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 Rd., Nantz District, Kaohsiung 811, Taiwan, Republic of China*
[2]*Department of Mechanical Engineering, National Taiwan University of Science and Technology, 43, Keelung Rd., Setion. 4, Taipei, 106, Taiwan, Republic of China*

Received 29 March 2009; accepted 13 December 2009
The original publication version is available at DOI 10.1002/app.31954
Journal of Applied Polymer Science, Vol.117, issue 1, pp.222-, 234, 2010

ABSTRACT: Industries often adopt a two-stage design for blow molded parts. The part thickness distribution is first determined by structure analysis to satisfy loading requirements followed by a programming of 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 a minimum part weight while satisfying performance constraints. Finite element analysis tools are applied to simulate the extrusion blow molding process and structure analysis. To reduce the number of simulation, the proposed scheme first establishes a neural network (NN) model from a small experimental design to simulate the system response, and searches for the model optimum using genetic algorithm (GA). Since the prediction generality of a NN from small training samples will be limited, this work proposes a 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 the reach of convergence. 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 HDPE bottle design is presented to illustrate the application, and to compare with 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. Wiley Periodicals, Inc. J Appl Polym Sci ,2010

Key words: Extruded Blow Molding, Soft Computing, Neural Network, Genetic Algorithm, Evolutionary Optimization

1. INTRODUCTION

Typical extrusion blow molding parts involves two design phases. The part thickness distribution is first determined by structure analysis to satisfy the loading requirements followed by the control of extrusion molding process to realize the thickness distribution. Recent advances in numerical tools have proven their advantages in the applications of structure analysis and process simulation. Design verification using the simulation tools requires lower cost and shorter time than conventional trial-and-error experiments. Performance optimization of blow molding parts then becomes feasible to search for a design with minimum part weight while satisfying the mechanical constraints. However, due to the complexity of numerical simulation, a streamlined design procedure with high searching efficient is still important.

The extrusion blow molding involves four processes: parison extrusion, mold clamping, parison inflation, and part solidification. First, the parison extrusion produces a molten thermoplastic tube coming out 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 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 manipulating 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 since it determine the part

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.

Journal of Applied Polymer Science, Vol. 117, 222 – 234, (2010)
Wiley Periodicals, Inc.

performance such as load resistance and part weight. The main goal of the parison programming is to control the die gap openings to obtain the desired thickness distribution of 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 Fig. 1, the die gap openings at 7 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.

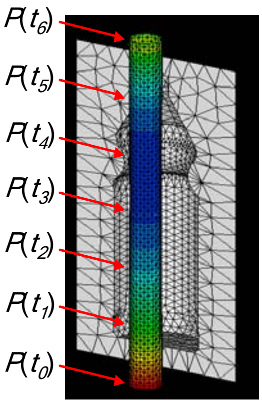


Fig. 1 Exemplar programming points of the parison extrusion of bottles.

Higher material efficiency will lead to a lighter part. A uniform wall thickness design for the final inflated part may lead to over-design for unloaded sections and under-design for critical loading areas if the parts are subject to mechanical loads such as impact and internal pressurization. Uniform thickness distribution will not guarantee an 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 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]. The performance optimization uses a gradient-based technique to determine the minimum part thickness distribution that satisfies the stress constraint following by the process optimization phase to determine the optimal die gap opening profile that minimizes the part weight subjected to the minimum thickness constraint derived from the performance optimization [6][7]. The minimum thickness constraint of each controlling point was determined by retaining the maximum thickness from 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 literatures address 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. Taguchi method [8] is well known for its efficiency and simplicity in parameter design. Inspired from statistical factorial experiments, Taguchi method features orthogonal arrays and analysis of 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 investigating range. The orthogonal array is a type of fractional factorial experiments. The application of orthogonal arrays reduces the number of experiments, which is particular effective for design optimization involving expensive experiments. ANOM study of experiment results reveals the effects of design parameters that 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 Algorithm (GA) applies the evolution principles found in nature to the problem of finding an optimal solution [10]~[12], and is popular in solving complex engineering problems. GA uses a selection operator to avoid trapping at a local optimum that often happens in classical optimizations, when a better optimum may be found outside the vicinity of the current solution. A lot of modifications of the methodology have been proposed since the concept had been first raised in 1975. Among them, competent GA [13] claims to find a global, or near global, solution in reasonable time. Banier and Brisset [14] introduce 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 too complex for usual CSP techniques and whose constraints are too complex for conventional genetic algorithm. The main idea is the handling of sub-domains of the CSP variables by genetic algorithm. 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 sub-problems 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 experiment 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 at the first place. A searching algorithm is applied to search for the optimum in the simulated model instead of interacting with actual engineering system [17][18]. However, a great number of training samples is often required to establish an accurate simulating model which 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. Though modeling imperfection is expectable, additional training samples apply only to the space of interest to reduce sampling cost. Here, the searched optimum from the imperfect model serves as additional training samples [19][20]. Therefore, the training and searching processes iterate to improve the network modeling gradually especially in the probable space of design optimum.

Sampling efficiency is important for the network modeling of the 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 literatures applied Taguchi's orthogonal arrays as training samples for a neural network model and searched for the optimum using GA [21]~[23]. Although the use of orthogonal arrays as training samples reduces the sampling cost, limited learning samples may greatly diminish prediction generality of the trained network model for complex systems like extrusion blow molding. A lower sampling cost is traded for a lower prediction generality. Previous study often overlooks the possibility of the lack of prediction generality for a network model from deficient training samples and a unbounded search for the optimum in the feasible domain of a network model might lead to erroneous results. Even the confirmation result of the search optimum is used to retrain the network model, the iteration often take a long time to converge. Reliable prediction of such 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 the part design and the molding process control. The soft computing based optimization scheme searches for the die programming of the molding process of minimum part weight while satisfying the performance constraints. The design objective is to search for a feasible stress distribution with a minimum deviation to the material allowable stress from manipulating the die gap opening at designate 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.

2. OPTIMIZATION STRATEGY

The proposed optimization strategy, Prediction Reliability Guided Search of Evolving Network Modeling (PREGSEN), first establishes a neural network from a small experimental design, and searches for the optimum of the trained model using genetic algorithm. To cope with possible deficiency of prediction generality due to small learning samples, the strategy introduces the fuzzy prediction reliability to direct the evolution decision in genetic algorithm and increase the evolving priority surrounding training samples. The verification experiment of the derived optimum from 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 convergence of optimum. The flowchart of the proposed optimization strategy is illustrated in Fig. 2

2.1. Evolving Neural Network Model

Neural network technologies are effective in establishing a simulation model from sampling data for engineering systems. Back propagation network (BPN) is a type of supervised learning networks and the most widely used network model [24]. Previous researches [2][16] have proposed a prediction model for extrusion blow molding applications using BPN from extensive experimental data. However, this study applies 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 cost, the number of training samples will be limited, which 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 only provide a quasi-optimum. The verification 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 design optimum to increase the sampling efficiency.

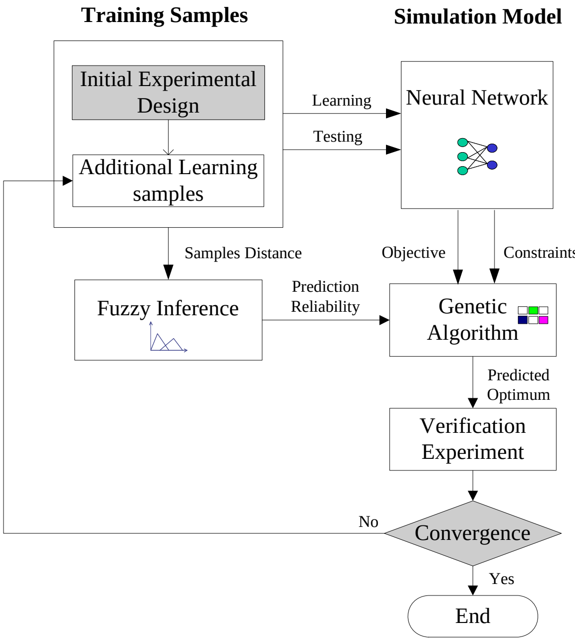


Fig. 2 Optimization flowchart of PREGSEN.

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, which is particular effective for design optimization involving expensive experiments or time-consuming simulations. The control variables are factorized in the preliminary investigating 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 to determine the accuracy and the generality of the network.

2.2. Extrapolation Distance (ED)

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

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

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 a neural network model is better than the extrapolating designs. Also, the closer the predictive design to the training samples, the higher prediction accuracy. This study proposes the ED as a neighboring index of a predictive design, which is defined as the minimum mean Euclid distances between the prediction and the training samples.

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

To facilitate the calculation of the distance among designs, the values of the continuous variable x_k are normalized to z_k using the following transformation:

$$z_k = \frac{\left(x_k - \dfrac{\left(\max(x_k) + \min(x_k)\right)}{2} \right)}{\left(\dfrac{\left(\max(x_k) - \min(x_k)\right)}{2} \right)} \quad (3)$$

where $\max(x_k)$ represents the maximal and $\min(x_k)$ represents the minimal values of the design variable x_k in the training samples. For discrete variables, the factorial values are assigned equally spaced between -1 and +1.

The interpolating designs often have higher prediction accuracy than extrapolating designs in neural network models. This study defines the smallest convex hyper polyhedron surrounding all training samples as the Sampling Enclosure Envelope Space (SES) that is used to differentiate interpolation design and extrapolation designs in a multidimensional simulation model. The boundary of SES is constructed by a set of n-dimensional hyper-planes which are determined by n non-coplanar points from the training samples. If the prediction point is inside or on the boundary of SES, it is an interpolating design; otherwise, the prediction point is an extrapolating design. A two dimensional example is shown in **Fig. 3**, where D_1 is an interpolating design and D_2 is an extrapolating design.

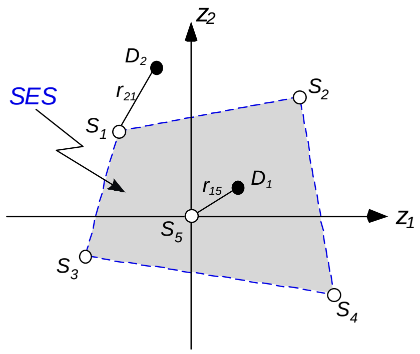


Fig. 3 Extrapolation distances of predicted designs for a two-dimensional example.

The training samples are represented as normalized coordinates z_1 and z_2 to calculate the EDs for predicting designs. The ED is assumed positive for an extrapolating design, and negative for an interpolating design. For instance, an NN is trained from five samples, S_1 ~ S_5, as shown in Fig. 3. The ED of the interpolating design D_1 is designated as "$-r_{15}$" because r_{15} is the shortest mean Euclid distances among r_{1i}, i =1~5. Likewise, the ED of the extrapolating design D_2 is "$+r_{21}$". For an interpolating design with a small ED, it is expected to have a better prediction accuracy, and for an extrapolating design with a large ED, the prediction accuracy is likely doubtful.

2.3. The Fuzzy Reasoning of the Prediction Reliability

Fuzzy systems are widely used in engineering applications to convert the expert knowledge into a mathematic 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 followed by a defuzzifier to convert the linguistic conclusion into a 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 *Extrapolation Distance* of a predictive design. The association of prediction reliability and extrapolation distances is based on two fuzzy concepts. One is to assign less reliability for the prediction point with a farther distance from the learning samples, and the other is to assign less reliability for extrapolating designs than interpolating designs. Seven single-input single output inference rules are proposed based on the nature of simulated models as follows:

R1: If the *ED* of the design is PB then prediction reliability is Bad.

R2: If the *ED* of the design is PM then prediction reliability is Poor.

R3: If the *ED* of the design is PS then prediction reliability is Fair.

R4: If the *ED* of the design is ZE then prediction reliability is Excellent.

R5: If the *ED* of the design is NS then prediction reliability is Good.

R6: If the *ED* of the design is NM then prediction reliability is Fair.

R7: If the *ED* of the design is NB then prediction reliability is Poor.

Seven linguistic levels are defined to describe the condition variable ED: PB(Positive Big), PM(Positive Medium), PS(Positive Small), ZE(Zero), NS(Negative Small), Negative Medium (NM), and NB(Negative Big). Five levels are defined to describe the assessment results for the prediction reliability: Excellent, Good, Fair, Poor, and Bad. Because a small orthogonal arrays are used for the training samples, the maximum extrapolation distance, α, can be approximated using 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 Fig. 4 and Fig. 5. A simple center average defuzzifier is applied to derive the prediction reliability. Fig. 6 is the prediction reliability contour plot for a two-dimensional case using the fuzzy inference. The five solid dots in Fig. 6 represent the training samples in the simulated network. The fuzzy model can generally represent the intrinsic characteristic of the prediction reliability of a network model.

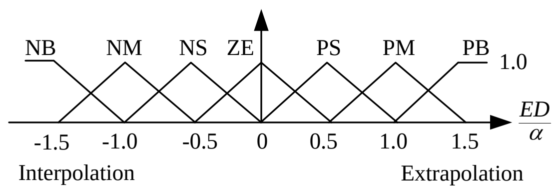


Fig. 4 Membership functions of the condition levels of *ED*.

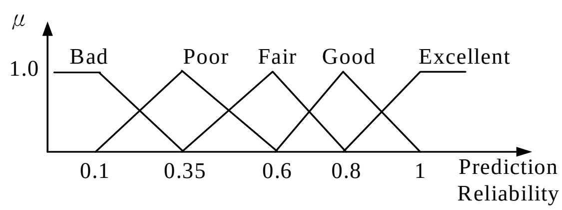


Fig. 5 Membership functions of the assessment levels of prediction reliability.

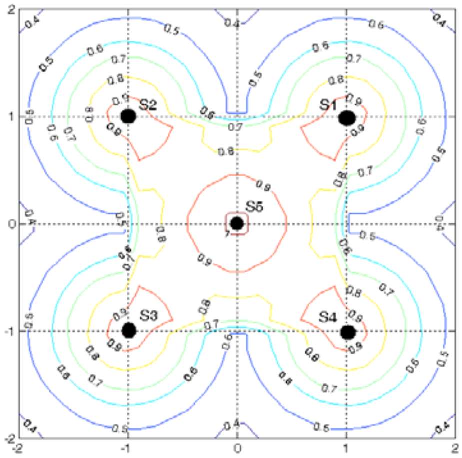


Fig. 6 Prediction reliability contour plot for a two-dimensional example with five training samples

2.4. The Search for the Model Optimum Using Genetic Algorithm

Taking advantage of the fast recall of neural network, GA is applied to search for the optimum of the trained network model established form engineering problems to reduce experimental costs. Genetic algorithms are categorized as global search heuristics, and 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 GA in previous literatures 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, which 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 to the definition of fitness function to prioritize the searching domains to the neighboring space of training samples, and 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 based on the predicted responses from the simulated network and the Prediction Reliability from the fuzzy inference. The fitness function is defined to be 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 higher rank will have advantage in the evolution 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.

$$\textit{Fitness} = \textit{Response Rank} + \textit{Reliability Rank} \qquad (4)$$

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 lack of generality due to scarce training samples. Because of the inclusion of prediction reliability to the fitness function, 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 training samples because of a possible enormous prediction error. The quasi-optimum, on the other hand, is more reliable even for a deficient simulated model.

2.5. Iterative Training and Search for 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 as the addition of new samples from the verification of optimum. If the addition learning sample is an extrapolating design, the SES in the reliability inference will expand, and the searching range in GA will adjust dynamically due to the normalization process and the fuzzy inference.

The proposed algorithm will secure the reliability of the searched optimum in iteration, and evolves the exploration range automatically. Global accuracy of the simulated model is not necessary for the search of optimum. Instead of increasing sampling points evenly distributed in the investigating range, additional sampling points will congregate in the most probable region of the global optimum using the proposed algorithm. The sampling efficiency will thus increase, which is particularly important in engineering applications. The training and searching process iterates until the reach of convergence of the predicted optimum. The quasi-optimum will gradually approach the global optimum. The convergence criteria include (1) the convergence of the predicted optimum and the verified result, and (2) the variation of the last three searched optimum within engineering tolerance. For engineering practice, a trade-off between design improvements and experimental costs is a more important concern.

3. 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 extrusion blow molding process for a High Density Polyethylene (HDPE) bottle. Two types of loading usually used in industrial applications are investigated including an internal pressurization at 90 (psi) and a top displacement of 3.75 (mm) for 5 seconds as illustrated in Fig. 7. The maximum allowable stress, corresponding to the ultimate tensile strength of the material, is 33 MPa. For this material, the Young's modulus is 879 MPa and the thickness shrinkage is 5%. The simulating software of blow molding, BlowSim, is applied to estimate the thickness distribution of the blown bottle. A finite element analysis software, ANSYS is used for the structural analysis of the bottle.

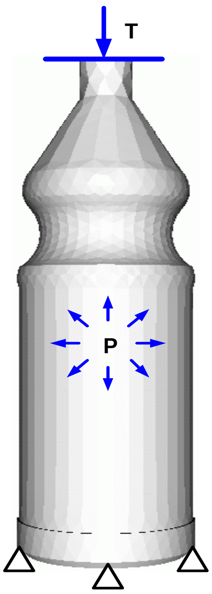


Fig. 7 Two mechanical testing loads of the HDPE bottle: an internal pressurization and a top displacement loading

3.1. The Formulation for Performance Optimization

The design objective is to obtain a wall thickness distribution of minimal weight by manipulating the die gap programming 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: $P(t_j)$, $j=0 \sim 6$.

Constraints: $s_i[P(t_j), P, T] \leqq \sigma_a$

where $P(t_j)$ are the die gap openings of the controlling points as illustrated in Fig. 1, s_i are 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 thickness results in the increase of its stress level. To increase the material efficiency, the stress distribution should be as close to the allowable stress as possible. The smaller variance of the stress distribution, the closer the mean can be moved toward the material yield strength, which leads to thinner elements, and thus reduce part weight. However, any element 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 stress distribution around the allowable stress level and the constraint penalty function as illustrated in **Fig. 8**.

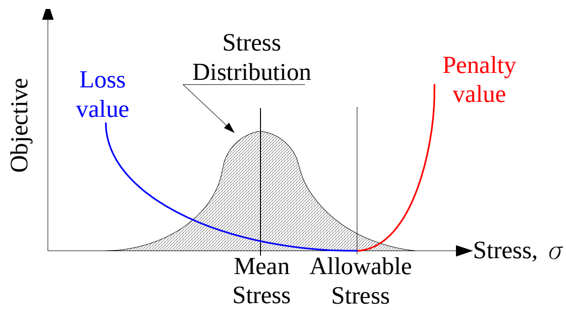


Fig. 8 Illustration of the design objective for performance optimization

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

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

where n is the total number of nodes of the simulation model, s_i the stress of node i, and σ_a the allowable stress of the bottle material.

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

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

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 using a second order singularity function as shown in equation (7). This portion accounts for the penalty of the FEM nodes violating the stress constraint.

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

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

3.2. Design optimization using Taguchi method

Taguchi method applies the analysis of means (ANOM) to estimate parameter sensitivities, which is popular in engineering applications. The die gap openings at 7 discrete extrusion times are selected as the design variables: $P(t_0)$, $P(t_1)$, $P(t_2)$, $P(t_3)$, $P(t_4)$, $P(t_5)$, and $P(t_6)$ to control the parison thickness at 7 evenly distributed sections. The initial design adopts a uniform die gap opening of 75%. Assume three-level for each design variable, a minimal orthogonal array of L18 is selected as the experimental design (Table 1). 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.

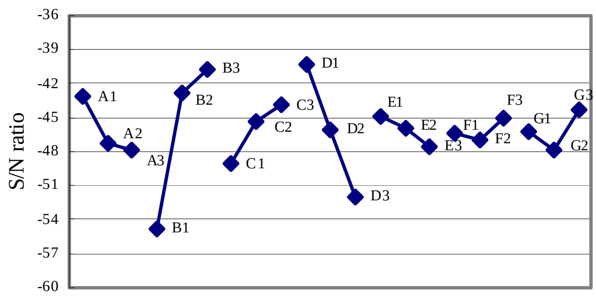
$$S/N = -10 \cdot \log(MOBJ) \tag{8}$$

Fig. 9 is the effect plot for the die opening from the experimental design of Table 1. A design with higher S/N ratio has a smaller value of the objective function. Taguchi's parameter design scheme suggests the optimum treatment to be $A_1B_3C_3D_1E_1F_3G_3$. The verification result using BlowSim and ANSYS for Taguchi's optimum shows a S/N ratio of –37.29 (dB) which is very different from the predicted value of -25.63 (dB) using Taguchi's additive model. The verified performance of the predicted optimum is not even the best among the design experiments. The parameter design using Taguchi method fails due to possible reasons including the interactions among variables and significant system nonlinearity.

Table 1. Experimental design using L18 orthogonal array

L_{18}	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	21791.25	-43.38
2	55	75	75	75	75	75	75	4625.35	-36.65
3	55	95	95	95	95	95	95	3308.56	-35.20
4	75	55	55	75	75	95	95	650545.19	-58.13
5	75	75	75	95	95	55	55	214988.62	-53.32
6	75	95	95	55	55	75	75	5544.66	-37.44
7	95	55	75	55	95	75	95	64469.80	-48.09
8	95	75	95	75	55	95	55	6349.12	-38.03
9	95	95	55	95	75	55	75	119726.34	-50.78
10	55	55	95	95	75	75	55	1245531.06	-60.95
11	55	75	55	55	95	95	75	25238.52	-44.02
12	55	95	75	75	55	55	95	7907.71	-38.98
13	75	55	75	95	55	95	75	645177.17	-58.10
14	75	75	95	55	75	55	95	1565.06	-31.95
15	75	95	55	75	95	75	55	31459.45	-44.98
16	95	55	95	75	95	55	75	997229.07	-59.99
17	95	75	55	95	55	75	95	222059.20	-53.46
18	95	95	75	55	75	95	55	5040.66	-37.02
Initial	75	75	75	75	75	75	75	8229.76	-39.15

[a] *MOBJ*: Modified Objective Function



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: $P(t_6)$

Fig. 9 Effect plot for the die opening during the blow molding of the bottle

3.3. Optimization of Bottle Thickness Distribution using PREGSEN

3.3.1. Establishing the simulated Neural Network model

Training samples are essential to the prediction quality of network models. The L18 orthogonal array from previous Taguchi's application is used as learning samples to reduce the number of experiments and to maintain a good sample representation. Another two–level orthogonal array (L8) illustrated in Table 2 is selected as the testing samples for the network training. The level values, 65% and 85%, are set in between the 3-level values, 55%, 75% and 95%, of the learning samples.

The steepest gradient method is assumed to train the weighting matrices of the Back Propagation Network (BPN). There is no definite rule available to determine appropriate parameters in the networks training. This study applies a simple Taguchi's 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 using an L9 orthogonal array experiments and the analysis of mean (ANOM) for the optimal training efficiency at first 10 epochs. The parameter design in this case suggests 19 neurons in the hidden layer, the initial learning rate of 0.5, the decreased learning rate of 0.85, and the increased learning rate of 1.15.

Table 2. Testing samples using L8 orthogonal array

L_8	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	34842.0	-45.42
2	65	65	65	85	85	85	85	438815.8	-56.42
3	65	85	85	65	65	85	85	4051.5	-36.08
4	65	85	85	85	85	65	65	10328.1	-40.14
5	85	65	85	65	85	65	85	3880.4	-35.89
6	85	65	85	85	65	85	65	153796.5	-51.87
7	85	85	65	65	85	85	65	17050.2	-42.32
8	85	85	65	85	65	65	85	45926.9	-46.62

[a] *MOBJ*: Modified Objective Function

3.3.2. Evolving modeling and optimization

As illustrated in Fig. 2, the prediction reliability is introduced to the fitness function of the optimization search using GA. The anchor parameter α in the member function of Fig. 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 using a binary genetic algorithm (BGA). In this study, the bit length of encoded chromosome is assumed 12. Because that the design variables have been normalized using Eq.(3), the searching boundary in GA is set to be ±1.5 to explore possible optimum outside the preliminary design space. The simulated Neural Network model will then provide the response estimation for each chromosome combination.

The probability of crossover should have a larger value; typically P_c ranges from 0.5 to 1.0. The single-point crossover and mutation were used in this studied. The mutation operator must be used with low probability; typically, the mutation probability ranges from 0.01 to 0.1. Again, the parameters of the GA were obtained using 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 neural network and genetic algorithm (NNGA). GA will assume global accuracy in the investigating range and search for a design with the best performance in the current simulated model. The derived optimum from the GA search might be different from the actual optimum of the engineering system due to the imperfection of current simulated model. The searched optimum is verified using 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 Fig. 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 has not shown convergence yet after 51 iterations.

Next, the proposed algorithm, PREGSEN is applied to the same problem. The fitness function is modified using the fuzzy prediction reliability. As illustrated in Fig. 11, although the verified value and the quasi-optimum of the initial network model obtained using prediction reliability guided GA search are still different due to lack of generality of the initial model. The difference is greatly reduced due to a more reliable quasi-optimum is provided from the reliability guided search. As 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 (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 Fig. 11, there is significant constraint violation as we examine their corresponding objectives (Table 3). The penalty loss represents that the stresses of some FEM nodes exceed the allowable stress of the bottle material which might result in part failure. Also they haven't 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 4.

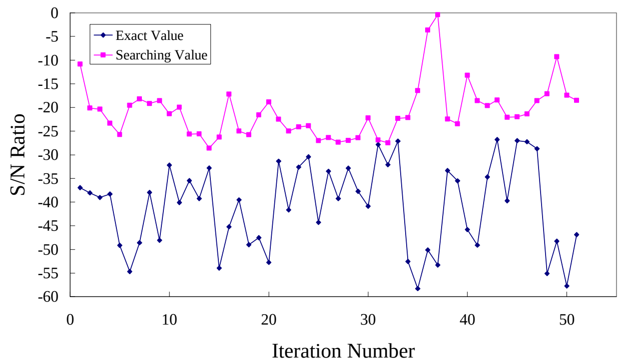


Fig. 10 Iteration result for a simple recursion of NN and GA

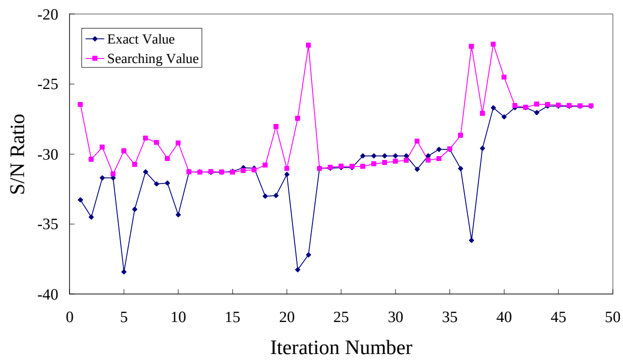


Fig. 11 Iteration result using PREGSEN

the proposed method with Taguchi method and a simple iteration of NN-GA in terms of the design feasibility, part weight, and searching efficiency. Fig. 12 represents the profiles of optimal die gap openings of parison programming, and Table 5 shows the stress distributions under test loads for the initial design and the optimums obtained from various methods. Taguchi's ANOM approach is liable to parameter interactions and system non-linearity, and fails to find a lighter weight design than the initial design. The optimum from Taguchi method has a larger distribution and a smaller mean stress, which results in a poor material efficiency, and still a strong violation for the stress constraint.

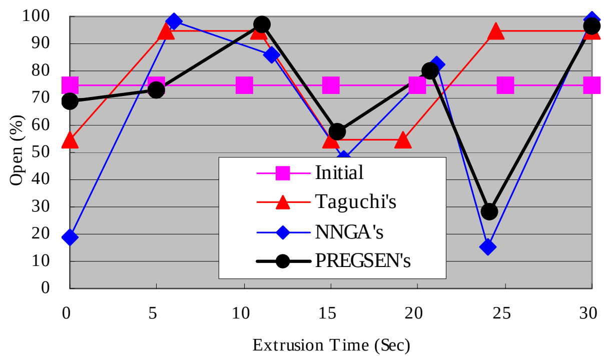


Fig. 12 The die gap opening for various optimal designs.

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

3.4. Comparison of the results

This session compares the optimization results from

Table 3 Comparison of different iterations

Iteration No.	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] Prediction error = the difference between the predicted objective and the verified result
[b] Coefficient of Variation (COV) = the standard deviation divided by the mean of last three searched objectives

Table 4 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

PREGSEN provides a much reliable and efficient search as shown in Table 5. The definition of the fitness function will suppress the exploration of the regions far away from the current training points even the prediction from the current network model looks promising. However, exploration range of GA will grow dynamically as the addition of new training samples from the when the verification of the predicted optimum. PREGSEN has reached the optimum convergence at iteration 48. PREGSEN's optimum exhibits the smallest stress deviation and leads to a design with the weight of 114.31 (g) while satisfying the stress constraints. Fig. 13 is the comparison of the stress distribution under test loads using ANASYS, and shows that PREGSEN's optimum has the most even stress distributions among various designs.

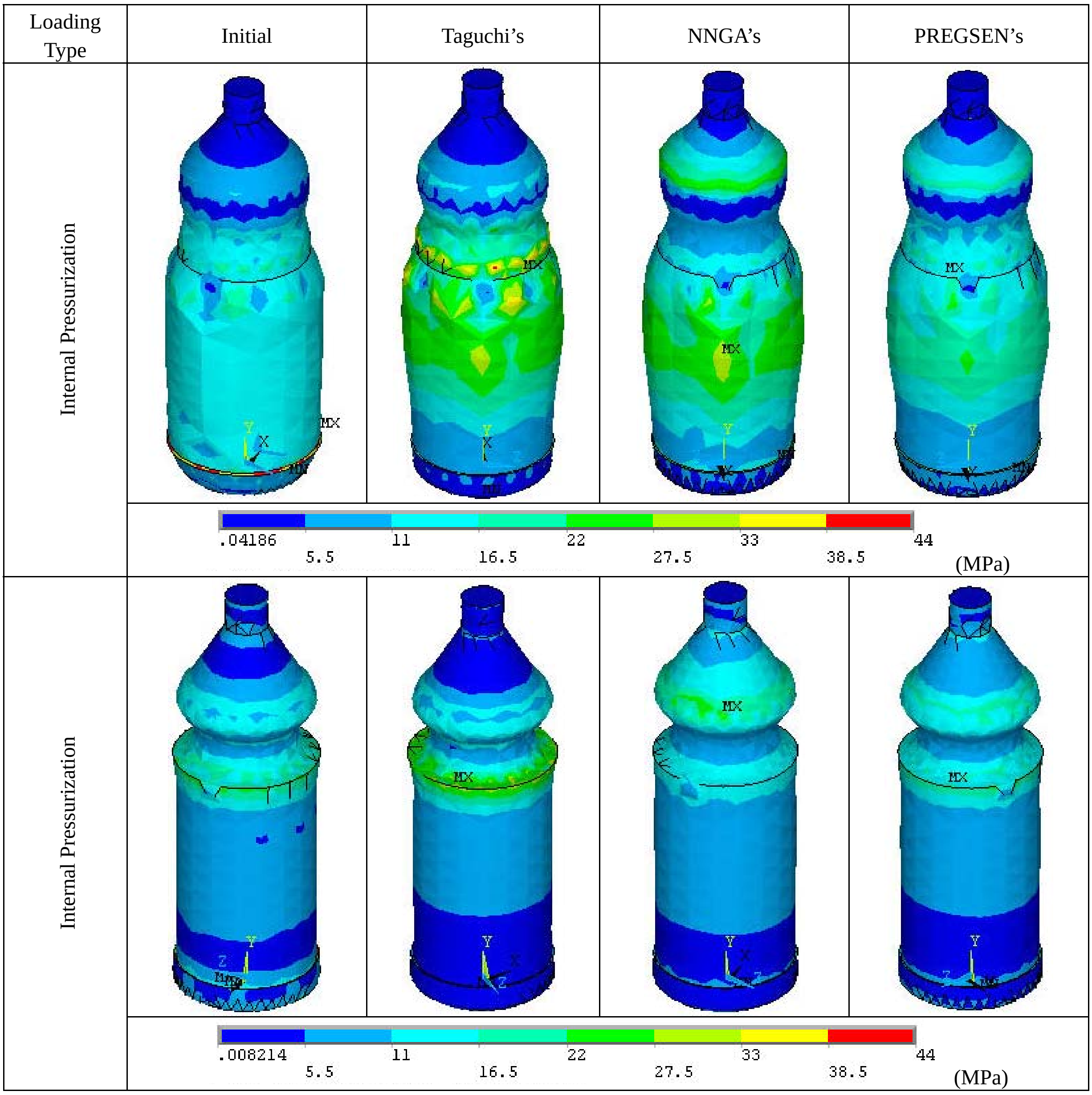


Fig. 13 Comparison of the stress distribution under test loads

Table 5 Comparison of various optima

	Mean Stress	Std. Dev. Stress	Quality Loss	Penalty Loss	Objective	Weight (g)
Initial	13.95	8.78	439.8	7789.9	8229.7	118.7
Taguchi's	12.84	9.36	494.1	4868.2	25362.3	119.2
NNGA's	12.76	7.39	464.3	14.0	478.3[a]	110.5[a]
PREGSEN's	12.62	6.21	453.9	0.0	453.9	114.3

[a] Not yet converge at the iteration of 51. The listed result is the best design so far.

4. CONCLUSIONS

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

Finally, the searching scheme, PREGSEN, proposes an evolving network model that starts from a small number of training samples using Taguchi's orthogonal array and selectively evolves for the most probable space of design optimum to increase sampling efficiency. For complex simulation systems as the finite element analysis of structure mechanics and extrusion blow molding process, the number of engineering simulations will greatly affect 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 neural network.

The prediction reliability of the network model is likely restricted to the surroundings of the learning samples. The accuracy of extrapolating prediction depends on model complexity. If the model is nearly linear, the extrapolating prediction will be pretty accurate. As the model nonlinearity increases, extrapolating accuracy decreases because of unknown trend out of 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 optimum search. The proposed optimization scheme applies the fuzzy reasoning of the prediction reliability for the evolving network model to guide the GA search for a reliable quasi-optimum instead of a false optimum of the imperfect network 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 approaching the actual peak. If the predicted optimum is extrapolated, the verification suggests additional information to explore probable region of 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.

Acknowledgments

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC91-2212-E-155-007, and Dr. F. Thibault for his consultation in the simulation using BlowSim.

References

[1] Huang, H. X.; Lu, S. Journal of Reinforced Plastics and Composites 2005, 24, 1025.
[2] Diraddo, R. W.; Garcia-Rejon, A. Polymer Engineering and Science 1993, 33, 653.
[3] Laroche, D.; Kabanemi, K.; Pecora, L.; Diraddo, R. Polymer Engineering and Science 1999, 39, 1223.
[4] Yu, J.; Hung, T. R.; Thibault, F. Proceedings of the ASME Design Engineering Technical Conference 2002, 2, 133.
[5] Gauvin, C.; Thibault, F.; Laroche, D. Polymer Engineering and Science 2003, 43, 1407.
[6] Lee, D. K.; S. K. Soh Polymer Engineering and Science 1996, 36, 1513.
[7] Hsu, Y. L.; Liu, T. C.; Thibault, F.; Lanctot, B. Journal of Engineering Manufacture, Proc Instn Mech Engrs, Part B 2004, 218, 197.
[8] Taguchi, G. Int. Journal of Production Research 1978, 16, 521.
[9] Tahboub, K. K.; Rawabdeh, I. A. Journal of Quality in Maintenance Engineering 2004, 10, 47.

[10] Holland, J. H. Adaptation in Natural and Artificial Systems; Ann Arbor, MI: The University of Michigan Press, 1975.
[11] Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning; Addison-Wesley, Massachusetts, USA, 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, 1997.
[13] Santarelli, S.; Yu, T. L.; Goldberg, D.; Altshuler, E.; O'Donnell, T.; Southall, H.; Mailloux, R. Mathematical and Computer Modeling 2006, 43, 990.
[14] Barnier, N.; Brisset, P. Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence 1998, 645.
[15] Pelikan, M.; Sastry, K.; Goldberg, D. International Journal of 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. Materials and Design 2006, 27, 735.
[18] Wang, L. Applied Mathematics and Computation 2005, 170, 1329.
[19] Yu, J.; Chen, X.; Hung, T. R.; Thibault, F. Journal of Intelligent Manufacturing 2004, 15, 625.
[20] Hsu, Y.; Dong, Y.; Hsu, M. JSME International Journal, Series C 2001, 44, 103.
[21] Su, C. T.; Chiu, C. C.; Chang, H. H. Int. Journal of Industrial Engineering: Theory Applications and Practice 2000, 7, 224.
[22] Singh, A. K.; Panda, S. S.; Chakraborty, D.; Pal, S. K. International Journal of Advanced Manufacturing Technology 2006, 28, 456.
[23] Sanjari, M.; Taheri, K.; Movahedi, R. International Journal of Advanced Manufacturing Technology 2009, 40, 776.
[24] Cybenko, G. Math. Control Signals Syst 1989, 2, 303.
[25] Thierens, D.; Goldberg, D. International Conference on Evolutionary Computation 1994, 1, 508.