitting methods to learning data, the prediction reliability will be related to the distance between the point of interest and the nearest learning sample [11] and the estimation error of the learning sample [12].

This study proposes a novel optimization methodology based the iterative constrained search in the evolutionary regional network model that is defined as the union of the neighboring space of the training sample. The regional model will evolve intelligently based on the fuzzy inference of the prediction accuracy. The process parameter optimization of the extrusion blow molding of a HDPE gas tank with uniform thickness is presented to demonstrate the robustness and efficiency of the proposed scheme.

II. REGIONAL NEURAL NETWORK MODEL FROM LIMITED TRAINING SAMPLES

A. Generality of ANN model

To reduce experiment costs in the engineering applications of design optimization, an artificial neural network (ANN) model is established from a set of finite sample data served as a surrogate model to predict system responses in optimum search. A four-layer structure with two hidden layers is selected, which provides good approximation for most engineering applications [5]. This study applies Taguchi's method to tune both network structure and parameters of the neural network including the number of hidden neurons, transfer functions, and learning parameters . The number and the distribution of training data are essential to the prediction accuracy of a simulated neural network. If the distribution of training data is biased or the number of samples is not sufficient, the generality of the trained network will be in jeopardy, which affect the prediction accuracy of optimum. Taguchi's orthogonal array is often used as the initial training samples to reduce the number of experiments [13]. However, a small number of experiment will also cause low generality of trained network.

For the example of the Peaks function as shown in (1), the theoretical contour plot is highly non-linear and includes 3 peaks and 2 valleys and flattens out gradually outside the range shown in Fig.1(a). Fig. 1(b) is a simulated network model from 9 random learning samples and 4 testing samples. The simulated model shows a rough profile of the original function, but significant errors present especially for the prediction farther away from training samples. Increasing the number of samples is a simple solution, but will raise a cost concern.

$$z = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right)e^{-x^2-y^2} - \frac{1}{3}e^{-(x+1)^2-y^2} \quad (1)$$

B. Regional Neural Network Model

Empirical rules for the prediction accuracy of a simulated model from limited training samples suggest:

(1) The prediction accuracy is worse for a design farther away from the training samples.
(2) The prediction accuracy of an interpolation design is better than an extrapolation designs.

This study proposes the trained NN is effective only in the reliable region of the model. Here, the Sampling Distance, r_{ij}, is proposed as a neighboring index between a predictive design, P_i, and the sample S_j, which is defined as the mean Euclid distance:

$$r_{ij} = \left[\frac{1}{n}\sum_{k=1}^{n}(P_{ik} - S_{jk})^2\right]^{0.5} \quad (2)$$

where n represents the number of variables. To prevent the scaling problem, continuous variables x_k are first normalized to z_k to transform all the dimensional entries of the training samples into the space of [-1, +1]. For discrete variables, the factorial values are assigned equally spaced between -1 and +1.

As the predictive errors of the training samples will be controlled to an acceptable level, the neighboring space of the training samples is likely as reliable as the training samples. The reliable region is defined as a hyper-sphere centered at a training sample. To differentiate interpolating and extrapolating designs, the Sampling Enclosure Space (*SES*) is defined as a least convex hyper-polyhedron enclosing all training samples. Two parameters: the Reliable Interpolating Radius (R_I) and the Reliable Extrapolating Radius (R_E) are proposed to define the reliable regions of a simulated model (Fig. 2). The reliable regions outside and inside the Sampling Enclosure Space (*SES*) are defined by hyper-spheres with radius R_E and hyper-spheres with radius R_I respectively. Because the interpolating designs have better prediction accuracy than extrapolating designs, R_E is often smaller than R_I. The determination of the reliable radii will depend on the model complexity and prediction accuracy, and is a key issue in the following study. However the determination of R_E and R_I will be an important issue.

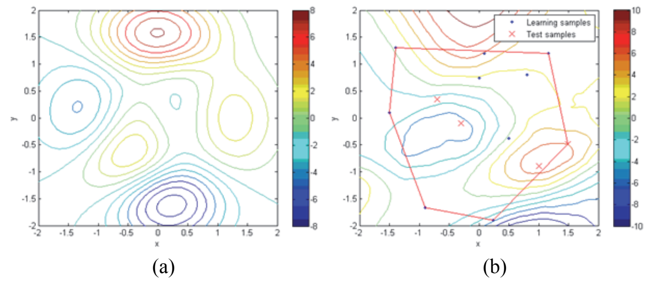


Fig. 1. Contour plots of the Peaks function (a) theoretical plot (b) sample contour plot of the ANN model from 13 training samples

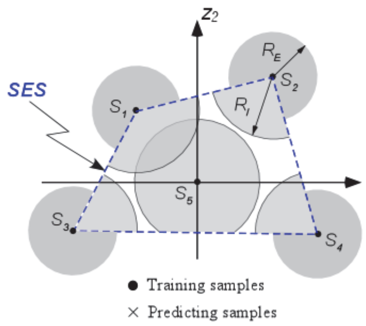


Fig. 2. Schematic reliable regions of a two-dimensional example with 5 learning samples.

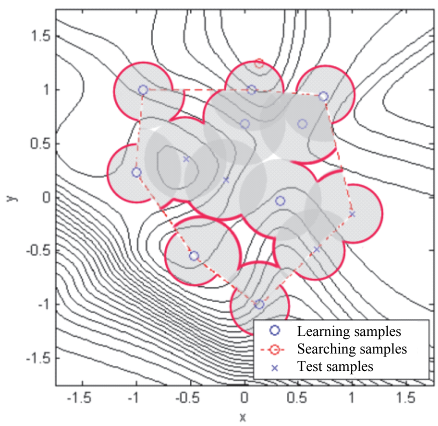


Fig. 3. The reliable regions ($R_I = 0.3$, $R_E = 0.2$) of a simulated network model

III. EVOLUTIONARY OPTIMIZATION USING REGIONAL NEURAL NETWORK

A. Flowchart of Evolutionary Optimization

The searching flowchart of the proposed optimization scheme, Evolutionary Regional Neural network with Genetic Algorithm, (ERNGA), is shown in Fig. 4. A simulated network model is established from the initial training data. The proposed scheme confines the GA search to the reliable regions of the network model for a quasi-optimum. The reliable regions are the union of the hyper-spheres defined by the R_I and the R_E surrounding the training data. The verification of the quasi-optimum provided by the GA search is introduced to retrain the neural network and to adjust the reliable radii using a fuzzy inference. The reliable regions of the simulated network will then continue to evolve from the accumulation of training samples and self-learning mechanism of the reliable radii. The optimum search iterates until the convergence of optimum.

B. Fuzzy Inference for the Reliable Radii

The selections of R_I and R_E depend on the model generality. The verification of the regional optimum will become a feedback mechanism to adjust the Reliable Radii based on the following fuzzy concepts. If the verification result is close to the model prediction, increase the Reliable Radii to expand the reliable regions; otherwise, decrease the Reliable Radii. This section proposes a heuristics based fuzzy inference to intelligently evolve the Regional Neural Network Model as Table 1.

Table 1 FUZZY INFERENCE RULES OF RELIABLE RADII

1	If Extrapolation and *MEI* is [Small] then [Slightly Increase] R_E and [Increase] R_I
2	If Extrapolation and *MEI* is [Medium] then [Maintain] R_E and [Slightly Increase] R_I
3	If Extrapolation and *MEI* is [Large] then [Slightly decrease] R_E and [Maintain] R_I
4	If Interpolation and *MEI* is [Small] then [Maintain] R_E and [Slightly increase] R_I
5	If Interpolation and *MEI* is [Medium] then [Maintain] R_E and [Slightly decrease] R_I
6	If Interpolation and *MEI* is [Large] then [Slightly decrease] R_E and [Decrease] R_I

Three linguistic levels are defined to describe the condition variable of prediction error: Large, Medium, and Small based on the Modeling Error Index, (*MEI*) in Eq.(3). Five action levels are defined to describe the adjustment factor (*AF*) for the Reliable Radii: Increase, Slightly Increase, Maintain, Slightly Decrease, and Decrease. Standard triangle membership functions are applied to define the fuzzy variables as illustrated in Fig. 5 and Fig. 6.

$$MEI = \frac{|Y_j - T_j|}{RMSE_{Test}} \tag{3}$$

where Y_j is the model prediction, T_j is the verification result of the quasi-optimum at iteration j, and $RMSE_{Test}$ is the root mean squared error of the testing samples.

The adjustment factors from the fuzzy inference are used to modify the reliable radii of interpolation and extrapolation as shown in Eq. (4) and (5) respectively, and dynamically adjust the reliable regions in the next iteration.

$$(R_I)_{i+1} = AF_I \times (R_I)_i \tag{4}$$

$$(R_E)_{i+1} = AF_E \times (R_E)_i \tag{5}$$

Therefore, the searched result will be less sensitive to the imperfection of the trained model. If the verification result is good, the reliable regions will expand in the next iteration to investigate more possible regions; otherwise, the reliable region will retract for a more conservative search.

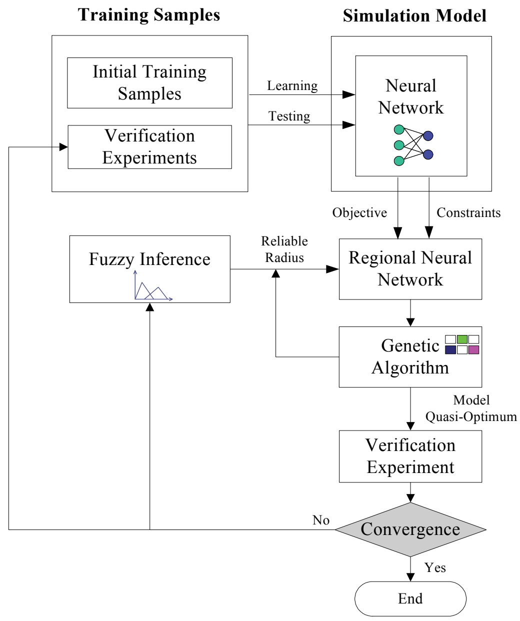


Fig. 4.Optimization flowchart of ERNGA.

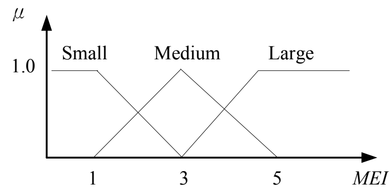


Fig. 5. Membership functions of the condition levels of *MEI*.

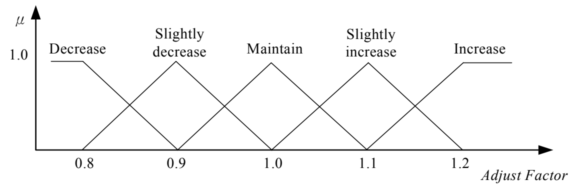


Fig. 6. Membership functions of the action levels of the adjusting factor (*AF*).

C. Optimum Search of the Peaks Function

Optimization of the Peaks function in (1) is used to illustrate the applications of ERNGA. L9 OA is selected for the learning samples, and L4 OA is selected for the testing samples in the initial investigating ranges of $x, y \in [-2,+2]$ for the Peaks function. Fig. 7 show that the iterations converge smoothly to the theoretical optimum with relative error of

0.7% at 24th iteration although the accuracy of the initial NN model is poor due to a bad distribution in the flat region for initial samples. As the addition of the learning samples from the searched optima and the self-learning mechanism of the reliable regions, the accuracy of the simulated model improves, especially in the most probable regions of the global optimum as shown in Fig. 8. ERNGA will congregate additional samples in the most probable regions of the design optimum without wasting costly experimental resources in unlikely regions.

If the constraint of the reliable regions is relieved as in convention intelligent optimization, GA will assume global accuracy, and search for the design with the best fitness based on the model prediction in the investigating range. The prediction accuracy of the optimum is poor because of the lack of generality for the network model. Although the best result of the iteration is happened to be close to the global optimum with the relative error of 1.6%, the iteration is very unstable and shows no convergence tendency in 45 iterations as shown in Fig. 9. The search result is very sensitive to the global accuracy. Also, unlike ERNGA, additional samples from the iteration of conventional NN and GA may scatter all over, and are thus less efficient.

A continuous discrepancy presents between the predicted optimum and the verified results for the conventional NN and GA iteration due to the over-confidence on the global accuracy of the simulated models. On the other hand, the fuzzy inference of reliable radii in ERNGA constrains the GA search and provides a reliable quasi-optimum. The evolutionary regional network model expands intelligently to the most probable regions of global optimum.

IV. Optimization of Extrusion Blow Molding

A. Parision Programing for Extrusion Blow Molding

Extrusion blow molding is a low cost manufacturing process for complex hollow parts [14] which involves four processes: parison extrusion, mold clamping, parison inflation, and part solidification. First, the parison extrusion produces a molten thermoplastic tube from the die. The parison shape is determined by the die geometry, die gap programming, and flow rate. To control the parison thickness over time, a mandrel is moved in and out to the die as in Fig. 10. The parison profile can be controlled by manipulating the die gap opening over time. The parison is clamped and high-pressure air is blown into it to obtain the hollow part. The design objective of the parison programming is to control the die gap openings to obtain a uniform thickness of the blown part [15].

The programming points are the die gap openings of the parison in the extruder specified as a function of time. For the gas tank made of high density polyethylene (HDPE) shown in Fig. 11, the die gap openings at 13 discrete extrusion times: $P(t_1) \sim P(t_{13})$ are identified as the design variables. The finite element tool, BlowSim developed by National Research Council of Canada is applied to simulate parison extrusion and blow molding processes to obtain the thickness distribution of the inflated part [16].

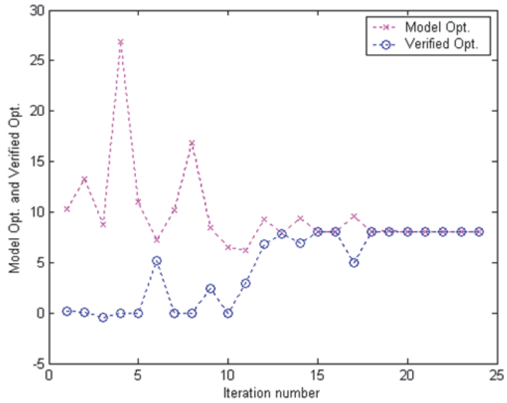


Fig. 7. ERNGA iteration for the Peaks problem

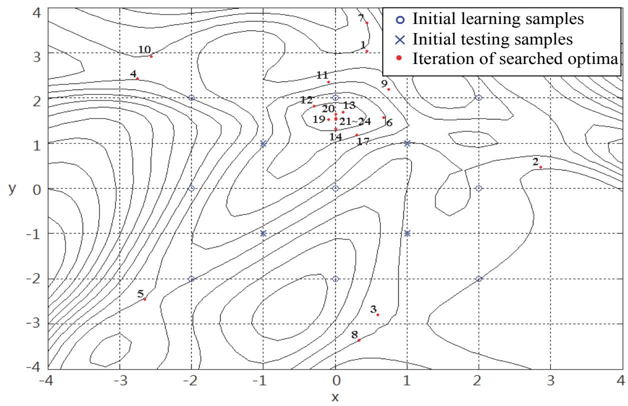


Fig. 8. The contour plot of the simulated model and the distribution of the searched optima using ERNGA

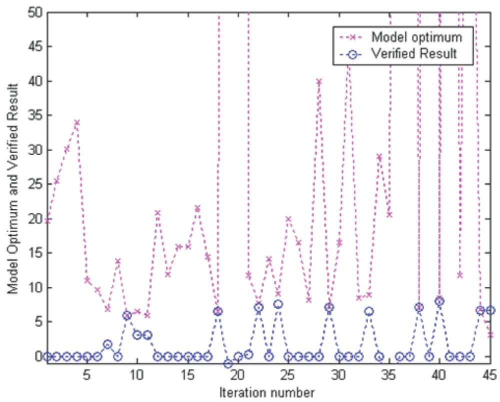


Fig. 9. Conventional NN and GA iteration using for the Peaks problem

B. Objective Function

The design objective is to target a uniform part thickness of 5 mm. Any deviation from the on-target thickness will impose a quality loss. To reach this goal, the objective function is defined as the mean squared deviation of the wall thickness,

$$Objective = \frac{\sum_{i=1}^{n}(h_i - T)^2}{n} \tag{6}$$

where h_i stands for the thickness of node i, T for the target thickness, and n for the total number of nodes of the simulation model. The parameter design searches for the optimum with a minimum objective.

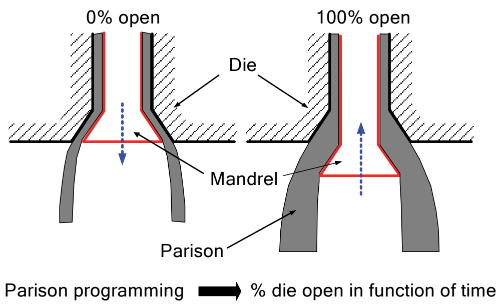
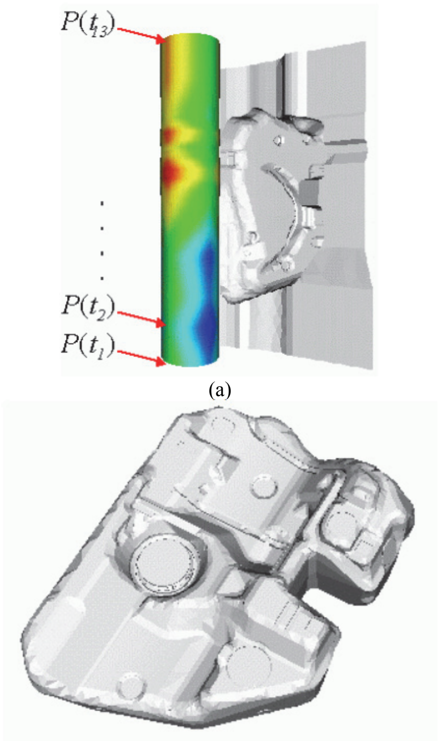


Fig. 10. Control of the parison thickness using the parison programming



(a)

(b)

Fig. 11. Extrusion blow molding of the gas tank example. (a) Exemplar programming points of the parison extrusion (b) the blown part.

C. Design of Experiment

The die gap openings at 13 discrete extrusion times are selected as the design variables. The initial design adopts a uniform die gap opening of 10% for first four control points and 20% for the rest. Taguchi's L36 orthogonal array as in Table 2 is selected as the experimental design. For each opening, we assume a three-level variation around the initial design located in the middle of the design space. The range between upper and lower levels represents the design space.

Taguchi's method applies the analysis of means (ANOM) to estimate the factor effect. Fig. 12 represents the factor effects for each die opening on the objective. The additive model predicts the optimum treatment to be $A_1B_3C_3D_2E_2F_1G_1H_1I_1J_1\ K_1L_1M_3$. The variation of Taguchi's optimum using BlowSim simulation shows the objective of 6.44, which is not even the best design in Table 2. The failure of Taguchi's approach might due to interactions among design variables and strong system non-linearity.

D. Establish Regional Nerual Network Model

This study applies Taguchi's L36 orthogonal array as the initial training samples for the regional NN model. The L36 experiments are divided into two smaller orthogonal arrays as differentiated by the shade in Table 2. Eighteen experiments are used as learning samples and the others are used as testing samples. There are 15 neurons in the hidden layer. The initial learning rate is set to 1.0 and the initial momentum term is set to 0.5. The steepest descent method is combined with learning samples to train the weighted matrix among network units, thus, increasing the prediction accuracy of the neural network.

Table 2. L36 ORTHOGONAL ARRAY

	A	B	C	D	E	F	G	H	I	J	K	L	M	
L36	Pt1	Pt2	Pt3	Pt4	Pt5	Pt6	Pt7	Pt8	Pt9	Pt10	Pt11	Pt12	Pt13	Obj.
1	5	5	5	5	30	30	30	30	30	30	30	35	35	4.93
2	5	10	10	10	40	40	40	40	40	40	40	40	40	10.6
3	5	15	15	15	50	50	50	50	50	50	50	45	45	18.5
4	5	5	5	5	30	40	40	40	40	50	50	45	45	13.9
5	5	10	10	10	40	50	50	50	50	30	30	35	35	13.1
6	5	15	15	15	50	30	30	30	30	40	40	40	40	10.2
7	5	5	10	15	30	40	50	50	30	40	40	45	40	12.1
8	5	10	10	15	30	40	50	30	30	40	50	45	35	11.8
9	5	15	15	5	40	50	30	40	40	50	30	35	40	9.62
10	5	5	5	15	40	30	50	40	50	40	30	45	40	11.5
11	5	10	10	5	50	40	30	50	30	50	40	35	45	11.3
12	5	15	15	10	30	50	40	30	40	30	50	40	35	11.5
13	10	5	10	15	30	50	40	30	50	50	40	35	40	13
14	10	10	15	5	40	30	50	40	30	30	50	40	45	10.9
15	10	15	5	10	50	40	30	50	40	40	30	45	35	12.6
16	10	5	10	15	40	30	30	50	40	50	50	40	35	13.2
17	10	10	15	5	50	40	40	30	50	30	30	45	40	12.1
18	10	15	5	10	30	50	50	40	30	40	40	35	45	10.8
19	10	5	10	5	50	50	50	30	40	40	30	40	45	15
20	10	10	15	10	30	30	30	40	50	50	40	45	35	11.6
21	10	15	5	15	40	40	40	50	30	30	50	35	40	10.4
22	10	5	10	10	50	50	30	40	30	30	50	45	40	14.9
23	10	10	15	15	30	30	40	50	40	40	30	35	45	7.59
24	10	15	5	5	40	40	50	30	50	50	40	40	35	12.4
25	15	5	15	10	30	40	50	50	30	50	30	40	40	10.7
26	15	10	5	15	40	50	30	30	40	30	40	45	45	10.3
27	15	15	10	5	50	30	40	40	50	40	50	35	35	12.6
28	15	5	15	10	40	40	30	30	50	40	50	35	45	10.7
29	15	10	5	15	50	50	40	40	30	50	30	40	35	15.6
30	15	15	10	5	30	30	50	50	40	30	40	45	40	11.4
31	15	5	15	15	50	40	50	40	40	30	40	35	35	11.2
32	15	10	5	5	30	50	30	50	50	40	50	40	40	18.6
33	15	15	10	10	40	30	40	30	30	50	30	45	45	8.08
34	15	5	15	5	40	50	40	50	30	40	40	40	45	12.3
35	15	10	5	10	50	30	50	30	40	50	50	35	40	13.1
36	15	15	10	15	30	40	30	40	50	30	30	40	45	8.63
Initial	10	10	10	10	40	40	40	40	40	40	40	40	40	11.5

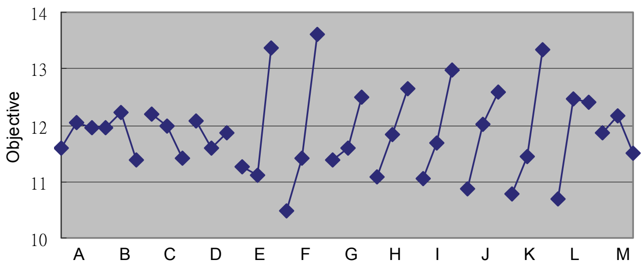


Fig. 12 Factor effects plot of the 13 design variables

Table 3 Genetic Algorithm Parameters

Population size	Pool selection style	Scale style	Cross Over rate	Mutation rate	Max iteration
37	Parent and offspring	Linear scale	0.75	0.1	100

E. *Iteration for ERNGA*

Real-parameter Genetic Algorithm (RGA) [17] is applied to search for the quasi optimum of the regional network model. The GA parameters used in this study are listed in Table 3. The fitness function adopts the negation of the objective in (6) predicted from the simulated model.

The prediction generality of a simulated network is limited if the number of training samples is deficient. This study suggests constraining the GA search to the reliable regions of the network model. During the evolution processes of mutation, crossover, and reproduction, the offspring outside the reliable regions of the network model will be discarded and regenerated to constrain the searching domain. The GA search constrained to the regional neural network prevents the prediction error due to deficient generality and provides a reliable quasi-optimum.

The extrusion blow molding using the parameter of the provided quasi optimum is conducted to verify the replication quality, which is later introduced to the learning samples to retrain the model. The self-learning mechanism from this fuzzy inference of reliable radii dynamically adjusts the reliable regions. Therefore, the searched result will be less sensitive to the imperfection of the trained model. If the verification result is good, the reliable regions will expand in the next iteration to investigate more possible regions; otherwise, the reliable region will retract for a more conservative search. The evolution and the optimum search of the Regional Neural Network iterate until the convergence of the optimum. The iteration history of ERNGA is shown in Fig. 13.

F. *Comparison of Results*

Fig. 14 presents the die gap openings of the parison programming obtained from Taguchi's method and ERNGA. ERNGA reached the convergence criteria at iteration 6. Table 4 compares the experimental results of initial design, Taguchi's optimum and ERNGA's optimum. Each iteration requires an additional function call to the finite element simulation. There are total 43 simulation samples after 7 iterative evolving and search using ERNGA. The optimum design from ERNGA exhibits the smallest objective of 3.42 which stands for of an average thickness of 5.63 (mm) and the standard deviation of 1.74 (mm), which outperforms 11.48 of the initial design and 6.44 of Taguchi's optimum. Fig. 15 compares the thickness distribution of the optimization results. ERNGA's result is not only closer to the target thickness of 5 (mm), but also presents a more uniform distribution.

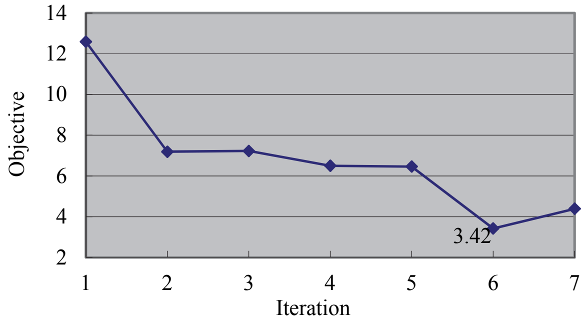


Fig. 13. Iteration history of the gas tank example using ERNGA

Table 4 Comparison of Various Optima

	Initial Design	Taguchi's	ERNGA's
Mean	7.40	6.17	5.63
Standard Deviation	2.39	2.25	1.74
Objective	11.48	6.44	3.42

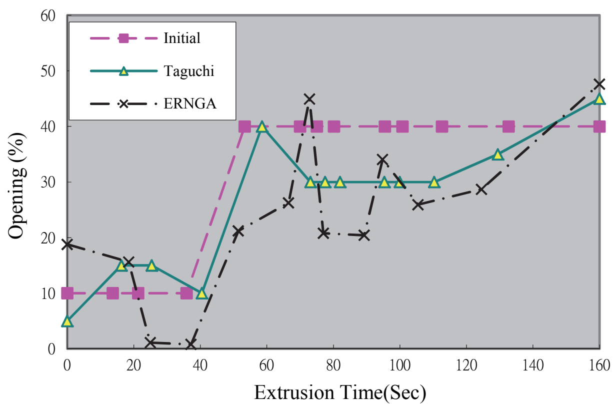


Fig. 14 Comparison of the die gap openings of the parison programming obtained using Taguchi's method and ERNGA

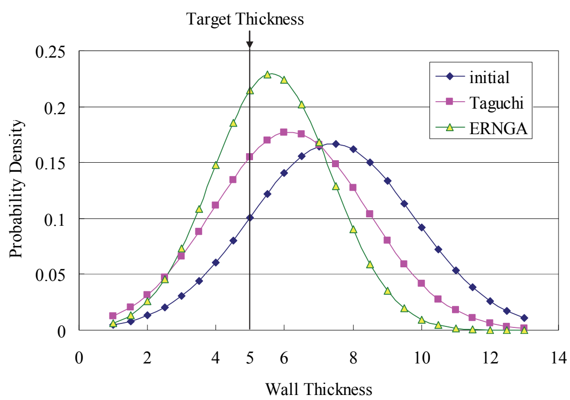


Fig. 15 Thickness distributions of various designs for the gas tank molding problem

V. Conclusions

The integration of a simulated neural network from sampling data following by an optimum search using genetic algorithm has shown a promising tool in real life applications. However, if the generality of prediction accuracy is compromised due to limited number or possible biased distribution of training samples, the iteration process may be unstable and inefficiency. Certainly, a well-trained network will have better prediction accuracy and thus reduce the number of iteration in the optimum search, but the optimization of the simulated model is no guarantee in engineering applications. The proposed evolutionary regional neural network reduces the sensitivity of the searched optimum to the trained generality of the network model. The proposed evolutionary regional network constrains the optimum search in the reliable space that evolves intelligently according to the prediction accuracy, and reaches a reliable optimum with much less iteration. ERNGA provides additional samples in the most probable regions to increase sampling efficiency, which is particularly important in engineering applications. The engineering application of extrusion blow molded gas tank demonstrates the effectiveness of the proposed scheme compared with Taguchi's method and conventional intelligent optimization.

References

[1]. G. Taguchi, "Performance Analysis Design." *Int. Journal of Production Research*, 16, 1978, 521-530.

[2]. J. S. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence*, Prentice-Hall, 1997.

[3]. K. I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, Vol. 2, 183–192, 1989.

[4]. H. F. Leung, H. K. Lam, S. H. Ling, and K. S. Tam, "Tuning of the structure and parameters of a neural network using an improved genetic algorithm, IEEE Trans. Neural Network," Vol. 14, no. 1, 79–88, Jan. 2003.

[5]. W. Sukthomya, J. Tannock, "The optimisation of neural network parameters using Taguchi's design of experiments approach: an application in manufacturing process modelling," *Neural Computing and Applications,* 11/2005; Vol.14, Issue.4, 337-344.

[6]. A.G. Williamson, "Refining a neural network credit application vetting system with a genetic algorithm," *Journal of Microcomputer Applications*, Vol. 18, Issue 3, July 1995, 261-277.

[7]. D.B. Fogel, "An Introduction to Simulated Evolutionary Optimization," *IEEE transductions on Neural Networks*, Vol. 5, No. 1, 1994.

[8]. H. Oktem, T. Erzurumlu, and F. Erzincanli, "Prediction of minimum surface roughness in end milling mold parts using neural network and genetic algorithm," *Materials and Design*, Vol. 27, 735–744, 2006.

[9]. S. Sette, L. Boullart, and L. V. Langenhove, "Optimising a Production Process by a Neural Network/Genetic Algorithm Approach," *Engineering Applications in Artificial Intelligence*. Vol. 9, No. 6, 681-589, 1996.

[10]. M. Bhattacharya "Evolutionary Approaches to Expensive Optimisation," *International Journal of Advanced Research in Artificial Intelligence*, Vol. 2, No. 3, 53-59, 2013

[11]. J. Yu, X. Chen, T.R. Hung and F. Thibault, "Optimization of Extrusion Blow Molding Processes Using Soft Computing and Taguchi Method", *Journal of Intelligent Manufacturing,* 15, 2004, 625-634.

[12]. T. Hanai, N. Iwata, T. Furuhashi, H. Honda and T. Kobayashi, "Proposal of Reliability Index in Search for Reliable Solution of Reverse Calculation Based on Fuzzy Neural Network Modeling", *Journal of Chemical Engineering of Japan*, Vol. 37, No. 4, 523-530, 2004.

[13]. D.-C. Ko, D.-H. Kim, B.-M. Kim, Application of artificial neural network and Taguchi method to preform design in metal forming considering workability, *International Journal of Machine Tools and Manufacture*, Vol. 39, Issue 5, 1999, 771-785.

[14]. D. Rosato and D. Rosato, *Blow Molding Handbook*, Hanser Publisher, Munich, 1989.

[15]. R. W. Diraddo and A. Garcia-Rejon, "On-Line Prediction of Final Part Dimensions in Blow Molding: A Neural Network Computing Approach", *Polymer Engineering and Science*, Vol. 33, No. 11, 1993, 653-664.

[16]. BlowSim, 6.0, Industrial Materials Institute, National Research Council Canada 1999.

[17]. J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-Parameter Optimization with Differential Evolution," *IEEE Congress on Evolutionary Computation,* Vol. 1, 2005, 506-513.