A fuzzy optimization algorithm for blow moulding process

Authors: Yeh-Liang Hsu, Tzu-Chi Liu, Ming-Sho Hsu, Francis Thibault (2001-10-21); recommended: Yeh-Liang Hsu (2001-10-22).

Note: This paper is revised and presented at 「中國機械工程學會第十八屆學術研討會」. The original paper was presented at The Sixth International Conference on CSCW Design, July 12-14, 2001, London, Ontario, Canada.

A fuzzy optimization algorithm for blow moulding process

Abstract

This paper demonstrates a fuzzy optimization algorithm for determining the optimal gap openings of the programming points in the blow moulding process. Traditional numerical optimization algorithms treat the optimization problem as pure mathematical problems. Valuable engineering knowledge is not utilized in the optimization process. The idea of the fuzzy optimization algorithm is that, instead of using purely numerical information to get the new design point in the next iteration, engineering knowledge and human supervision process can be modeled in the optimization algorithm using fuzzy rules. The fuzzy optimization algorithm developed in this paper works well on the blow moulding examples.

Keywords: fuzzy optimization, blow moulding.

1. Introduction

Blow moulding is the forming of a hollow part by “blowing” a mould-cavity-shaped parison that is made by thermoplastic molten tube [Lee, 1990]. The thickness of parison determines the thickness of the blown hollow part, which is controlled by the gap opening

http://designer.mech.yzu.edu.tw
A fuzzy optimization algorithm for blow moulding process

space between the die and the mandrel. Figure 1(a) illustrates parison programming, which manipulates the gap opening of die at programming points to control the thickness of parison. In order to obtain uniform thickness distribution of the hollow part as shown in Figure 1(b), the thickness of programmed parison must be non-uniform. For example, in Figure 1(b), the parison thickness for the greatest expansion area must be thicker than those of the other areas.

![Figure 1. Illustration of parison programming](http://designer.mech.yzu.edu.tw)

BlowSim is a finite element software package designed to simulate the extrusion blow moulding, injection stretch blow moulding, and thermoforming processes. It is developed by the Industrial Material Institute (IMI) of National Research Council (NRC), Canada. Figure 2 shows the thickness distribution of a sample problem simulated by BlowSim when the gap opening is set at 75% at all time.
A fuzzy optimization algorithm for blow moulding process

In the blow moulding process, it is desirable to obtain a final part of constant thickness. It is therefore an optimization problem on how to control the gap openings to minimize the deviation of the thickness of the final part from the target thickness. For the sample problem shown in Figure 2, the objective is to minimize the average deviation of the thickness of all nodes from the target thickness:

$$\text{min. } f = \left( \frac{1}{n} \sum_{j=1}^{n} (T^j_i - T_t)^2 \right)^{1/2}$$

where $T_t$ is the target thickness, $T^j_i$ is the thickness at node $j$ in the $i$-th iteration, and $n$ is the total number of nodes. We are able to extract the thickness of all nodes from the simulation results of BlowSim and plug them into Eq.(1) to get the objective function value. The gap-opening rates at discrete time points are the design variables. Obviously, the thickness at node $j$ in the $i$-th iteration $T^j_i$ are functions of the design variables, namely, the gap opening rates.

We can certainly use a gradient type numerical optimization algorithm to solve for the optimal opening rates. On the other hand, manufacturing engineers usually adjust the gap opening rates empirically: reduce the gap opening rate if the final part is too thick, and vice versa.
The purpose of this paper is to illustrate how the concept of “fuzzy optimization algorithms” to close the gap between numerical optimization algorithms and engineering experience, using the optimization of the blow moulding process as an example. This paper first explains the concept of fuzzy optimization algorithms. Then in our blow moulding optimization problem, engineering experiences for adjusting the gap opening rates are modeled as fuzzy rules which determines the search direction of a line search algorithm. The optimization results are presented, and the advantages of this approach are discussed.

2. The concept of “fuzzy optimization algorithms”

The optimization process can be viewed as a closed-loop control system. Figure 3 shows a general block diagram of an automatic control system. An error detector compares a signal obtained through feedback elements, which is a function of the output response, with the reference input. Any difference between these two signals constitutes an error or actuating signal, which actuates the control elements. The control elements in turn alter the conditions in the plant (controlled member) in such a manner as to reduce the original error.

![Figure 3. General block diagram of an automatic control system](http://designer.mech.yzu.edu.tw)

Figure 4 shows a general block diagram for an optimization process. Comparing with Figure 3, an optimization model in an optimization process is analogous to the plant in a control system; an optimization algorithm is analogous to the controllers. Initial parameters are input to the optimization algorithm, which in turn generates a trial design point according to its search rules. The optimization model is then evaluated at this trial design point, and the information such as objective and constraint function values and sensitivity is fed back to the
termination test. If the termination test fails, the optimization algorithm is triggered again to generate the next design point, using the numerical information from previous iterations.

![Figure 4. General block diagram of a design optimization process](http://designer.mech.yzu.edu.tw)

Traditional numerical optimization algorithms are analogous to direct digital controllers. The algorithms are usually “crisply” designed for well defined mathematical models. Their numerical rules for generating the next design point are exact and definite, and they can usually be proved to have nice converging behavior when applying to well defined mathematical models.

However, in engineering optimization problems, we seldom have well defined mathematical models. In the blow moulding example, we know that the thickness at node \( j \) in the \( i \)-th iteration \( T_i^j \) are functions of the design variables, namely, the gap opening rates. But these functions often do not have exact algebraic forms in terms of the design variables, and they can only be evaluated through experiments or computer simulations, which are expensive and imprecise in nature. Usually sensitivity calculation is also done through imprecise finite difference methods. Very often the cost of the number of function evaluations required to meet the “crisp” definition of the numerical algorithms is simply too high to be affordable.

On the other hand, when we apply numerical optimization algorithms on an engineering problem, we treat the engineering problem as a pure mathematical problem. Engineering heuristics are totally ignored in the numerical optimization algorithms. This motivates the idea that, in addition to crisp numerical rules, the human supervision process should also be modeled in an optimization algorithm using fuzzy rules; the “controllers” in the optimization process may as well be fuzzy controllers!
This concept was proposed by Hsu et al [1995]. Mulkay and Rao [1998] also presented the same idea. In both work, fuzzy heuristics are used to control the parameters of the optimization algorithm to improve its performance. In the following section, fuzzy heuristics are used to generate the new design point of the next iteration.

3. The optimization algorithm used in “BlowOp”

“BlowOp” is the optimization module of BlowSim used to obtain a uniform thickness distribution and minimize part weight. In BlowOp, the parison thickness, which are directly affected by gap opening rates, are used as design variables. The optimization procedure is given by the following equation:

\[ A_{i+1} = A_i - \alpha_u \alpha_p (T_i - T_s) \]  

where \( A_{i+1} \) and \( A_i \) are parison thickness of each programming point at iteration \( i+1 \) and iteration \( i \). They are related to gap opening rates and extrusion time. \( T_i \) is the thickness of the part at iteration \( i \). \( T_s \) is target thickness of the final part, which has uniform thickness distribution. \( \alpha_u \) is a user-defined proportional gain; while \( \alpha_p \) is a gain defined by the inflation model:

\[ \alpha_p = A_i / T_i. \] 

This optimization procedure is a line search type of algorithm, which is commonly used in various numerical optimization algorithms. The standard line search algorithm will be [Vanderplaats, 1984]:

\[ x^{i+1} = x^i + \alpha s \] 

where \( x \) is the vector of design variables, \( s \) is the search direction, and \( \alpha \) is the step length.

In standard line search algorithms, step length \( \alpha \) is decided after several trials according to certain numerical rules. In the procedure shown in equation (2), \( \alpha \) takes a fixed value 0.5.

In numerical optimization algorithms, search directions are defined by numerical information, for example, gradient or Hessian of the objective function. However, in the procedure shown in equation (2), the search direction is determined by \( \alpha_p = A_i / T_i \), and \( (T_i - T_s) \).
Figure 5 shows the iteration history of this optimization procedure, when applied on the sample problem shown in Figure 2. The gap opening is set at 75% at all time in the first iteration. The target thickness $T_s$ is 2mm. In the first iteration, the maximum part thickness is 6.804mm, the minimum part thickness is 1.394mm, and the value of the objective function (Eq. (1)) is 2.474. After 10 iterations, the objective function value reduces from 2.474 to 0.953.

![Figure 5. Iteration history of BlowOp](http://designer.mech.yzu.edu.tw)

The search algorithm in BlowOp actually uses engineering heuristics in a crisp format. The change in parison thickness between iterations is determined by $\alpha_p(T_i - T_s)$. The term $\alpha_p$ is affected by the geometry of the bottle. The part thickness becomes smaller if the radius of the bottle is large. Therefore $\alpha_p$ decreases as the radius increases. The term $(T_i - T_s)$ represents the simple rule mentioned earlier: reduce the gap opening rate if the final part is too thick, and vice versa.

While the search procedure in Eq. (2) makes perfect sense to engineers, there is no theoretical justification to why the change in parison thickness between iterations is determined by $\alpha_p$ “multiplied” by $(T_i - T_s)$. If a 3rd or 4th factors are found, it will be hard to find a proper “crisp” formula to combine these 3 or 4 terms together.
4. The fuzzy optimization algorithm

The engineering heuristics used in the optimization procedure discussed in the previous section are

(1) If the thickness of a certain node is larger than the target thickness, then reduce the respective opening rate.

(2) If the thickness of a certain node is smaller than the target thickness, then increase the respective opening rate.

(3) If the radius at a certain node is large, the change of opening rate is large.

(4) If the radius at a certain node is small, the change of opening rate is small.

Engineering rules (1) and (2) are first used to try out the fuzzy optimization idea on the sample problem shown in Figure 2. Eq.(4) can be written into the following scalar form for our optimization problem:

\[ P_{i+1}(t_j) = P_i(t_j) + \alpha S_j \]  

where the gap-opening rates at 7 discrete time points as the design variables: \( P_i(t_0) \), \( P_i(t_1) \), \( P_i(t_2) \), \( P_i(t_3) \), \( P_i(t_4) \), \( P_i(t_5) \), \( P_i(t_6) \). In our fuzzy optimization algorithm, the change of opening rate \( S_j \) is to be determined by fuzzy rules. The input variables to the fuzzy rules are thickness \( T_i \) at the 7 discrete time points.

Five levels are defined to describe the linguistic variables: PB(Positive Big), PS(Positive Small), ZE(Zero), NS(Negative Small), and NB(Negative Big). Standard membership functions shown in Table 1 and Figure 6 are used. Five fuzzy rules are established:

R1: If thickness is PB then \( S_j \) is NB
R2: If thickness is PS then \( S_j \) is NS.
R3: If thickness is ZE then \( S_j \) is ZE.
R4: If thickness is NS then \( S_j \) is PS.
R5: If thickness is NB then \( S_j \) is PB.

Table 1. Membership function
Figure 6. Membership functions

Table 2 is the quantization table for the input and output variables. This table is derived from engineering experience to the blow moulding process and the optimization process. Step size $\alpha$ in Eq.(4) is fixed at 1.0. Figure 5 shows the nice converging behavior of this optimization procedure, when applied on the sample problem shown in Figure 2. The gap opening is set at 75% at all time in the first iteration. Figure 7 compares the iteration history with that of BlowOp. From Figure 7, it appears that BlowOp and the fuzzy optimization algorithm has similar performance in this case. After 10 iterations, the objective function value reduces from 2.474 to 0.834.
Table 2. Quantization table

<table>
<thead>
<tr>
<th>Thickness</th>
<th>fuzzy input</th>
<th>Fuzzy reasoning</th>
<th>$s_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target thickness + 2*Max(Thickness- target thickness)/3</td>
<td>2</td>
<td>2</td>
<td>1- $P(t_i)$</td>
</tr>
<tr>
<td>Target thickness + Max(Thickness- target thickness)/3</td>
<td>1</td>
<td>1</td>
<td>[1- $P(t_i)$]/2</td>
</tr>
<tr>
<td>Target thickness</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target thickness + Min(Thickness- target thickness)/3</td>
<td>-1</td>
<td>-1</td>
<td>-[ $P(t_i)/2$]</td>
</tr>
<tr>
<td>Target thickness + 2*Min(Thickness- target thickness)/3</td>
<td>-2</td>
<td>-2</td>
<td>- $P(t_i)$</td>
</tr>
</tbody>
</table>

Figure 7. Iteration history of the fuzzy optimization algorithm with one fuzzy input

5. Adding a new set of fuzzy rules

The fuzzy optimization algorithm discussed in the previous section considers only the engineering heuristics related to the final part thickness. We now include the other engineering heuristics related to the radius of the die into the fuzzy optimization algorithm:

R6: If the Radius is PB, then $S_i$ is PB.
R7: If the Radius is PS, then $S_j$ is PS.
R8: If the Radius is ZE, then $S_j$ is ZE.
R9: If the Radius is NS, then $S_j$ is NS.
R10: If the Radius is NB, then $S_j$ is NB

The five levels and their membership functions are defined in Section 4. Table 4 is the quantization table for the input and output variables. Note that right now there are two fuzzy inputs.

Figure 8 compares the iteration history of the fuzzy optimization algorithm with two fuzzy inputs with that of BlowOp. The fuzzy optimization algorithm has better performance in this case. It converges faster, and the final objective function value is lower. After 10 iterations, the objective function value reduces from 2.474 to 0.811.

![Iteration history of the fuzzy optimization algorithm with two fuzzy inputs](http://designer.mech.yzu.edu.tw)

**Figure 8. Iteration history of the fuzzy optimization algorithm with two fuzzy inputs**

Ideally, the objective function should converge to zero, which indicates constant thickness throughout the whole part. Figure 8 also shows the iteration history when we increase number of programming points from 7 to 31, using the same fuzzy optimization algorithm. With this increase in resolution, the value of the objective function further drops to 0.515.
Table 4. Quantization table with engineering rules

<table>
<thead>
<tr>
<th>Thickness</th>
<th>Fuzzy input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target thickness + 2*(Max(Thickness-target thickness))/3</td>
<td>2</td>
</tr>
<tr>
<td>Target thickness + Max(Thickness-target thickness)/3</td>
<td>1</td>
</tr>
<tr>
<td>Target thickness</td>
<td>0</td>
</tr>
<tr>
<td>Target thickness + Min(Thickness-target thickness)/3</td>
<td>-1</td>
</tr>
<tr>
<td>Target thickness + 2*(Min(Thickness-target thickness))/3</td>
<td>-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radius</th>
<th>Fuzzy input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean(Radius) +(2*(Max(Radius)-mean(Radius))/3)</td>
<td>2</td>
</tr>
<tr>
<td>Mean(Radius) +(Max(Radius)-mean(Radius))/3</td>
<td>1</td>
</tr>
<tr>
<td>mean(Radius)</td>
<td>0</td>
</tr>
<tr>
<td>2*mean(Radius)/3</td>
<td>-1</td>
</tr>
<tr>
<td>mean(Radius)/3</td>
<td>-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuzzy reasoning</th>
<th>$S_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$1 - P_i(t_j)$</td>
</tr>
<tr>
<td>1</td>
<td>$[1 - P_i(t_j)]/2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>$- P_i(t_j)/2$</td>
</tr>
<tr>
<td>-2</td>
<td>$- P_i(t_j)$</td>
</tr>
</tbody>
</table>
Figure 9. Variations in thickness distribution through the iterations

Figure 9 shows the variations in thickness distribution through the iterations. It appears that the thickness distribution is close to constant in the final part from iteration 10. Figure 10 compares the profiles of optimal gap openings of programming points obtained from BlowOp, the fuzzy optimization algorithm with one fuzzy input, the fuzzy optimization algorithm with two fuzzy inputs, and the case with 31 programming points.
6. Automobile fluid reservoir example

We then apply the same fuzzy optimization algorithm described in the previous sections to an automobile fluid reservoir example, again with one fuzzy input and two fuzzy inputs. As shown in Figure 11, the shape of the reservoir is very complicated and unsymmetrical. The quantization table and the membership functions are the same as defined in Section 4 and in Section 5. The target thickness is 5 mm.

Figure 11 is the final thickness distribution after 10 iterations using the fuzzy optimization with one fuzzy input. Figure 12 compares the iteration history of the fuzzy optimization algorithm with that of BlowOp. Judging from Figure 12, BlowOp and the fuzzy optimization algorithm have similar performance in this case. After 10 iterations, the objective function value of BlowOp decreases from 1.4773 to 1.2338, and the objective function value of fuzzy optimization algorithm with one fuzzy input decreases to 1.2934. Because the shape of the reservoir is unsymmetrical, at one horizontal cross section near the bottom of the reservoir, the thickness may vary from 3.8643 mm to 4.1446 mm. Therefore it may not be possible to further reduce the objective function value using only gap openings as design variables.
The “radius” of a cross section of the reservoir is hard to define. We use “mean radius” instead in our section fuzzy input. But the fuzzy optimization algorithm with two fuzzy inputs works poorly in this case. As shown in Figure 12, the objective function value increases from 1.4773 to 1.6538, which indicates that we may need more experiences in the blow moulding of complex geometry model to construct better fuzzy rules.
7. The bottle case with internal pressure and top load

Back to the bottle case, but now we have two types of loading, one is internal pressure and the other is top load, as illustrated in Figure 13. A performance optimization [Thibault, 2001] to minimize the weight of the bottle subject to stress constraints has been performed, and the result obtained is a non-uniform part thickness distribution as shown in Figure 14.

Figure 13 Two different types of loading were applied: (left) internal pressure and (right) a top loading.

We used the same fuzzy optimization algorithm described in previous sections to obtain the optimal gap openings that will achieve this non-uniform part thickness. Figure 14 compares the thickness distribution obtained by the optimal gap openings and the target non-uniform thickness distribution after 30 iterations.
A fuzzy optimization algorithm for blow moulding process

8. Conclusions and discussion

This paper demonstrates how to develop a fuzzy optimization algorithm for determining the optimal gap openings of the programming points in the blow moulding process. Traditional numerical optimization algorithms treat the optimization problem as pure mathematical problems. Valuable engineering knowledge is not utilized in the optimization process. The idea of developing a fuzzy optimization algorithm is that, instead of using purely numerical information to get the new design point in the next iteration, engineering knowledge and human supervision process can be modeled in the optimization algorithm using fuzzy rules. The fuzzy optimization algorithm developed in this paper works well on our blow moulding examples.

In the blow moulding examples, one major advantage of the fuzzy optimization algorithm over traditional numerical optimization algorithms is that the first order gradient information, which is often expensive or unavailable in real engineering problems, is not required. Another advantage is its flexibility. As shown in the paper, it is easy to expand the fuzzy optimization algorithm if new rules and/or new fuzzy inputs are considered.

Figure 14 Compare the thickness with Fuzzy optimization and target
9. Acknowledgement

This research is supported by the joint project between National Science Council, Taiwan, and Industrial Material Institute, National Research Council, Canada, NSC 89-2212-E-155-019. This support is gratefully acknowledged.

10 References


