Design Optimization of Extrusion-Blow-Molded Parts Using Prediction-Reliability-Guided Search of Evolving Network Modeling

Jyh-Cheng Yu,[1] Jyh-Yeong Juang[2]

[1]*Department of Mechanical and Automation Engineering, National Kaohsiung First University of Science and Technology, 2 Juoyue Road, Nantz District, Kaohsiung 811, Taiwan, Republic of China*
[2]*Department of Mechanical Engineering, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei 106, Taiwan, Republic of China*

Received 29 March 2009; accepted 13 December 2009
DOI 10.1002/app.31954
Published online 2 March 2010 in Wiley InterScience (www.interscience.wiley.com).

ABSTRACT: Industries often adopt a two-stage design for blow-molded parts. The part thickness distribution is first determined by structural analysis to satisfy loading requirements, and this is followed by programming of the die-gap opening to realize the thickness distribution. This study proposes a soft-computing-based optimization scheme integrating part design and molding process control to search for the die-gap programming of the molding process with minimum part weight while satisfying performance constraints. Finite element analysis tools are applied to simulate the extrusion-blow-molding process and structural analysis. To reduce the number of simulations, the proposed scheme first establishes a neural network (NN) model from a small experimental design to simulate the system response, and it searches for the model optimum with a genetic algorithm (GA). Because the prediction generality of an NN from small training samples will be limited, this work proposes fuzzy reasoning for the prediction reliability of the model to guide the GA search for a quasi-optimum. The verification of the optimum is added to retrain the model, and the process iterates until convergence is reached. The iteration automatically distributes additional samples in the most probable space of the design optimum for the evolving model and improves the sampling efficiency. A high-density polyethylene bottle design is presented to illustrate the application and for comparison with the Taguchi method and a simple iteration of NN and GA. The proposed scheme outperforms the other two and provides a feasible optimum from a robust convergence. © 2010 Wiley Periodicals, Inc. J Appl Polym Sci 117: 222–234, 2010

Key words: computer modeling; molding; simulations; stress

INTRODUCTION

Typical extrusion-blow-molded parts involve two design phases. The part thickness distribution is first determined by structural analysis to satisfy the loading requirements, and this is followed by the control of the extrusion-molding process to realize the thickness distribution. Recent advances in numerical tools have proven their advantages in the applications of structural analysis and process simulation. Design verification using the simulation tools requires lower costs and less time than conventional trial-and-error experiments. Performance optimization for blow-molding parts then becomes feasible: a design with minimum part weight can be sought while the mechanical constraints are satisfied. However, because of the complexity of numerical simulation, a streamlined design procedure with high searching efficiency is still important.

Extrusion blow molding involves four processes: parison extrusion, mold clamping, parison inflation, and part solidification. First, parison extrusion produces a molten thermoplastic tube coming from the die. The parison shape is determined by the die geometry, die-gap programming, and flow rate. The parison is then clamped, and high-pressure air is blown into it to obtain the final part. Finite element tools such as BlowSim (developed by the National Research Council of Canada) provide an integrated simulation for parison-extrusion and blow-molding processes to obtain the final thickness distribution of the inflated part.[1,2]

By manipulation of the die-gap opening over time, the parison profile can be controlled. Clearly, there is a direct relationship between the parison thickness and the inflated part thickness. The parison thickness profile is critical because it determine part performance (e.g., the load resistance and part weight).

Correspondence to: J.-C. Yu (jcyu@ccms.nkfust.edu.tw).
Contract grant sponsor: National Science Council of the Republic of China; contract grant number: NSC91-2212-E-155-007.

© 2010 Wiley Periodicals, Inc.

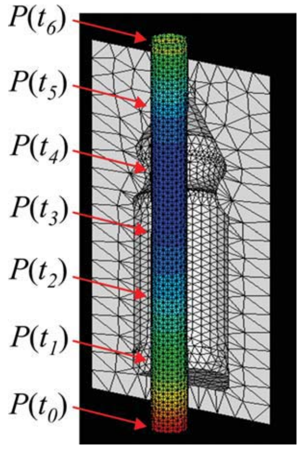

Figure 1 Exemplar programming points of the parison extrusion of bottles. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The main goal of parison programming is to control the die-gap openings to obtain the desired thickness distribution in the final parts.[3] The programming points are then used to specify the die-gap openings of the parison in the extruder as a function of time. For the bottle example in Figure 1, the die-gap openings at seven discrete extrusion times—$P(t_0)$, $P(t_1)$, $P(t_2)$, $P(t_3)$, $P(t_4)$, $P(t_5)$, and $P(t_6)$—are identified as the design variables.

Higher material efficiency will lead to a lighter part. A uniform wall thickness design for the final inflated part may lead to overdesign for unloaded sections and underdesign for critical loading areas if the parts are subjected to mechanical loads such as impact and internal pressurization. A uniform thickness distribution will not guarantee optimum performance. An optimum part thickness profile has to satisfy the requirement of mechanical strength with minimum part weight. Consequently, the problem can be converted to the determination of the die-gap-opening profile of the extruder such that the weight of the final blown part is minimized; this is subject to the constraint that the Von Mises stresses of the part under test loads should not exceed the material yield stress.[4]

Often, the optimization process is conducted in two stages.[5] Performance optimization uses a gradient-based technique to determine the minimum part thickness distribution that satisfies the stress constraint, and this is followed by the process optimization phase, which is used to determine the optimal die-gap-opening profile that minimizes the part weight, which is subject to the minimum thickness constraint derived from the performance optimization.[6,7] The minimum thickness constraint of each controlling point is determined by the retention of the maximum thickness from the individual test load. However, the stress of each element is not only a function of the local thickness. A part satisfying the minimum thickness constraint may not guarantee the satisfaction of the stress constraint.

Many studies have addressed the optimization of parison programming to achieve the required thickness distribution of blown parts. The searching efficiency becomes an important issue for time-consuming simulations and expensive experiments such as blow molding. The Taguchi method[8] is well known for its efficiency and simplicity in parameter design. Inspired by statistical factorial experiments, the Taguchi method features orthogonal arrays and analysis of the mean (ANOM) to analyze the effects of design variables. Each variable is assumed to have finite levels (set points), such as two or three levels, within the investigated range. The orthogonal array is a type of fractional factorial experiment. The application of orthogonal arrays reduces the number of experiments, and this is particularly effective for design optimization involving expensive experiments. An ANOM study of experimental results reveals the effects of design parameters, and these are used to determine the optimal level of each parameter.[9] However, the prediction of the optimal design is sensitive to the selection of factorial levels and interaction effects. Also, the restriction of parameter values to factorial levels reduces the possibility of having better designs between preset levels.

Genetic algorithms (GAs) apply the evolutionary principles found in nature to the problem of finding an optimal solution,[10–12] and they are popular for solving complex engineering problems. GAs use a selection operator to avoid trapping at a local optimum, which often happens in classical optimizations, when a better optimum can be found outside the vicinity of the current solution. A lot of modifications of the methodology have been proposed since the concept was first raised in 1975. Among them, competent GAs[13] claim to find a global or nearly global solution in a reasonable time. Banier and Brisset[14] introduced a GA mixed with constraint satisfaction problem (CSP) techniques. The approach is designed for combinatorial problems whose search spaces are too large and/or objective functions that are too complex for the usual CSP techniques and whose constraints are too complex for conventional GAs. The main idea is the handling of subdomains of the CSP variables by a GA. By combining the achievements of genetic and evolutionary computation with the advanced methods of machine learning and probabilistic modeling, the Bayesian optimization algorithm is capable of solving problems decomposable into subproblems of bounded order quickly, accurately, and reliably.[15]

The integration of trained network models and a searching algorithm becomes attractive for engineering optimization. The numerical network model replaces the exact engineering system during the optimum search to reduce experimental costs.[16] There are two types of integration in terms of the modeling strategy. One aims to establish a simulating model with accurate generality for the engineering system in the first place. A searching algorithm is applied to search for the optimum in the simulated model instead of interacting with the actual engineering system.[17,18] However, a great number of training samples are often required to establish an accurate simulating model, and this is not cost-realistic in engineering applications. Also, the training accuracy varies with the complexity of the problems, and no universal strategy guarantees prediction generality. Some others start from a network model from smaller training samples. Although modeling imperfection is to be expected, additional training samples apply only to the space of interest to reduce the sampling cost. Here, the searched optimum from the imperfect model serves as an additional training sample.[19,20] Therefore, the training and searching processes iterate to improve the network modeling gradually, especially in the probable space of the design optimum.

Sampling efficiency is important for the network modeling of applications with a high sampling cost. The selection of well-designed experiments such as Taguchi's orthogonal arrays as training samples could balance sampling cost and prediction accuracy. Some studies have applied Taguchi's orthogonal arrays as training samples for a neural network (NN) model and searched for the optimum with GAs.[21–23] Although the use of orthogonal arrays as training samples reduces the sampling cost, limited learning samples may greatly diminish the prediction generality of the trained network model for complex systems such as extrusion blow molding. A lower sampling cost is traded for lower prediction generality. Previous studies have often overlooked the possibility of the lack of prediction generality for a network model because of deficient training samples, and an unbounded search for the optimum in the feasible domain of a network model might lead to erroneous results. Even if the confirmation result of the search optimum is used to retrain the network model, the iteration often take a long time to converge. The reliable prediction of such a network model from deficient samples is likely restricted to the neighboring space of training samples. A guided search in an evolving network model would increase searching reliability and sampling efficiency.

This study proposes a novel optimization scheme integrating part design and molding process control. The soft-computing-based optimization scheme searches for the die programming of the molding process with minimum part weight while satisfying the performance constraints. The design objective is to search for a feasible stress distribution with a minimum deviation of the allowable material stress via the manipulation of the die-gap opening at designated programming points. To balance the simulation cost and the prediction accuracy, the study applies an evolving modeling and optimization strategy to increase the sampling efficiency. Two finite element programs, BlowSim and ANSYS, are introduced to simulate the thickness distribution of the extrusion-blow-molding processes and to perform the structural analysis under test loads. A bottle design is presented to illustrate the proposed method.

OPTIMIZATION STRATEGY

The proposed optimization strategy, prediction-reliability-guided search of evolving network (PREGSEN) modeling, first establishes an NN from a small experimental design and searches for the optimum of the trained model with a GA. To cope with a possible deficiency of prediction generality due to small learning samples, the strategy introduces fuzzy prediction reliability to direct the evolution decision in the GA and increase the evolving priority surrounding training samples. The verification experiment of the derived optimum from the GA search is then introduced to the learning samples to retrain and evolve the network model. Therefore, only one additional interaction with the actual engineering system is required in each iteration. The training and searching processes iterate until the optimum convergence. A flowchart of the proposed optimization strategy is illustrated in Figure 2.

Evolving NN model

NN technologies are effective in establishing a simulation model from sampling data for engineering systems. A back propagation network (BPN) is a type of supervised learning network and the most widely used network model.[24] Previous researchers[2,16] have proposed a prediction model for extrusion-blow-molding applications using a BPN from extensive experimental data. However, this study applies a BPN to establish a rough network model from a small number of training samples only for the purpose of optimum search. Often, the prediction accuracy of the network model will be closely related to the number of training samples. For an engineering application with expensive experimental costs, the number of training samples will be limited, and this will greatly affect the generality of the prediction model. In light of the limited prediction ability, the search of the neighboring regions surrounding the training samples is more reliable but will provide only a quasi-optimum. The verification

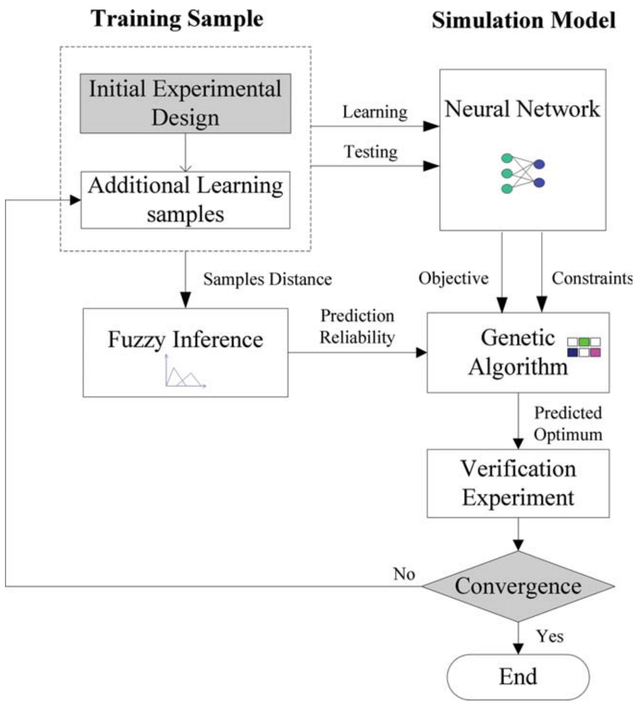

Figure 2 Optimization flowchart of PREGSEN. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

of the optimum will be applied to retrain the network. Therefore, the prediction accuracy of the model will improve in an evolving fashion, especially for the most probable region of the design optimum, to increase the sampling efficiency.

The proposed BPN model consists of a typical three-layer structure, namely, an input layer, a hidden layer, and an output layer. In this study, Taguchi's orthogonal arrays are suggested for the design of training samples to reduce the number of experiments, and this is particularly effective for design optimization involving expensive experiments or time-consuming simulations. The control variables are factorized in the preliminary investigated range. A minimal three-level orthogonal array is used for the learning samples, and a minimal two-level orthogonal array distributed in the middle of the variable range is used for the testing samples. Learning samples are used to determine the weighting matrices among neurons, and testing samples are used to determine the accuracy and generality of the network.

Extrapolation distance (ED)

For an NN trained from a limited number of training sampling, the reliability of the model might be restricted to the neighboring space of the learning samples, particularly for a complex system. Experience tells us that the prediction accuracy of the model is getting worse if the predicted design is far away from the training samples. The mean Euclid distance (r_{ij}) between the predictive designs (D_i) and the sample data (S_j) is defined as follows:

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

where $D_i = [d_{i1}, d_{i2}, \ldots, d_{in}]$, $S_j = [s_{j1}, s_{j2}, \ldots, s_{jn}]$, and n represents the number of variables.

As a rule of thumb, the prediction accuracy for the interpolating designs of an NN model is better than that for the extrapolating designs. Also, the closer the predictive design is to the training samples, the higher the prediction accuracy is. This study proposes the ED as a neighboring index of a predictive design, which is defined as the minimum mean Euclid distance between the prediction and the training samples:

$$ED_i = \min(r_{ij}) \quad (2)$$

To facilitate the calculation of the distance between designs, the values of the continuous variable (x_k) are normalized to z_k with the following transformation:

$$z_k = \frac{\left\{x_k - \frac{[\text{Maximum}(x_k) + \text{Minimum}(x_k)]}{2}\right\}}{\frac{[\text{Maximum}(x_k) + \text{Minimum}(x_k)]}{2}} \quad (3)$$

where Maximum(x_k) and Minimum(x_k) represent the maximal and minimal values of design variable x_k in the training samples, respectively. For discrete variables, the factorial values are assigned with equal spacing between -1 and $+1$.

The interpolating designs often have higher prediction accuracy than the extrapolating designs in NN models. This study defines the smallest convex hyperpolyhedron surrounding all training samples as the sampling enclosure space (SES), which is used to differentiate interpolation designs and extrapolation designs in a multidimensional simulation model. The SES boundary is constructed by a set of n-dimensional hyperplanes, which are determined by n noncoplanar points from the training samples. If the prediction point is inside or on the boundary of the SES, it is an interpolating design; otherwise, the prediction point is an extrapolating design. A two-dimensional example is shown in Figure 3, where D_1 is an interpolating design and D_2 is an extrapolating design.

The training samples are represented as normalized coordinates Z_1 and Z_2 to calculate the EDs for predicting designs. The ED is assumed to be positive for an extrapolating design and negative for an interpolating design. For instance, an NN is trained

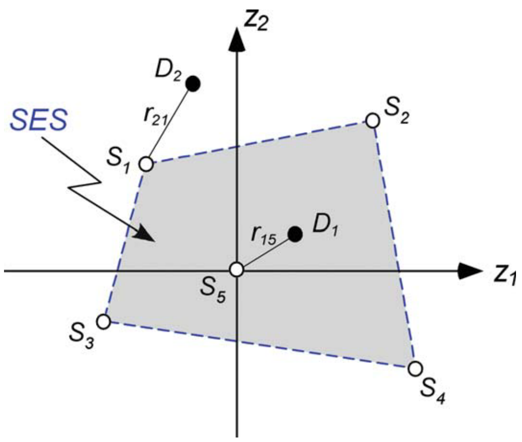

Figure 3 EDs of predicted designs for a two-dimensional example. Z_1 and Z_2 represent the horizontal coordinate and the vertical coordinate for two-dimensional problem. Z_1 is normalized of the input variable ($X1_l$) between -1 and $+1$, ($l = 1 \sim n$). Z_2 is normalized of the input variable ($X2_l$) between -1 and $+1$, ($l = 1 \sim n$). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

from five samples (S_1–S_5), as shown in Figure 3. The ED of the interpolating design is designated as $-r_{15}$ because r_{15} is the shortest mean Euclid distance among r_{1i} (i = 1–5). Likewise, the ED of the extrapolating design is $+r_{21}$. An interpolating design with a small ED is expected to have better prediction accuracy, and for an extrapolating design with a large ED, the prediction accuracy is likely doubtful.

Fuzzy reasoning of the prediction reliability

Fuzzy systems are widely used in engineering applications to convert expert knowledge into a mathematical reasoning model. Typical fuzzy systems consist of a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier.[12] The fuzzifier converts the input data into linguistic fuzzy variables. The expert's reasoning is then expressed as a set of fuzzy conditional statements based on the fuzzy variables. The decision can be reasoned from the fuzzy inference engine, and this is followed by the defuzzifier for conversion of the linguistic conclusion into crisp output.

Here a fuzzy model is proposed to determine the prediction reliability. The prediction reliability of the network model will be related to the ED of a predictive design. The association of prediction reliability and EDs is based on two fuzzy concepts. One is to assign less reliability to the prediction point with more distance from the learning samples, and the other is to assign less reliability to extrapolating designs than interpolating designs. Seven single-input, single-output inference rules are proposed that are based on the nature of the simulated models:

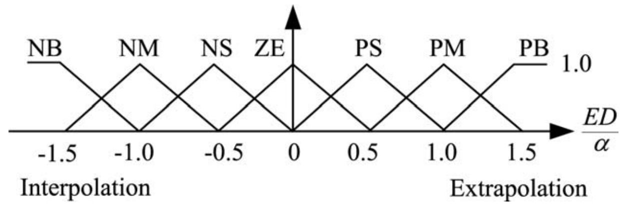

Figure 4 Membership functions of the condition levels of ED.

- R1: If the ED of the design is positive big (PB), prediction reliability is bad.
- R2: If the ED of the design is positive medium (PM), prediction reliability is poor.
- R3: If the ED of the design is positive small (PS), prediction reliability is fair.
- R4: If the ED of the design is zero (ZE), prediction reliability is excellent.
- R5: If the ED of the design is negative small (NS), prediction reliability is good.
- R6: If the ED of the design is negative medium (NM), prediction reliability is fair.
- R7: If the ED of the design is negative big (NB), prediction reliability is poor.

Seven linguistic levels are defined to describe the condition variable ED: PB, PM, PS, ZE, NS, NM, and NB. Five levels are defined to describe the assessment results for the prediction reliability: excellent, good, fair, poor, and bad. Because small orthogonal arrays are used for the training samples, the maximum ED (α) can be approximated with random sampling in the preliminary variable range and used in the definition of the membership function for the linguistic levels of ED. Standard membership functions associated with these statements are illustrated in Figures 4 and 5. A simple center average defuzzifier is applied to derive the prediction reliability. Figure 6 presents a prediction reliability contour plot for a two-dimensional case using the fuzzy inference. The five solid dots in Figure 6 represent the training samples in the simulated network. The fuzzy model can generally represent the intrinsic characteristics of the prediction reliability of a network model.

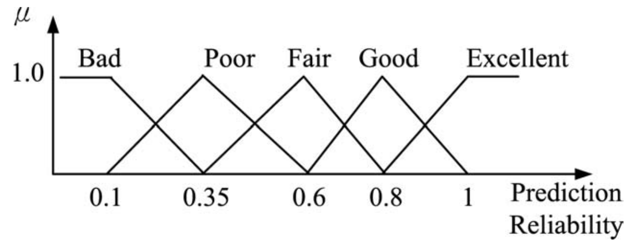

Figure 5 Membership functions of the assessment levels of prediction reliability.

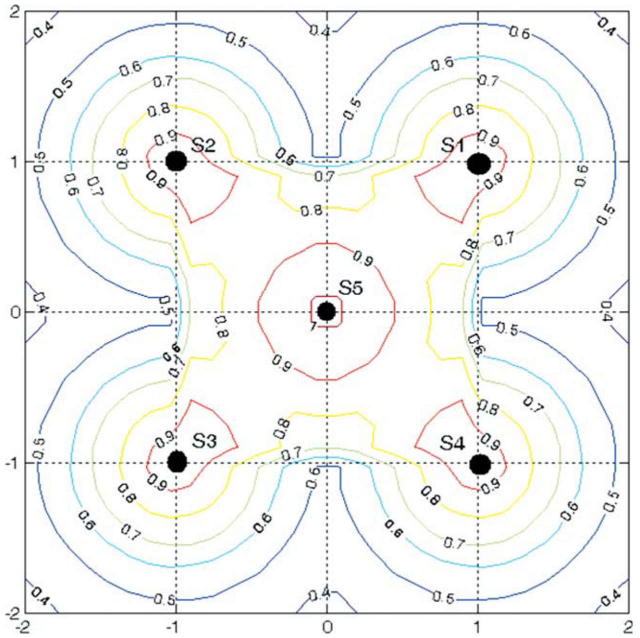

Figure 6 Prediction reliability contour plot for a two-dimensional example with five training samples. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Search for the model optimum using GA

Taking advantage of the fast recall of an NN, a GA is applied to search for the optimum of the trained network model established from engineering problems to reduce experimental costs. GAs are categorized as global search heuristics, and they are capable of searching for a global optimum for a simulated model. Optimizing GA search is not the focus of this study. Any improvement over the searching efficiency of GAs in previous studies can be applied to search for the model optimum. Whether the model optimum is the exact optimum of the engineering system depends on the accuracy of the simulated model. If a perfect network model for the system is available, the searched optimum will be the exact optimum. However, a great number of training samples will be required, and this is not cost-realistic in engineering applications. The prediction generality of a simulated network is limited if the number of training samples is deficient. A unbounded search of the trained network might lead to erroneous results.

Here, the fuzzy inference of the prediction reliability is introduced into the definition of the fitness function to prioritize the searching domains to the neighboring space of the training samples, and this thus ensures the searching reliability. The training samples are assumed to be the initial population in this study. For each generation in the evolution, the designs in the population are sorted and ranked from the best to the worst on the basis of the predicted responses from the simulated network and the prediction reliability from the fuzzy inference. The fitness function is defined as the sum of the response rank and the reliability rank, as shown in Eq. (4). During the evolution processes of mutation, crossover, and reproduction, the design with the higher rank will have an advantage in the evolutionary selection using roulette wheel selection.[10] This definition of the fitness function will ensure the prevailing of a reliable optimum at the end of evolution:

$$\text{Fitness} = \text{Response rank} + \text{Reliability rank} \qquad (4)$$

By a series iteration of selection and reproduction, the GA search will provide a quasi-optimum of the network model. The current model is possibly lacking generality because of scarce training samples. Because of the inclusion of prediction reliability in the fitness function, the GA tends toward a conservative search surrounding training samples with a balance between reliability and optimality. The theoretical optimum of the trained model is not desirable if the optimum is far away from the training samples because of a possibly enormous prediction error. The quasi-optimum, on the other hand, is more reliable even for a deficient simulated model.

Iterative training and search for the design optimum

The verification result of the quasi-optimum will be introduced to the learning samples to retrain the model. Only one verification experiment is required for the optimum obtained from the guided GA search of the evolving network model. Although the guided search using the prediction reliability might restrict the search domain to the neighboring space of training samples, the search space will be modified with the addition of new samples from the verification of the optimum. If the additional learning sample is an extrapolating design, the SES in the reliability inference will expand, and the searching range in the GA will adjust dynamically because of the normalization process and the fuzzy inference.

The proposed algorithm will secure the reliability of the searched optimum in iteration and will evolve the exploration range automatically. Global accuracy of the simulated model is not necessary for the search for the optimum. Instead of increasing sampling points evenly distributed in the investigated range, additional sampling points will congregate in the most probable region of the global optimum with the proposed algorithm. The sampling efficiency will thus increase, and this is particularly important in engineering applications. The training and searching process iterates until the convergence of the predicted optimum is reached. The quasi-optimum will gradually approach the global optimum.

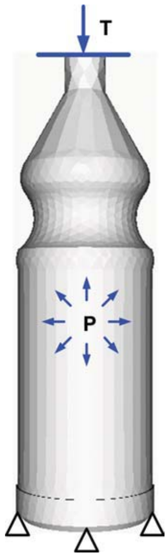

Figure 7 Two mechanical testing loads of the high-density polyethylene bottle: P and T. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The convergence criteria include (1) the convergence of the predicted optimum and the verified result and (2) the variation of the last three searched optima within the engineering tolerance. For engineering practice, a tradeoff between design improvements and experimental costs is a more important concern.

OPTIMIZATION OF BLOW-MOLDING PROCESSING CONDITIONS FOR PERFORMANCE DESIGN

This session presents the application of the proposed optimization strategy to obtain the optimal parameter design of an extrusion-blow-molding process for a high-density polyethylene bottle. Two types of loadings usually used in industrial applications are investigated, and they include internal pressurization at 90 psi and a top displacement of 3.75 mm for 5 s, as illustrated in Figure 7. The maximum allowable stress, corresponding to the ultimate tensile strength of the material, is 33 MPa. For this material, Young's modulus is 879 MPa, and the thickness shrinkage is 5%. The software simulating blow molding, BlowSim, is applied to estimate the thickness distribution of the blown bottle. The finite element analysis software ANSYS is used for the structural analysis of the bottle.

Formulation for performance optimization

The design objective is to obtain a wall thickness distribution of minimal weight by manipulation of the die-gap programming; this is subject to the stress distribution below the allowable level. The initial formulation for this optimization can be represented as follows:

Minimize: Part weight $[P(t_j)]$
Design variable: $P(t_j)$, $j = 0–6$
Constraints: $s_i[P(t_j),P,T] \leqq \sigma_a$

where $P(t_j)$ represents the die-gap openings of the controlling points, as illustrated in Figure 1; s_i represents the stresses of node i; σ_a is the allowable stress of the material; P is the internal pressure load; and T is the top displacement load.

The reduction of an element's thickness results in an increase in its stress level. To increase the material efficiency, the stress distribution should be as close as possible to the allowable stress. The smaller the variance of the stress distribution is, the more closely the mean can be moved toward the material yield strength, and this leads to thinner elements and thus reduces part weight. However, any element's stress exceeding the allowable strength might result in part failure. In this work, the constrained optimization problem is replaced by an unconstrained minimization of the variance of the stress distribution around the allowable stress level and the constraint penalty function, as illustrated in Figure 8.

The modified objective function [Eq. (5)] contains two portions, the quality loss due to the variation of the stress distribution and the penalty loss due to the constraint violation:

$$MOBJ = \frac{\sum_{i=1}^{n} (s_i - \sigma_a)^2}{n} + \sum_{i=1}^{n} < s_i - \sigma_a >^2 \quad (5)$$

where n is the total number of nodes of the simulation model and s_i is the stress of node i.

The quality loss due to the variation of the stress distribution is estimated by the mean squared deviation of

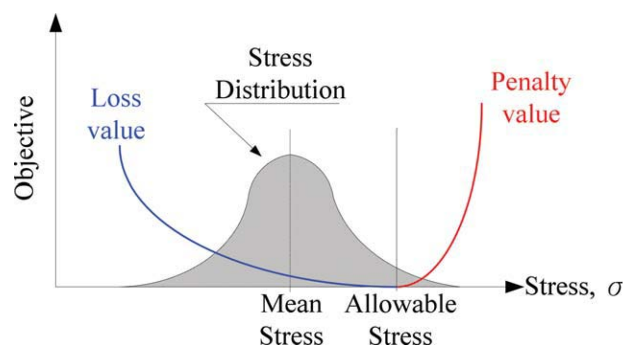

Figure 8 Illustration of the design objective for performance optimization. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

TABLE I
Experimental Design with the L18 Orthogonal Array

L18	A $P(t_0)$	B $P(t_1)$	C $P(t_2)$	D $P(t_3)$	E $P(t_4)$	F $P(t_5)$	G $P(t_6)$	*MOBJ*[a]	S/N
1	55	55	55	55	55	55	55	21,791.25	−43.38
2	55	75	75	75	75	75	75	4,625.35	−36.65
3	55	95	95	95	95	95	95	3,308.56	−35.20
4	75	55	55	75	75	95	95	65,0545.19	−58.13
5	75	75	75	95	95	55	55	214,988.62	−53.32
6	75	95	95	55	55	75	75	5,544.66	−37.44
7	95	55	75	55	95	75	95	64,469.80	−48.09
8	95	75	95	75	55	95	55	6,349.12	−38.03
9	95	95	55	95	75	55	75	119,726.34	−50.78
10	55	55	95	95	75	75	55	1,245,531.06	−60.95
11	55	75	55	55	95	95	75	25,238.52	−44.02
12	55	95	75	75	55	55	95	7,907.71	−38.98
13	75	55	75	95	55	95	75	645,177.17	−58.10
14	75	75	95	55	75	55	95	1,565.06	−31.95
15	75	95	55	75	95	75	55	31,459.45	−44.98
16	95	55	95	75	95	55	75	997,229.07	−59.99
17	95	75	55	95	55	75	95	222,059.20	−53.46
18	95	95	75	55	75	95	55	5,040.66	−37.02
Initial	75	75	75	75	75	75	75	8,229.76	−39.15

[a] *MOBJ*, modified objective function.

the Von Mises stress from the allowable stress. The average quality loss can be reformulated into two parts: the deviation of the mean stress from the allowable stress and the variation of the stress around the mean:

$$\frac{\sum_{i=1}^{n}(s_i-\sigma_a)^2}{n} = (\bar{s}_i-\sigma_a)^2 + \frac{\sum_{i=1}^{n}(s_i-\bar{s}_i)^2}{n} \approx (\bar{s}_i-\sigma_a)^2 + \nu \quad (6)$$

where $\bar{s}_i$ is the mean stress and v is the distribution variance from the structural analysis. Reducing the quality loss leads to a smaller stress distribution and a mean stress closer to the allowable stress.

The second portion of the modified objective function, the penalty loss, is formulated with a second-order singularity function as shown in Eq. (7). This portion accounts for the penalty of the FEM nodes violating the stress constraint:

$$< s_i - \sigma_a >^2 = \begin{cases} 0, & \text{if} \quad s_i \le \sigma_a \\ (s_i-\sigma_a)^2, & \text{if} \quad s_i > \sigma_a \end{cases} \quad (7)$$

The search for the design of the minimum objective function will increase the material efficiency and thus provide a thickness distribution of minimum part weight while satisfying the loading requirements.

Design optimization using the Taguchi method

The Taguchi method applies ANOMs to estimate parameter sensitivities, and it is popular in engineering applications. The die-gap openings at seven discrete extrusion times [$P(t_0)$, $P(t_1)$, $P(t_2)$, $P(t_3)$, $P(t_4)$, $P(t_5)$, and $P(t_6)$] are selected as the design variables to control the parison thickness in seven evenly distributed sections. The initial design adopts a uniform die-gap opening of 75%. Under the assumption of three levels for each design variable, a minimal orthogonal array (L18) is selected as the experimental design (Table I). The factorial levels locate the initial design in the middle of the design space of 55–95% for each opening. The logarithm transformation of the modified objective function will be used as the signal-to-noise ratio (S/N) for Taguchi's parameter design:

$$S/N = -10 \cdot \log(MOBJ) \quad (8)$$

Figure 9 presents an effect plot for the die opening from the experimental design in Table I. A design with a higher S/N ratio has a smaller value of the objective function. Taguchi's parameter design

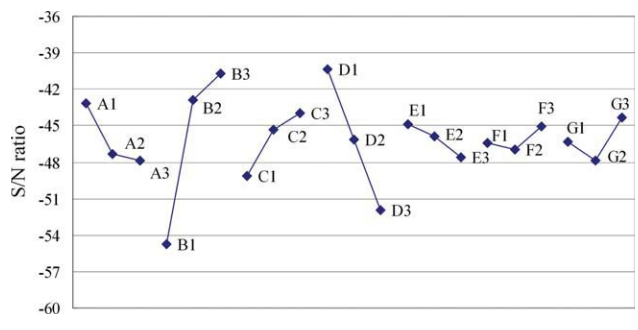

Figure 9 Effect plot for the die opening during the blow molding of the bottle. A: $P(t_0)$; B: $P(t_1)$; C: $P(t_2)$; D: $P(t_3)$; E: $P(t_4)$; F: $P(t_5)$; G: $P(t_6)$. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

TABLE II
Sample Testing with the L8 Orthogonal Array

L8	A $P(t_0)$	B $P(t_1)$	C $P(t_2)$	D $P(t_3)$	E $P(t_4)$	F $P(t_5)$	G $P(t_6)$	$MOBJ$[a]	S/N
1	65	65	65	65	65	65	65	34,842.0	−45.42
2	65	65	65	85	85	85	85	438,815.8	−56.42
3	65	85	85	65	65	85	85	4,051.5	−36.08
4	65	85	85	85	85	65	65	10,328.1	−40.14
5	85	65	85	65	85	65	85	3,880.4	−35.89
6	85	65	85	85	65	85	65	153,796.5	−51.87
7	85	85	65	65	85	85	65	17,050.2	−42.32
8	85	85	65	85	65	65	85	45,926.9	−46.62

[a] *MOBJ*, modified objective function.

scheme suggests the optimum treatment to be $A_1B_3C_3D_1E_1F_3G_3$. The verification result from BlowSim and ANSYS for Taguchi's optimum shows an S/N ratio of −37.29 dB, which is very different from the value of −25.63 dB predicted with Taguchi's additive model. The verified performance of the predicted optimum is not even the best among the design experiments. The parameter design using the Taguchi method fails for possible reasons including the interactions among variables and significant system nonlinearity.

Optimization of the bottle thickness distribution using PREGSEN

Establishing the simulated NN model

Training samples are essential to the prediction quality of network models. The L18 orthogonal array from the previous Taguchi application is used as a learning sample to reduce the number of experiments and to maintain good sample representation. Another two-level orthogonal array (L8), illustrated in Table II, is selected as the testing sample for the network training. The level values (65 and 85%) are set between the three-level values (55, 75, and 95%) of the learning samples.

The steepest gradient method is assumed to train the weighting matrices of the BPN. There is no definite rule available to determine appropriate parameters in the network training. This study applies a simple Taguchi parameter design to determine the number of neurons in the hidden layer, the initial learning rate, the decreased learning rate, and the increased learning rate. Three-level factorial parameters are assumed. The optimal parameter design is derived with an L9 orthogonal array experiment and an ANOM for the optimal training efficiency in the first 10 epochs. The parameter design in this case suggests 19 neurons in the hidden layer, an initial learning rate of 0.5, a decreased learning rate of 0.85, and an increased learning rate of 1.15.

Evolving modeling and optimization

As illustrated in Figure 2, the prediction reliability is introduced into the fitness function of the optimization search using a GA. The anchor parameter α in the member function of Figure 4 is 1.4 for this experimental design of L18 + L8. The training samples are used as the initial population in each epoch. Each sample needs to be encoded by a gene with a binary genetic algorithm. In this study, the bit length of the encoded chromosome is assumed to be 12. Because the design variables have been normalized with Eq. (3), the searching boundary in GA is set to

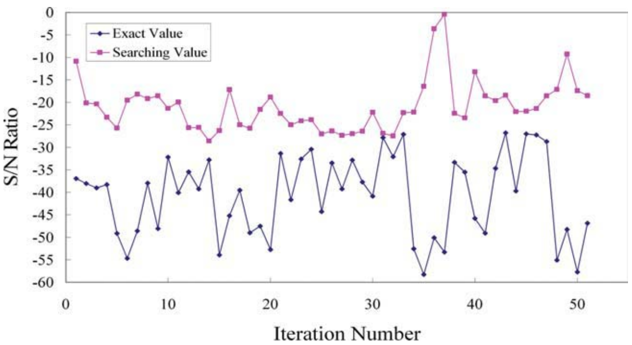

Figure 10 Iteration result for a simple recursion of NNGA. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

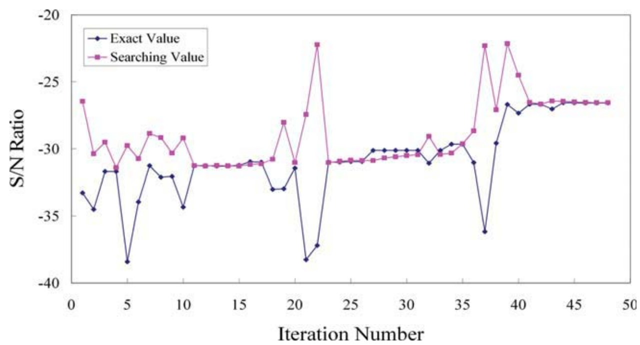

Figure 11 Iteration result with PREGSEN. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

TABLE III
Comparison of Different Iterations

Iteration number	Predicted objective	Quality loss	Penalty loss	Verified objective	Prediction error[a]	COV[b]
15	1349.7	467.3	861.7	1329.0	20.7	0.0079
25	1222.4	470.9	779.0	1249.8	27.4	0.0055
48	452.7	453.9	0.0	453.9	3.2	0.00005

[a] Difference between the predicted objective and the verified result.
[b] Standard deviation divided by the mean of the last three searched objectives.

be ±1.5 to explore a possible optimum outside the preliminary design space. The simulated NN model will then provide the response estimation for each chromosome combination.

The probability of crossover should have a larger value; typically, the probability of crossover ranges from 0.5 to 1.0. A single-point crossover and mutation have been used in this study. The mutation operator must be used with a low probability; typically, the mutation probability ranges from 0.01 to 0.1. Again, the parameters of the GA have been obtained with Taguchi's parameter design. In the GA search of the evolving network model, the initial population size of 26, the crossover rate of 0.8, the mutation rate of 0.1, the optimization tolerance of 0.01, the maximum generations of 300, and the elitist strategy[25] are used.

If the prediction reliability of the current network model is not considered, the iteration is a simple iteration of the neural network and genetic algorithm (NNGA). The GA will assume global accuracy in the investigated range and search for a design with the best performance in the current simulated model. The optimum derived from the GA search might be different from the actual optimum of the engineering system because of the imperfection of the current simulated model. The searched optimum is verified with BlowSim analysis, and the verification design is added to the previous learning samples to retrain the network model. The iteration process of this conventional NNGA is shown in Figure 10. The result shows a continuous discrepancy between the predicted optimum and the verified results due to the lack of sufficient generality for a simulated network from limited training samples. The iteration process had not yet shown convergence after 51 iterations.

Next, the proposed algorithm, PREGSEN, is applied to the same problem. The fitness function is modified with fuzzy prediction reliability. As illustrated in Figure 11, although the verified value and the quasi-optimum of the initial network model obtained with the prediction-reliability-guided GA search are still different because of the lack of generality of the initial model, the difference is greatly reduced because a more reliable quasi-optimum is provided by the reliability-guided search. With the addition of the quasi-optimum to the learning samples to retrain the evolving network model, the iteration quickly converges.

The convergence criteria are defined as follows: (1) the prediction error of the objective is less than 5, and (2) the coefficient of variation (COV) of the last three searched optima is less than 0.001. Although the iteration result seems to converge at iterations 15 and 25, as shown in Figure 11, there is a significant constraint violation as we examine their corresponding objectives (Table III). The penalty loss shows that the stresses of some finite element nodes exceed the allowable stress of the bottle material, and this might result in part failure. Also, they have not met the convergence criteria. Both criteria are reached at iteration 48. There is no constraint violation. The prediction error is 3.2, and the COV is about 0.00005. The optimum die-gap opening is listed in Table IV.

Comparison of the results

This session compares the optimization results from the proposed method with those from the Taguchi method and a simple iteration of NNGA in terms of the design feasibility, part weight, and searching efficiency. Figure 12 presents the profiles of optimal die-gap openings of parison programming, and Table V shows the stress distributions under test loads for the initial design and the optima obtained with various methods. Taguchi's ANOM approach is liable to parameter interactions and system nonlinearity and fails to find a design lighter than the initial design. The optimum from the Taguchi method has a larger distribution and a smaller mean stress, and this results in poor material efficiency and still a strong violation of the stress constraint.

TABLE IV
PREGSEN's Optimum

	$P(t_0)$	$P(t_1)$	$P(t_2)$	$P(t_3)$	$P(t_4)$	$P(t_5)$	$P(t_6)$	Objective	S/N
PREGSEN's optimum	69.0	73.2	97.0	57.9	79.8	28.4	96.5	453.9	−26.57

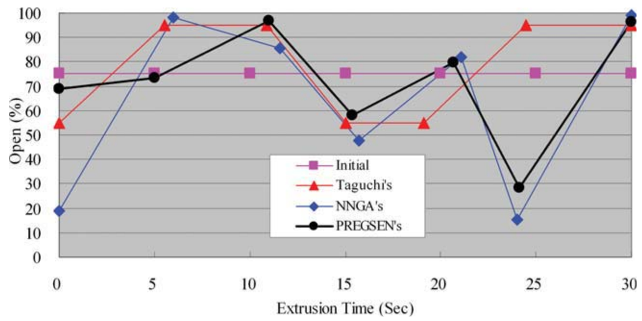

Figure 12 Die-gap openings for various optimal designs. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The iteration result for a conventional recursion using NNGA shows a continuous discrepancy between the predicted optimum and the verified results, as shown in Figure 10, and this is due to the lack of sufficient generality for a simulated network from limited training samples. The iteration process had not yet shown any convergent tendency after 51 iterations. Although the current optimum from the conventional NNGA seems to have a lighter weight, the design is infeasible because of the constraint violation.

PREGSEN provides a quite reliable and efficient search, as shown in Table V. The definition of the fitness function will suppress the exploration of the regions far away from the current training points, and even the prediction from the current network model looks promising. However, the exploration range of the GA will grow dynamically with the addition of new training samples from the verification of the predicted optimum. PREGSEN reached the optimum convergence at iteration 48. PREGSEN's optimum exhibits the smallest stress deviation and leads to a design with a weight of 114.31 g while satisfying the stress constraints. Figure 13 presents a comparison of the stress distribution under test loads with ANSYS and shows that PREGSEN's optimum has the most even stress distributions among the various designs.

CONCLUSIONS

This study presents an integration strategy for the part design and process control of extrusion-blow-molding parts. Subject to mechanical constraints, the strategy minimizes the part weight and provides the optimum die-gap programming in one optimization process. The material efficiency in terms of the stress distribution from the structural analysis of the predicted thickness profile of the bottle is used as the design objective. The mechanical constraints are embedded into the design objective with a penalty function to ensure design feasibility. The search for the optimum die-gap programming of the extrusion-blow-molding process then provides a feasible design with minimum part weight. A case study of a bottle design has been presented, and the comparison results show that the proposed strategy is capable of minimizing the part weight without violation of mechanical constraints with robust searching reliability.

Finally, the searching scheme, PREGSEN, is an evolving network model that starts from a small number of training samples with Taguchi's orthogonal array and selectively evolves for the most probable space of the design optimum to increase the sampling efficiency. For complex simulation systems such as the finite element analysis of structure mechanics and extrusion-blow-molding processes, the number of engineering simulations will greatly affect the optimization cost. However, generality imperfection is inevitable for a simulated model from small training samples, even though great endeavors are applied to the training of NNs.

The prediction reliability of the network model is likely restricted to the surroundings of the learning samples. The accuracy of the extrapolating prediction depends on the model's complexity. If the model is nearly linear, the extrapolating prediction will be pretty accurate. As the model nonlinearity increases, the extrapolating accuracy decreases because of an unknown trend outside the data range, especially for designs farther away from the data range. Although extrapolating designs are less reliable, ruling out possible better designs outside the range of current learning samples is not desirable for an optimum search. The proposed optimization scheme applies the fuzzy reasoning of prediction reliability to the evolving network model to guide the GA search for a reliable quasi-optimum instead of a false optimum of the imperfect network

TABLE V
Comparison of Various Optima

	Mean stress	Standard deviation stress	Quality loss	Penalty loss	Objective	Weight (g)
Initial	13.95	8.78	439.8	7789.9	8229.7	118.7
Taguchi	12.84	9.36	494.1	4868.2	25362.3	119.2
NNGA	12.76	7.39	464.3	14.0	478.3[a]	110.5[a]
PREGSEN	12.62	6.21	453.9	0.0	453.9	114.3

[a] This had not yet converged at the iteration of 51. The listed result is the best design so far.

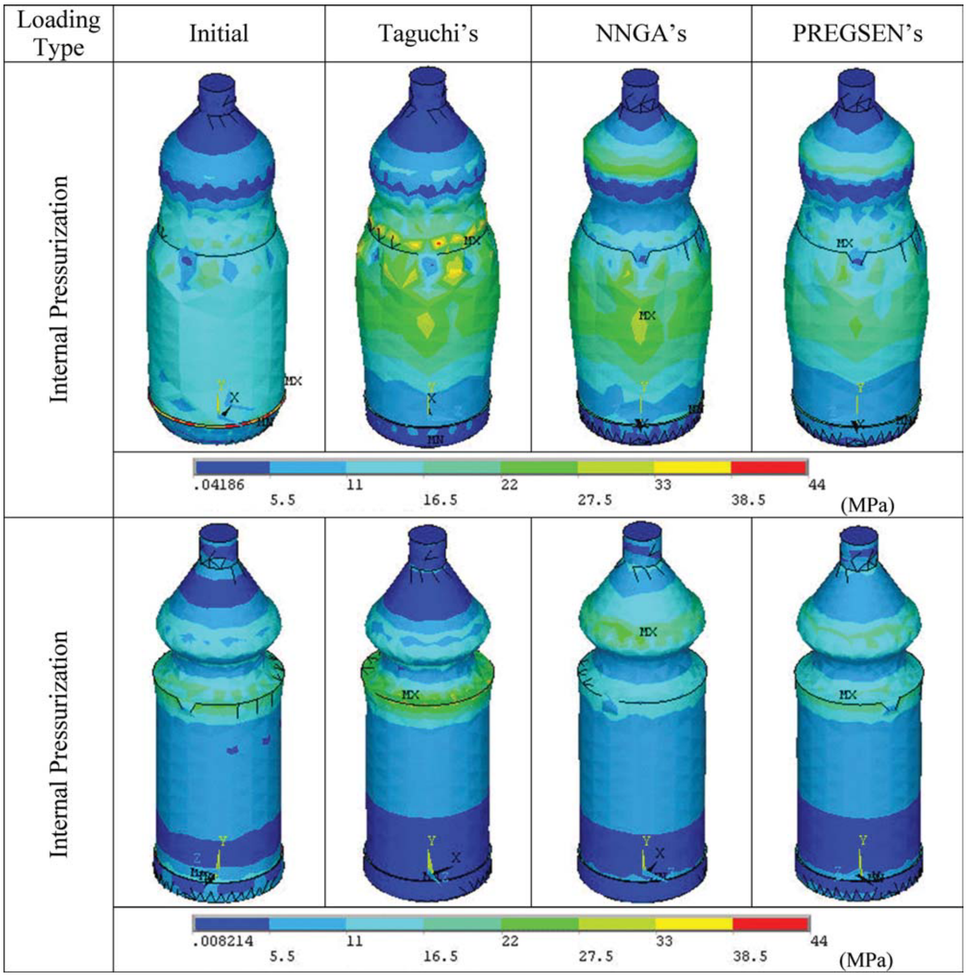

Figure 13 Comparison of the stress distribution under test loads. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

model. The methodology aims to balance reliability and optimality. Verification of the provided optimum will be added to the learning samples to retrain the network model. If the predicted optimum is interpolated, the verification refines the regional accuracy of the network model to further approach the actual peak. If the predicted optimum is extrapolated, the verification suggests additional information to explore the probable region of the optimum and modifies the current network model. The searching and retraining processes iterate until the convergence of the search result. The illustrated example shows a stable and efficient iteration process and demonstrates the merit of the proposed method.

The authors thank F. Thibault for his consultation in the simulation using BlowSim.

References

1. Huang, H. X.; Lu, S. J Reinforced Plast Compos 2005, 24, 1025.
2. Diraddo, R. W.; Garcia-Rejon, A. Polym Eng Sci 1993, 33, 653.
3. Laroche, D.; Kabanemi, K.; Pecora, L.; Diraddo, R. Polym Eng Sci 1999, 39, 1223.
4. Yu, J.; Hung, T. R.; Thibault, F. Proc ASME Des Eng Tech Conf 2002, 2, 133.
5. Gauvin, C.; Thibault, F.; Laroche, D. Polym Eng Sci 2003, 43, 1407.
6. Lee, D. K.; Soh, S. K. Polym Eng Sci 1996, 36, 1513.
7. Hsu, Y. L.; Liu, T. C.; Thibault, F.; Lanctot, B. J Eng Manufacture Proc Instrum Mech Eng Part B 2004, 218, 197.
8. Taguchi, G. Int J Prod Res 1978, 16, 521.
9. Tahboub, K. K.; Rawabdeh, I. A. J Qual Maintenance Eng 2004, 10, 47.
10. Holland, J. H. Adaptation in Natural and Artificial Systems; Ann Arbor, MI: University of Michigan Press, 1975.
11. Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning; Addison-Wesley: Reading, MA, 1989.
12. Jang, J. S.; Sun, C. T.; Mizutani, E. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence; Prentice-Hall: Upper Saddle River, NJ, 1997.
13. Santarelli, S.; Yu, T. L.; Goldberg, D.; Altshuler, E.; O'Donnell, T.; Southall, H.; Mailloux, R. Math Comput Model 2006, 43, 990.
14. Barnier, N.; Brisset, P. In Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, 1998; p 645.

15. Pelikan, M.; Sastry, K.; Goldberg, D. Int J Approximate Reasoning 2002, 31, 221.
16. Huang, H. X.; Lu, S. J Appl Polym Sci 2005, 96, 2230.
17. Oktem, H.; Erzurumlu, T.; Erzincanli, F. Mater Des 2006, 27, 735.
18. Wang, L. Appl Math Comput 2005, 170, 1329.
19. Yu, J.; Chen, X.; Hung, T. R.; Thibault, F. J Intell Manufacturing 2004, 15, 625.
20. Hsu, Y.; Dong, Y.; Hsu, M. JSME Int J Ser C 2001, 44, 103.
21. Su, C. T.; Chiu, C. C.; Chang, H. H. Int J Ind Eng: Theory Appl Practice 2000, 7, 224.
22. Singh, A. K.; Panda, S. S.; Chakraborty, D.; Pal, S. K. Int J Adv Manufacturing Technol 2006, 28, 456.
23. Sanjari, M.; Taheri, K.; Movahedi, R. Int J Adv Manufacturing Technol 2009, 40, 776.
24. Cybenko, G. Math Control Signals Syst 1989, 2, 303.
25. Thierens, D.; Goldberg, D. Int Conf Evol Comput 1994, 1, 508.