Minimizing Part Warpage in Injection Molding by Optimizing Wall Thickness Distribution

MARIO STUDER, FRANK EHRIG

Institute of Materials Science and Plastics Processing, University of Applied Science, 8640 Rapperswil, Switzerland

Correspondence to: Mario Studer; e-mail: mstuder@hsr.ch.

Received: April 21, 2014
Accepted: June 10, 2014

ABSTRACT: The warpage of injection-molded parts is an omnipresent problem in new product development. In this study, a method is developed to reduce the warpage by optimizing the wall thickness distribution of a part. This method is based on an iterative optimization procedure consisting of mold flow analysis, a mesh parameterization tool for geometry manipulation, and an optimization algorithm. Starting with the initial design for the mold cavity, the procedure optimizes the wall thickness distribution at user-defined area sections within prescribed tolerances. A unit-square parameterization technique is applied that uses mean-valued coordinates and a second-order polynomial thickness distribution. Three different optimization algorithms—an evolutionary algorithm, a surrogate-based response surface method, and a direct search method—are considered, and the optimization performances of these algorithms are evaluated for a warpage demonstrator and real industrial part. The results demonstrate the effectiveness of the developed procedure and plausibility of the optimized thickness distribution. Adv Polym Technol 2014, 33, 21454; View this article online at wileyonlinelibrary.com. DOI 10.1002/adv.21454

KEY WORDS: Injection molding, Processing, Simulations

Introduction

The high efficiency of injection molding makes it one of the most widely applied processes for producing plastic parts. During this process, a polymeric material is subjected to high thermal and mechanical stresses. The short processing times involved do not allow the induced mechanical stresses inside the mold to completely relax, making inhomogeneous deformations highly likely after ejection from the mold. Thus, the shapes of molded parts often differ from their intended design, resulting in warpage. If the warpage values lie beyond an acceptable tolerance and cannot be eliminated before the mold is fabricated, costly and time-consuming follow-up processes are required.

Therefore, the use of numerical simulations to predict the warpage of parts has become a significant research effort over the past three decades. Thus, numerous simulation software packages are available.[1] These packages solve equations for the conservation of energy, impulse, and mass using different approximation schemes and calculate the deformation trajectory during the molding process. The accuracy of the predicted warpage values depends strongly on the quality of the material–data input and the continuum mechanics approaches that are used.[2] Although research in this area is still in progress, especially for polymers with complex architectures,[3,4] such as semicrystalline and fiber-reinforced polymers, these simulations have been widely accepted in industry.

After running a simulation and predicting warpage, a high level of expertise is still required to produce a satisfactory design. Thus, many time-consuming iteration steps are required and there is no guarantee of finding the optimal design; thus, efficient optimization strategies are needed. Several researchers have studied this problem, and their approaches for reducing the warpage can be categorized into three primary methods:

1. optimization of the process variables,[5–15]
2. optimization of the gate location,[16–21] and
3. modification of the part design by optimizing the wall thickness of the part.[22,23]

Many studies[5–15] have been published on the optimization of process variables, such as the melting and mold temperatures, the injection speed, and the holding pressure or cooling time, to minimize the warpage. The objective is to obtain a functional relationship between the process variables and the warpage with a minimum of computational effort. Therefore, several surrogate models have been developed,[6] such as artificial neural networks, response surface methods, and combinations of surrogate models with evolutionary algorithms. This method can be used to effectively minimize the part warpage. However, some results yield impractical values for the optimized process conditions, such as holding-pressure times that are smaller than gate freeze off times, which would impair the reproducibility of the parts.

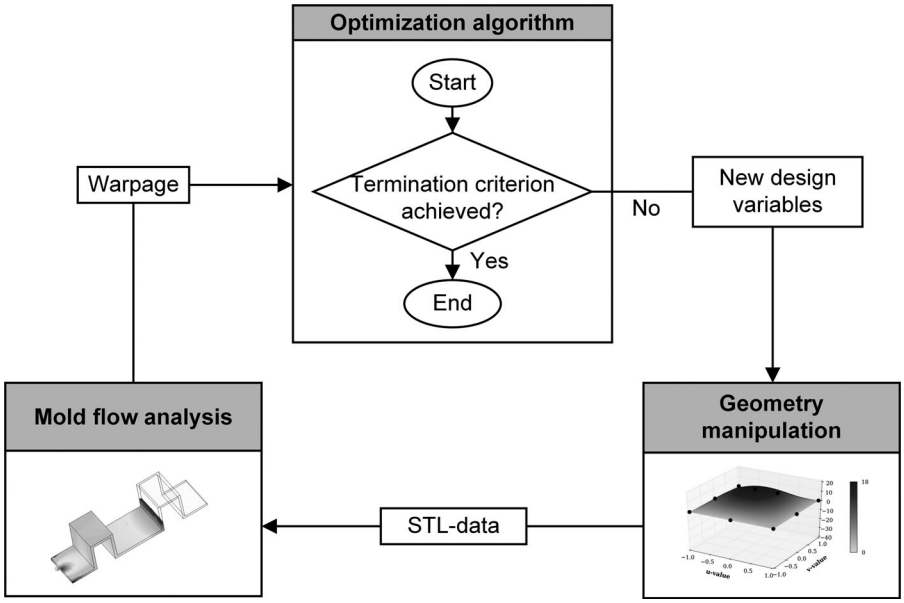

FIGURE 1. Flowchart of optimization procedure.

Another disadvantage of this method is that it places restrictions on the process window, which should be as wide as possible.

The gate location affects both filling and packing and thus the part warpage. Therefore, optimizing the gate location is another effective method for reducing the warpage.[16–21] However, in many cases, there are strong restrictions on the gate location, especially for parts with complex shapes or for molds with many moving elements. Therefore, this method of optimizing the gate location is not always useful.

Contrary to common design guidelines,[24] another effective approach for reducing the warpage is to optimize the wall thickness distribution because of its strong effects on the filling, holding, and cooling stages of the molding process. Lee and Kim[22] reduced the warpage of a housing part by varying the thickness of the part by discrete values at three different sections within defined bounds. The authors applied a modified complex method as an optimization algorithm to reduce the warpage by approximately 40%.

Despite impressive results and advances in computer-aided engineering, no subsequent or more detailed studies on optimizing wall thickness distribution have been published to date. One reason for the absence of such studies may be the low flexibility in changing the wall thickness of parts with complex shapes using few parameters and little effort. Another reason may be the difficulty of constructing an automatic optimization loop consisting of finite-element generation, mold flow analysis, and an optimization algorithm. In this study, the approach of Lee and Kim[22] is extended to more complex geometries. The user-defined geometry sections are parameterized to apply the variation in the wall thickness to different shape functions, which allows for large-scale variations using few variables. In this study, three different derivative-free state-of-the-art optimization algorithms, i.e., a genetic algorithm (GA), surrogate-based response surface method, and direct search method, are applied and compared to each other to optimize the variables.

Optimization Procedure

The optimization procedure, as illustrated in Fig. 1, consists of a mold flow analysis, which generates the warpage results, and a geometry manipulation tool, which modifies the initial design (Stereolithography-file) for the cavity depending on the optimization algorithm. This fully automated procedure is controlled by the optimization algorithm. Depending on the quality of the solution obtained using actual geometry, new design variables are generated and transferred to the geometry manipulation tool. After the geometry has been modified, the next mold flow analysis is performed. This procedure is repeated as many times as required until the solution converges or a termination criterion is reached.

To maintain practical process conditions during the optimization, the cooling and holding pressure times are changed in accordance with the actual wall thickness distribution. Fourier's one-dimensional law of heat conduction is used to calculate the modified values for the cooling time t_{cool} and holding pressure time t_{hold} of the actual design in a simplified manner as follows:

$$t_{act} = t_{ini}\left(\frac{s_{act}}{s_{ini}}\right)^2 \tag{1}$$

where t_{act} is the time (i.e., the holding pressure or cooling time) for the actual design, t_{ini} is the time for the initial design, and s_{act} and s_{ini} are the characteristic wall thicknesses of the actual and initial designs, respectively. The maximum wall thickness and wall thickness near the gate are chosen as the characteristic values for the modification of the cooling and holding pressure times, respectively. Sufficient metering is ensured by the constraint $t_{cool} \geq t_{hold} + 5s$. All of the other process variables remain

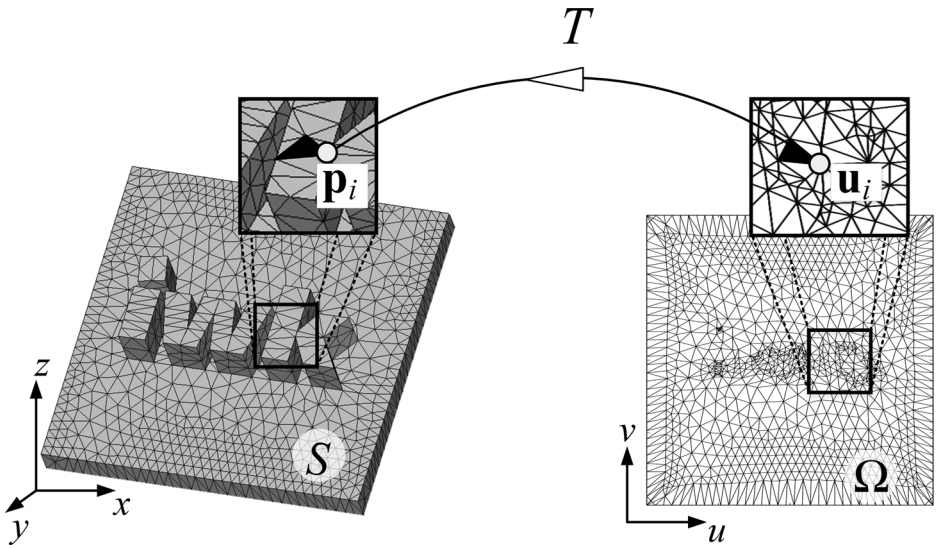

FIGURE 2. Parameterization of a triangle surface mesh.

unchanged, and their values are determined for the initial geometry considering the injection molding guidelines.[24]

GEOMETRY MANIPULATION

The geometry must be parameterized to implement the optimization algorithm, where wall thickness distribution is a function of several design variables. Depending on the computer-aided design (CAD) software for designing the parts, the wall thickness is typically modeled as either a Boolean combination of several solids (a constructive solid geometry) or of a collection of connected surface elements (a boundary representation). Therefore, variations in the wall thicknesses over large-scaled surface areas are limited and difficult to achieve using only a few variables, especially for complex geometries.[25]

For the aforementioned reasons and to remain independent from commercial CAD software, a Python-based software tool for automatic geometry manipulation is generated. Starting from a surface mesh for the initial design of the cavity, the software tool is able to change the wall thicknesses at user-defined surface areas with a low level of computational effort.

The algorithm for geometry manipulation is based on a shape-preserving surface parameterization following Floater,[26] which is widely used for many applications in computer graphics and geometric modeling, such as texture mapping, shape morphing, surface reconstruction and repairing, and grid generation.[27] Figure 2 illustrates the surface parameterization $T : \Omega \rightarrow S$, which describes a one-to-one mapping between a triangle $\mathrm{t} = [\mathbf{u}_i, \mathbf{u}_j, \mathbf{u}_k]$ on the two-dimensional parametric plane $\Omega \subset \Re^2$ to the corresponding triangle $\mathrm{T} = [\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k]$ of the real three-dimensional (3D) geometry $S \subset \Re^3$.

The computation of the parameterization is straightforward and comparable to a simplified finite element method. Replacing the edges of the triangles with springs that are connected at the vertices and fixing some of the vertices at the outer bound allow the positions of the inner vertices, $\mathbf{u}_i$, to be found by minimizing the total potential energy, E, as follows:

$$E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in N_i} \frac{1}{2} D_{ij} \| \mathbf{u}_i - \mathbf{u}_j \|^2 \tag{2}$$

where D_{ij} represents the spring constants between the inner vertex $\mathbf{u}_i$ and its neighboring vertices $\mathbf{u}_j$. Minimizing E yields a simplified expression in which every interior vertex $\mathbf{u}_i$ is an affine combination of its neighbors, $\mathbf{u}_j$, as follows:

$$\mathbf{u}_i = \sum_{j \in N_i} \lambda_{ij} \mathbf{u}_j \tag{3}$$

where λ_{ij} are normalized coefficients that sum to unity. Separating the parameter points for the interior ($j \leq n$) and boundary vertices ($j > n$) in the sum on the right-hand side of Eq. (3) produces the following linear system of equations:

$$\mathbf{u}_i - \sum_{j \in N_i, j \leq n} \lambda_{ij} \mathbf{u}_j = \sum_{j \in N_i, j > n} \lambda_{ij} \mathbf{u}_j \tag{4}$$

The computation of the coordinates $\mathbf{u}_i$ and $\mathbf{v}_i$ of the interior parameter points requires the solution of two sparse linear systems of dimension $n \times n$, where n is the number of interior points. The definition of the normalized coefficients plays a major role in preventing the unfolding of triangles and singularities. Therefore, Floater's approach is used, in which barycentric coordinates are generalized in accordance with the mean value theorem.[26] This procedure leads to the following geometrical definition for the normalized coefficients λ_{ij}:

$$\lambda_{ij} = \frac{w_{ij}}{\sum_{j=1}^{k} w_{ij}}, \quad w_{ij} = \frac{\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2)}{\| \mathbf{u}_i - \mathbf{u}_j \|} \tag{5}$$

The configuration of the angles α_i and α_{i-1} of the triangles is illustrated in Fig. 3.

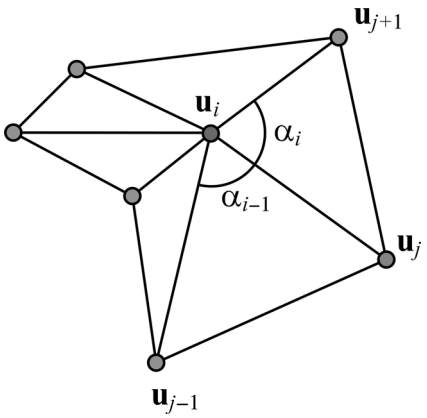

FIGURE 3. Arrangement of angles and indices on a star-shaped polygon.

The parameterization of the unit square is completed in two primary steps. First, the vertices on the exterior board of the element selection $S_i \subseteq S$ on the 3D mesh must be fixed on the surrounding of the unit square. Therefore, four edge vertices are defined, between which equidistant vertices are selected. This information is then used to construct a linear system (Eq. (4), which can be solved using special sparse algebra computation software.

This procedure differs from conventional parameterization, i.e., the pixels of a texture are mapped to the 3D mesh S, because the changes in the wall thickness are mapped from the parametric plane Ω to the 3D mesh S (Fig. 4). Thus, a generalized polynomial function in two dimensions, u and v, is used to model the distribution of the changes in the wall thickness, $\Delta z(u,v)$, on the unit square in the parametric plane:

$$\Delta z(u, v) = \sum_{l=0}^{n_p} \sum_{m=0}^{n_p} C_{l,m} u^l v^m \tag{6}$$

where $C_{l,m}$ are the polynomial coefficients and n_p is the polynomial degree. For simplification, instead of using the variables $C_{l,m}$, equivalent supporting points $\mathbf{P}_i$ with discrete values for the change in the thickness $\Delta z_{P,i} = \Delta z(\mathbf{P}_i)$ at the specific location were systematically spread over the unit square and fitted by a polynomial (Eq. (6). As an example, the locations of the supporting points $\mathbf{P}_i$ for polynomial degrees of $n_p = 1$ and $n_p = 2$ are illustrated in Fig. 4.

Using a one-to-one mapping of the parameterization, the changes in the wall thickness $\Delta z(\mathbf{u}_i)$ at each vertex $\mathbf{u}_i$ can be transferred to their corresponding locations on the 3D mesh $\mathbf{p}_i$. The normalized averages of the surrounding face normal vectors, $\vec{n}_i$, are used as the directions to apply the change, which leads to the following expression:

$$\mathbf{p}_i^* = \mathbf{p}_i + g\Delta z(\boldsymbol{u}_i)\vec{n}_i \quad g = \cos(\gamma_i/2)^{-1} \tag{7}$$

where $\mathbf{p}_i^*$ is the new position of the vertex, $\mathbf{p}_i$ is the position of the vertex on the initial mesh, and g is a scaling factor that accounts for the preservation of the edges according to the angle γ_i between the element surface normal of the two edge-building elements.

The developed approach yields a parameterized geometry for which the wall thickness distribution at the user-defined surface sections can be varied through the z-location $\Delta z_{P,i}$ at each supporting point $\mathbf{P}_i$. Thus, the design vector $\mathbf{x}$ can be expressed in terms of its associated design variables as $\mathbf{x} = \{\Delta z_{P,i}, \ldots, \Delta z_{P,n}\}$.

MOLD FLOW SIMULATION WITH CADMOULD 3D-F CMV6

The requirements for incorporating the mold flow simulation into the optimization procedure are as follows: (1) a low computational effort should be required and (2) the simulation should be able to interact with the optimization algorithm. The Cadmould 3D-F CMV6 from "Simcon kunststofftechnische Software GmbH" in Germany fulfills both requirements. The high computational speed is based on the company's patented 3D framework technology,[28] which solves the equations for the conservation of mass (Eq. (8), impulse (Eq. (9), and energy (Eq. (10) along and between the surfaces in a simplified manner following the Hele–Shaw procedure.[29] The equations solved by this method are as follows:

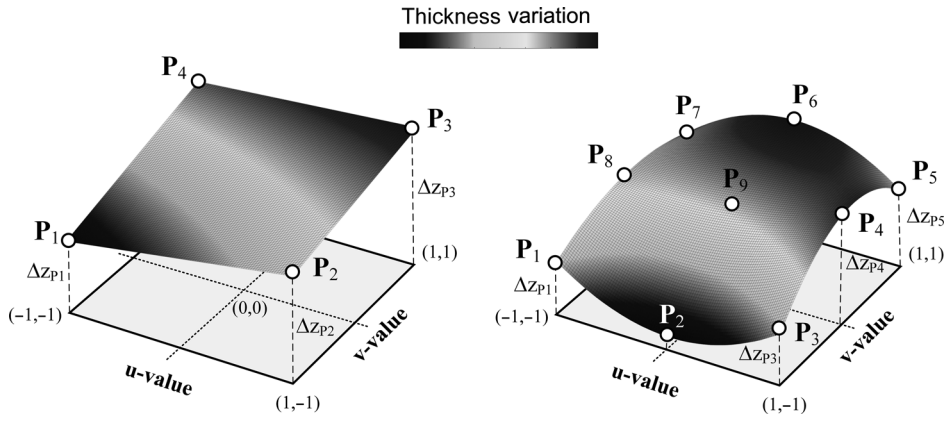

FIGURE 4. Location of supporting points $\mathbf{P}_i$ on the parameterization plane for polynomial degrees of $n_p = 1$ (left) and $n_p = 2$ (right).

$$\frac{\partial}{\partial x}\left(\int_0^h \rho v_x dz\right) + \frac{\partial}{\partial y}\left(\int_0^h \rho v_y dz\right) = \int_0^h \frac{d\rho}{dt} dz \tag{8}$$

$$\frac{\partial}{\partial z}\left(\eta \frac{\partial v_x}{\partial z}\right) - \frac{\partial p}{\partial x} = 0, \quad \frac{\partial}{\partial z}\left(\eta \frac{\partial v_y}{\partial z}\right) - \frac{\partial p}{\partial y} = 0 \tag{9}$$

$$\rho c_p \left(\frac{\partial T}{\partial t} + v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y}\right) = k_{th}\left(\frac{\partial^2 T}{\partial z^2}\right) + \eta\left[\left(\frac{\partial v_x}{\partial z}\right)^2 + \left(\frac{\partial v_y}{\partial z}\right)^2\right] \tag{10}$$

where x and y are planar coordinates, z is a gapwise coordinate, t is the time, v_x and v_y are the velocity components in the x and y directions, respectively, T is the temperature, p is the pressure, η is the shear viscosity, ρ is the density, c_p is the specific heat, and k_{th} is the thermal conductivity.

A viscous thermoelastic approach is used to calculate the residual stresses σ_{ij} and the resulting deformations for an unreinforced polymer, where the stresses σ_{ij} are zero above the nonflow temperature T_{NF}. The following constitutive equation is applied at temperatures lower than the nonflow temperature:

$$\sigma_{ij} = -\int_0^t K\left(\alpha_v \frac{\partial T}{\partial t'} - \frac{\partial \varepsilon_{kk}}{\partial t'}\right) dt' + 2G\int_0^t \frac{\partial}{\partial t'}\left[\varepsilon_{ij} - \frac{\delta_{ij}\varepsilon_{kk}}{3}\right] dt' \tag{11}$$

where K is the compression modulus, α_v is the coefficient of volumetric thermal expansion, G is the shear modulus, ε_{ij} are the strain components, δ_{ij} is Kronecker's delta, and ε_{kk} is the trace of the strain tensor. The compression modulus, coefficient of volumetric thermal expansion, and nonflow temperature are taken from the measurements of the specific volume, which depends on the pressure and temperature. The shear modulus is considered a function of temperature only.

The part is clamped inside the mold during the molding process. Thus, inherent thermal stresses accumulate, resulting in the spontaneous deformation of the part after it is ejected from the mold. While the part cools outside the mold, additional thermal stresses are induced that change the state of deformation such that the total potential energy is minimized; the deformation is calculated using a finite-element approach.

This process is a simplification of real viscoelastic behavior but is sufficiently accurate, especially for amorphous polymers.[2] The part warpage can be determined after the part deformations are calculated. The warpage is typically defined as the geometrical difference between the molded part and the isotropic shrunk cavity. This difference depends on the alignment between the part and cavity, which is calculated using a best-fit algorithm based on the iterative-closest-point (ICP) method.[30]

A more meaningful definition of the warpage is used in the formulation of an optimization algorithm, where only out-of-plane deformations resulting from the change in the surface normals during the molding process are evaluated (Fig. 5). Therefore, the angle $\phi_{w,i}$ between the element surface normal for the cavity, $\vec{n}_{c,i}$, and its direction after molding, $\vec{n}_{m,i}$, for each element

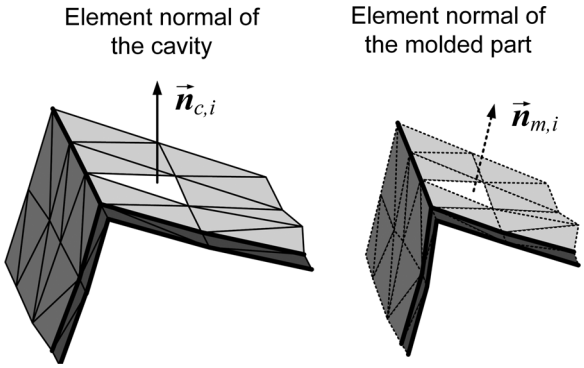

FIGURE 5. Illustration showing the definition of warpage.

i of an arbitrary user-defined element selection, $S_W \subseteq S$, of the part is calculated (Eq. (12) as follows:

$$\varphi_{w,i} = \angle\left[\vec{n}_{c,i}, \vec{n}_{m,i}\right]. \tag{12}$$

The scalar product is used such that the angles are always positive and can thus be used for minimization purposes.

DEFINITION OF THE OPTIMIZATION PROBLEM

The objective of the optimization procedure is to minimize the warpage over an arbitrary user-defined element selection, S_W. Therefore, two meaningful measures are calculated:

$$\bar{\varphi}_w = \frac{1}{n_{el}}\sum_{i=1}^{n_{el}} \varphi_{w,i} \quad \text{and} \quad \varphi_{w,\max} = \max\left\{\varphi_{w,i}\right\}, \tag{13}$$

where $\bar{\varphi}_w$ is the average change in the surface normal during molding and $\varphi_{w,\max}$ is the corresponding maximum change. The sum of both values is used as the objective function, F_w, in the optimization procedure as follows:

$$F_w = \frac{\bar{\varphi}_w}{\bar{\varphi}_w^0} + \frac{\varphi_{w,\max}}{\varphi_{w,\max}^0} \tag{14}$$

To obtain the same order of magnitude for both measures, each measure is divided by its corresponding resulting value for the initial geometry $\bar{\varphi}_w^0$ and $\varphi_{w,\max}^0$. Thus, the optimization problem can be stated as follows:

$$\min_{\boldsymbol{x}}\left\{F_w(\boldsymbol{x})\right\} \tag{15}$$

To limit the range of variation in the wall thickness and to avoid mesh penetrations that yield wall thicknesses near zero, each entry of the design vector $\mathbf{x}$ is constrained as follows:

$$\Delta z_{\min,i} \le \Delta z_{P,i} \le \Delta z_{\max,i} \quad i = 1 \ldots (n_p + 1)^2 \tag{16}$$

where $\Delta z_{\min,i}$ is the minimum z-location of the supporting point, $\mathbf{P}_i$, and $\Delta z_{\max,i}$ is the maximum valid value of the same point.

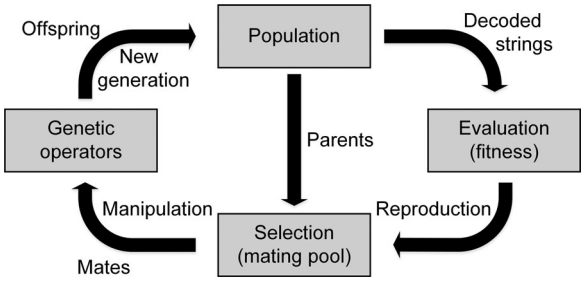

FIGURE 6. GA cycle.[34]

APPLIED OPTIMIZATION ALGORITHMS

The primary difficulty in the present optimization problem is that there is no analytical model available to describe the correlation between the design vector **x** and objective function F_w. Thus, information on the local gradients cannot be easily obtained and is not sufficiently accurate. Derivative-free iterative optimization (DFO) algorithms are required to solve this type of black-box optimization problem. DFO is independent of the derivative information, making it robust against small numerical fluctuations in the objective function because of infinitesimal changes in the design variables.[31]

The DFO algorithms can be divided into three primary groups: (1) random search methods, (2) deterministic sampling methods, and (3) surrogate-based optimization (SBO) methods.[31] To evaluate the different effects of these methods on the present optimization problem, a representative algorithm from each group that can perform global optimization under specified bounds is implemented. A GA, direct search method, and SBO method are considered and compared to each other.

Genetic Algorithm

A GA is a metaheuristic optimization strategy that was developed by Holland in 1975.[32] This algorithm is a simplification of biological evolution and has been successfully applied to several optimization problems.[33] This algorithm can parallelize and requires only a few parameters to be set. However, these advantages are offset by the high number of function evaluations that are required.[31] As illustrated in Fig. 6, a GA consists of a set of individuals, called the population and a set of evolutionary operations through which the population passes in each cycle. As in nature, only well-adapted individuals survive and bequeath their genes to their offspring. In view of the present problem, each individual corresponds to a design vector **x** with a set of design variables $\Delta z_{P,i}$ that are encoded in binary strings.

A GA operates iteratively through a simple cycle of stages (Fig. 6): (1) a population of strings is created, (2) each string is evaluated, (3) the best string is selected, and (4) genetic manipulation is performed to create a new population. The primary processes of genetic manipulation are crossover and mutation, for which the binary strings of two selected individuals at a random position are recombined or inverted (Fig. 7). The GA cycle runs until a termination criterion is reached. In addition, the convergence behavior of the GA is affected by the probabilities of crossover p_c and mutation p_m and the size of the population s_p.

Direct Search Method

Powell's nonlinear constrained optimization[35] by the linear approximations approach (COBYLA) is used as a direct search method. This iterative procedure starts with an initial simplex of dimension $(n + 1)$, where n is the size of the design vector **x**, and linear approximations to the objective and constraint functions are constructed by interpolation at the vertices. A trust region bound restricts each change in the variables. Thus, a new vector of variables is calculated that can replace one of the current vertices using a merit function based on the greatest constraint violation.[35] The method is controlled by two values, the initial value, ρ_{beg}, and the final value, ρ_{end}, of the trust region radius, ρ.

Surrogate-Based Optimization

SBO has been shown to be an effective approach in the design of computationally expensive models.[36] The primary objective of SBO is to capture the primary characteristics of the real objective function $F_R(x)$ in terms of a surrogate-based approximation $F_A(x)$ (Fig. 8), which can be minimized with low computational effort.[31]

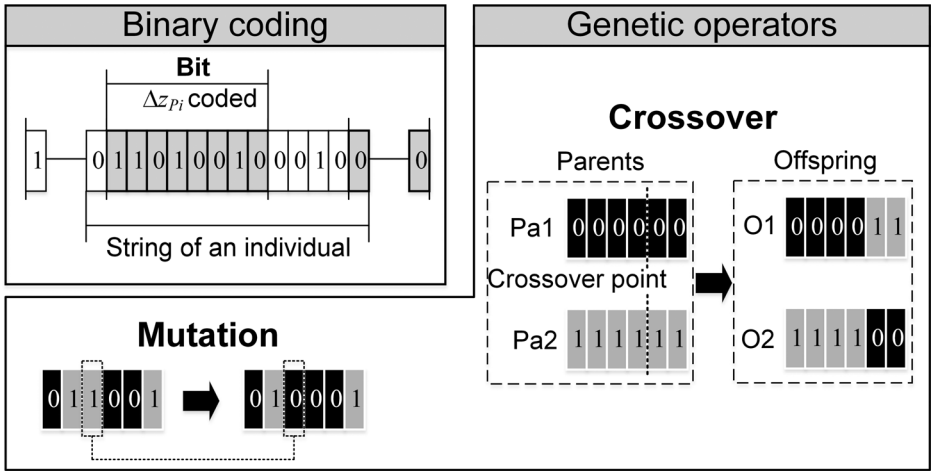

FIGURE 7. Binary representation of design variables and genetic operations during a GA cycle.

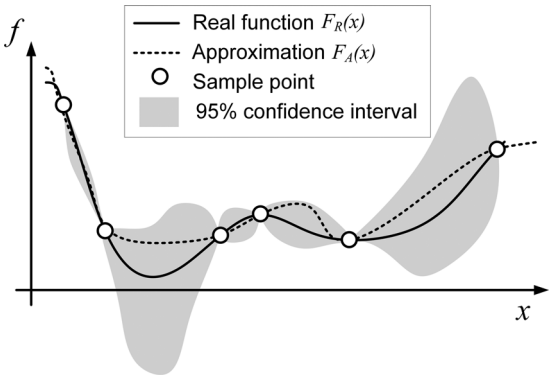

FIGURE 8. Surrogate-based optimization.

The key steps of SBO are as follows: (1) the generation of sampling points, (2) the evaluation of sampling points (simulation), (3) the construction of a surrogate model, (4) model validation, and (5) model minimization. Several approaches exist for constructing a surrogate model.[36] In the present study, the Kriging method,[37] which is more widely known as the Gaussian process (GP) method, is used to produce smooth surface fit models. The real objective function, $F_R(\mathbf{x})$, is replaced by an approximation, $F_A(\mathbf{x})$, which is based on a polynomial trend basis function, $f(\mathbf{x})$, for the global adjustment and a Gaussian process, $Z(\mathbf{x})$, for the local adjustment as follows:

$$F_R(\mathbf{x}) = F_A(\mathbf{x}) = \sum_{i=0}^{k} \beta_i f_i(\mathbf{x}) + Z(\mathbf{x}) \tag{17}$$

where k depends on the polynomial degree and β_i represents the regression coefficients. The mean of $Z(\mathbf{x})$ is zero, and its variance is $\sigma_z{}^2$. The spatial correlation function between two design vectors $\mathbf{x}^1$ and $\mathbf{x}^2$ of size n is given as follows:

$$R(\theta, q, \mathbf{x}^1, \mathbf{x}^2) = \prod_{j}^{n} \exp\left[-\theta_j \left|\mathbf{x}_j^1 - \mathbf{x}_j^2\right|^q\right]. \tag{18}$$

The unknowns θ_j and q are estimated using the maximum likelihood estimation (MLE) procedure proposed by Sacks et al.[38] Once the model is built using the existing sampling vectors, the next step is to find the minimum of the surrogate model. Therefore, the expected improvement (EI) approach proposed by Mockus et al.[39] is used. If improvement is expected, new variables are defined, the real improved value is evaluated, and the surrogate model is updated. This iterative procedure is a balance between a local and global search and ideally leads to a global minimum.[40]

Realization and Dataflow

The first step of the developed optimization procedure is the parameterization of the initial geometry and is similar to a preprocessing procedure (Fig. 9). A python script reads the initial geometry data with the recognized element selections for the optimization S_i and calculates the data's parameterization. The parameterization is executed only once, and the resulting data are stored in a separate file.

After the parameterization is completed, the primary optimization process is initiated. For this purpose, the previously determined parameterization data, the polynomial degrees, the bounds on the thickness variations, the variables for the process simulation, and the control parameters of the optimization algorithm are read once at the beginning. Then, the optimization starts, and the design variables are optimized according to the implemented optimization algorithm.

An important issue for achieving low numerical scatter in the objective function is the preservation of the initial mesh structure

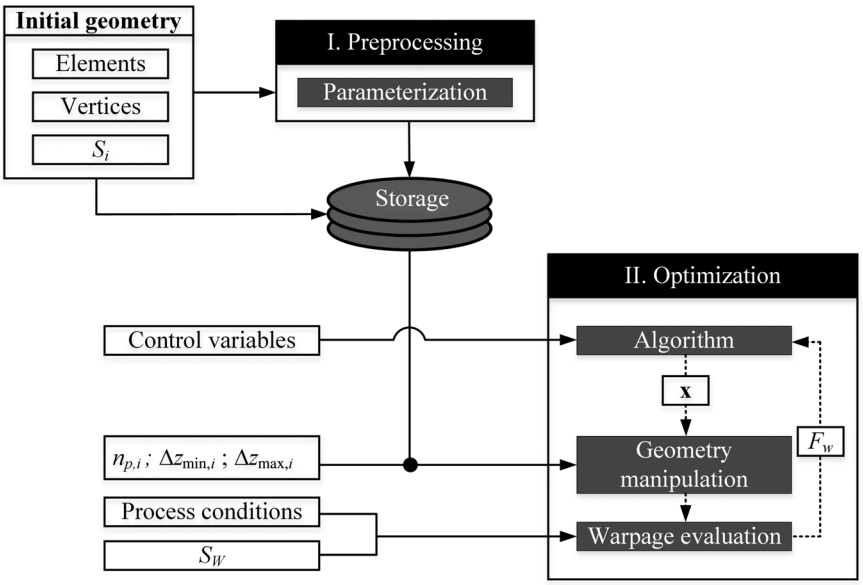

FIGURE 9. Dataflow of the developed optimization procedure.

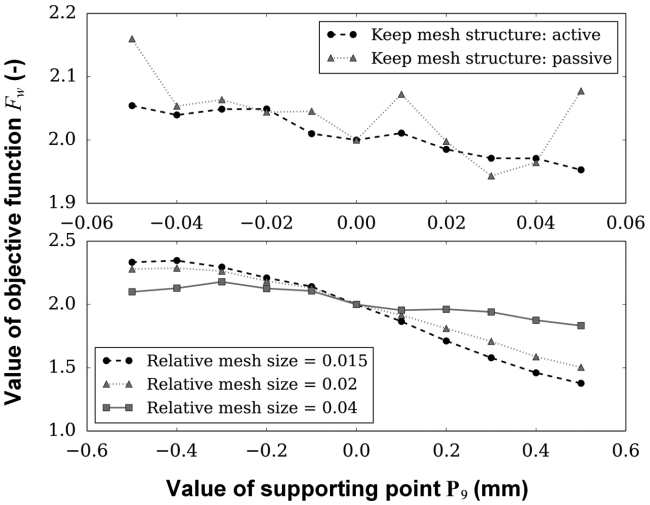

FIGURE 10. Effect of preserved mesh structure (top) and mesh size (bottom) on the objective function when design variable $\mathbf{P}_9$ is changed at different length scales.

at each remeshing cycle during the optimization procedure (see the top of Fig. 10). The size of the finite element mesh is another key factor. The goal is to find the optimal ratio between the accuracy and evaluation time. Figure 10 provides an example of the effect of three different mesh sizes on the objective function on the developed warpage demonstrator for the variation in the design variable $\mathbf{P}_9$. Figure 10 illustrates that a mesh size of 2% of the space diagonal of the geometry is sufficient to ensure a realistic trend of the objective function. Therefore, a mesh size of 2% is recommended.

IMPLEMENTED ALGORITHMS

The python-coded evolutionary computation framework Pyevolve is used as a representative GA. Therein, a rank-based selection is applied as the selection method. Alleles are used for each design variable to ensure that the upper and lower bounds on each variable are maintained. Elitism is considered to be the preservation of the fittest individual during evolution. Following recommendations in the literature,[41] the probability for one point crossover is set at 90%, and the termination criterion is that 100 generations are completed. The population size and mutation probability are varied in this study.

COBYLA is implemented in Scipy, the scientific python package. The trust region radiuses are set as $\rho_{\text{beg}} = 0.1$ and $\rho_{\text{end}} = 0.005$. The algorithm is initiated by the initial geometry, which results in a design vector $\mathbf{x}$ with zeroed entries.

The SBO is implemented from the design analysis kit for optimization and terascale applications (DAKOTA) project of Sandia National Laboratories. Here, random Latin hypercube sampling (LHS) is applied to generate the initial samplings. A reduced quadratic polynomial function is used for the global basis function. The dividing rectangle (DIRECT) algorithm is used to minimize of the objective function using the expected improvement approach.

All of the optimizations are executed on a 3.5-GHz Intel-i7 computer consisting of 8 cores and 16 GB of RAM.

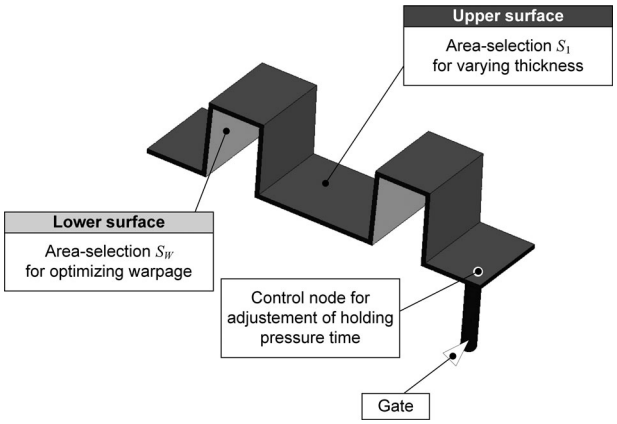

FIGURE 11. Initial cavity of the warpage demonstrator and area sections for optimization.

TABLE I
Process Conditions for the Warpage Demonstrator

Process Variable	Value
Injection speed (cm^3/s)	10.25
Postfill time (cooling and packing) (s)	22
Packing time (s)	16
Packing pressure (MPa)	50
Melting temperature (°C)	260
Surface temperature of cavity (°C)	60

Application to a Warpage Demonstrator

INPUT DATA

First, the capability of the presented optimization procedure is verified for a warpage demonstrator with dimensions of 124 mm × 30 mm × 30 mm. Figure 11 presents the initial geometry of the mold cavity, which has a constant wall thickness of 1.5 mm, the area section for the optimization S_1, and the area section for the warpage evaluation S_W. A polynomial degree of $n_p = 2$ and changes of 0.5 mm in both the positive and negative directions are mathematically defined as follows:

$$-0.5 \leq \Delta z_{\text{P,i}} \leq 0.5 \quad i = 1 \ldots (n_p + 1)^2. \tag{19}$$

ASA Luran S 757R is employed as the polymer material. Table I lists the process conditions that are determined based on the initial cavity geometry.

RESULTS AND DISCUSSION

Influence of the GA Settings

The convergence behavior of the GA is significantly influenced by the probability of mutation p_m and the size of population p_s.[41] First, these features are considered to determine the most practical values. Figure 12 illustrates the optimization history for four different probabilities for mutations of 0.02, 0.05,

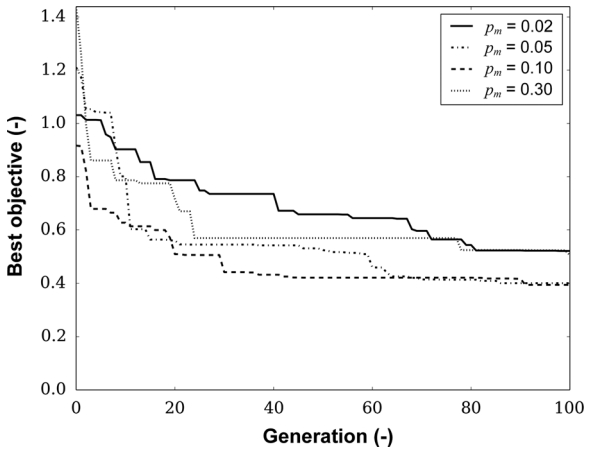

FIGURE 12. Effect of different probabilities for mutation p_m on the optimization history for the warpage demonstrator.

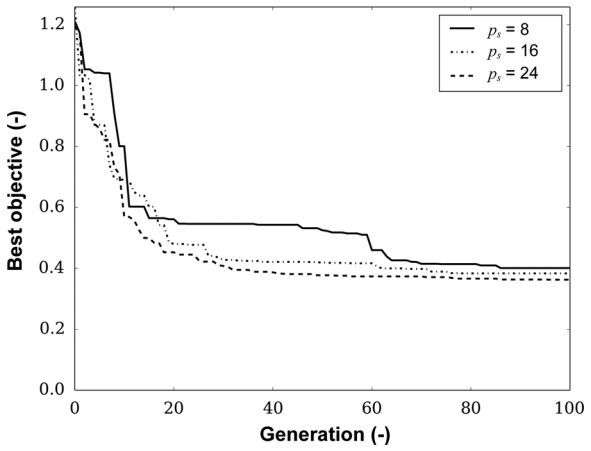

FIGURE 13. Effect of different population sizes p_s on the optimization history for the warpage demonstrator.

0.1, and 0.3 and a fixed population size of eight individuals. Thus, the values of the objective function after 100 generations are considerably lower for mutation rates of 0.05 and 0.1. The lowest probability for mutation yields the lowest convergence speed, whereas the highest probability only produces major improvements during the first 20% of the optimization process. These observations are meaningful because mutation produces diversity in the population. If the probability is set at an overly low value, then the algorithm tends to fall into a sector with a local minimum, whereas a high probability generates ongoing new solutions that hinder convergence toward an optimum.

Figure 13 illustrates the effect of different population sizes on the optimization history for a fixed mutation probability of 0.05. Doubling or tripling the population results in slightly lower objective function values for the best individual. Doubling the population reduces the minimum objective function value from 0.401 to 0.383, and tripling the population reduces the value from 0.401 to 0.363. These improvements are obtained at the cost of a higher computational effort, which increases almost linearly with the population size. For practical applications, it is useful to consider whether the same result can be achieved with a smaller number of function calls, i.e., by increasing the population size. For a population size of eight individuals, the best function value of 0.401 is reached by a population size of 24 individuals after 33 generations. This result corresponds to a reduction of eight function calls. In contrast, a population of 16 individuals requires 63 generations to achieve a function value of 0.401, which is 208 function calls more than for a population of eight individuals. Thus, in terms of function calls, the efficiency for a population size of 8 and 24 individuals is approximately equal.

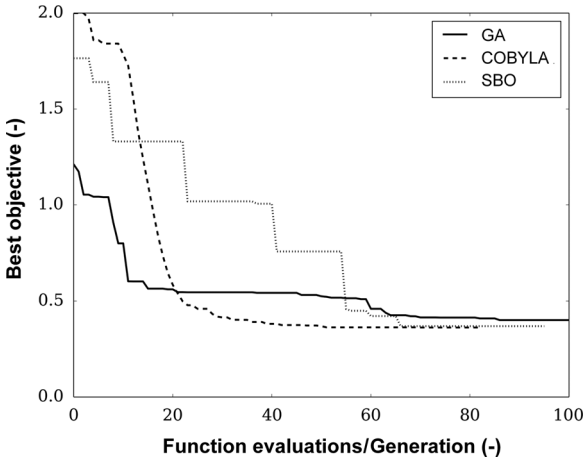

FIGURE 14. Effect of different optimization algorithms on the optimization history for the warpage demonstrator.

Comparison of the Different Optimization Algorithms

In the next step, the optimization results from the different optimization algorithms are considered. Figure 14 presents the optimization history for the three algorithms: the SBO, COBYLA, and GA, where the GA is based on a population size of eight individuals and a mutation probability of 0.05. For both the SBO and COBYLA, the results for the best objective function values are plotted versus the numbers of function evaluations, whereas for the GA, the best individuals are plotted versus the number of generations.

Each optimization algorithm leads to an impressive reduction in the objective function value compared to that obtained with an initial geometry of 2.0. The GA results in a slightly higher value of 0.401 than the values of 0.365 and 0.370 obtained using the COBYLA and SBO, respectively (Table II). The COBYLA terminates after 84 iterations of function calls, whereas the SBO terminates after 96 calls and the GA terminates after 800 calls. Whereas the optimization process for the COBYLA and GA begins at the first function call, the primary optimization process of the SBO begins after the completion of the Latin LHS of 55 observations.

The final values for the best objective functions and design variables are summarized in Table II. Except for $\Delta z_{P,3}$, $\Delta z_{P,4}$ and $\Delta z_{P,5}$, all of the design variables converge to similar values. This behavior can be explained by the comparably low influence of

TABLE II
Comparison of the Optimization Results

Parameter	GA	COBYLA	SBO
Best objective function value (−)	0.401	0.365	0.370
Function calls (−)	800	84	96
Final Design Variables			
$\Delta z_{P,1}$ (mm)	0.494	0.500	0.494
$\Delta z_{P,2}$ (mm)	0.473	0.500	0.494
$\Delta z_{P,3}$ (mm)	**−0.071**	**−0.258**	**−0.494**
$\Delta z_{P,4}$ (mm)	**−0.023**	**0.062**	**−0.173**
$\Delta z_{P,5}$ (mm)	**−0.471**	**−0.127**	**−0.395**
$\Delta z_{P,6}$ (mm)	−0.500	−0.500	−0.494
$\Delta z_{P,7}$ (mm)	−0.473	−0.500	−0.395
$\Delta z_{P,8}$ (mm)	0.468	0.500	0.494
$\Delta z_{P,9}$ (mm)	0.454	0.500	0.494

the three aforementioned variables on the objective function. Most of the design variables converge to both the upper and lower limits, which confirms that variable thicknesses are required to lower the warpage of the part.

Visualization of the Optimization Results

To better understand and classify the obtained optimization results, plots of the deformations and changes in the wall thickness are compared to each other.

The warpage plot in Fig. 15 presents the results as a graphical one-to-one comparison between the initial and SBO-optimized geometries. To facilitate the interpretation of the results, the typical warpage definition of the mold flow simulation program with the geometric difference between the isotropic shrunk geometry for the cavity and simulated part is used instead of the self-defined out-of-plane values from Eq. (12). All of the geometries are plotted using an identical color range and identical scaling factor of 10 for the resulting deformations. The side view of the deformed geometries illustrates the decrease in the warpage following the optimization of the wall thickness distribution along the flow path. In particular, the large deviations from the rectangular shape far from the gate are nearly eliminated. For all three algorithms, the resulting deformations are nearly identical.

Table III illustrates that the maximum warpage is reduced by the different optimization algorithms from 0.961 mm to 0.211 mm (GA), 0.192 mm (COBYLA), and 0.197 mm (SBO). All optimized geometries lead to increases in the volume and postfill time of approximately 15% to 17% and 38% to 44%, respectively, and decreases in the injection pressure of 24% to 27%. Considering the environmental aspects characterized by material usage, injection pressure, and postfill time, the GA-optimized geometry achieves the best result compared with the other optimized geometries.

Figure 16 presents the changes in the wall thickness between the initial and optimized designs. Here, the dark-colored area sections correspond to an increase in the wall thickness, and the light-colored regions correspond to a reduction in the wall thickness. Specific values are marked in the figure to illustrate these results. Each of the three algorithms forces the thickness to increase near the gate up to approximately 66% of the flow path, causing the thickness toward the end of the fill to decrease.

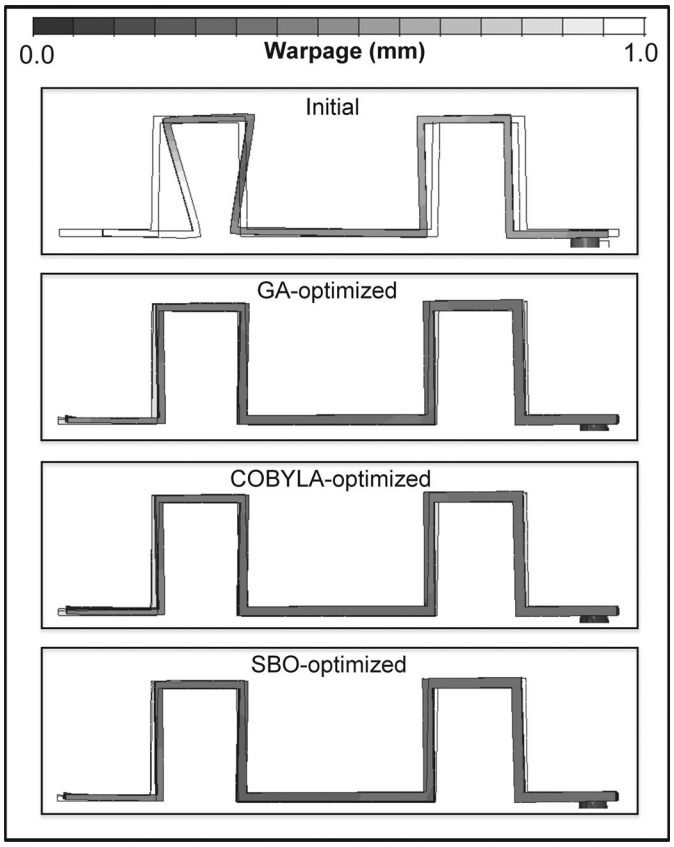

FIGURE 15. Warpage for the initial and the optimized designs.

TABLE III
Discrete Optimization Results for the Warpage Demonstrator

Parameter	Initial	GA	COBYLA	SBO
Maximum warpage (mm)	0.961	0.211	0.192	0.197
Volume (cm^3)	10.27	11.77	12.04	11.76
Postfill time (s)	22.0	30.4	31.2	31.6
Injection pressure (MPa)	82.3	59.8	62.3	61.2

The wall thickness of the three algorithms differs most at the upper side, which can be explained by the differences in the design variables $\Delta z_{P,3}$, $\Delta z_{P,4}$ and $\Delta z_{P,5}$ (Table II). The increase in the wall thickness near the gate improves the pressure transmission, which is accompanied by a reduction in the shrinkage. The reduction in the wall thickness far from the gate can be attributed to the smaller frozen-in layer at the end of filling.[42] This behavior leads to more balanced solidification and a reduction in the thermally induced stresses. Thus, the warpage is lowered.

For comparison, this improvement cannot be achieved only by an increase in the wall thickness of +0.5 mm. In this case, the objective function has a value of 1.161, which is approximately three times higher than the optimized function values.

Process Robustness of the Optimized Geometries

An optimized wall thickness distribution is not useful if it decreases the robustness of the injection-molding process. Thus, the

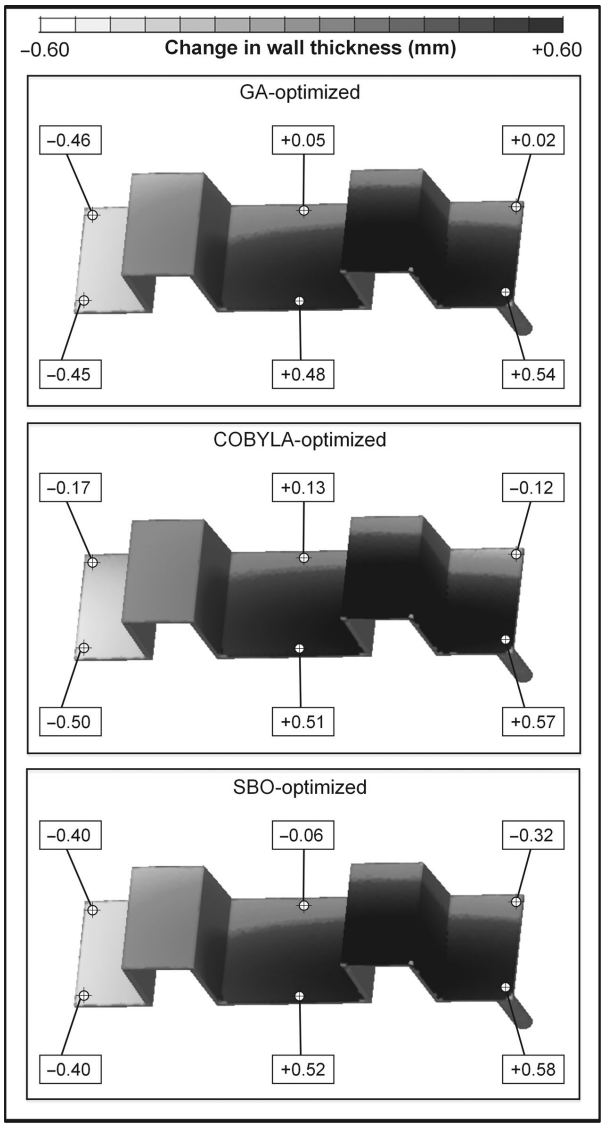

FIGURE 16. Changes in wall thickness for the warpage demonstrator from the optimization procedure using different optimization algorithms.

TABLE IV
Mean and Standard Deviation (STD) of the Process Conditions for Checking the Process Robustness

Parameter	Mean	STD
Melting temperature (°C)	260	13
Mold surface temperature (°C)	60	6
Holding pressure (MPa)	50	5
Injection speed (cm^3/s)	10.25	1.025

robustness is verified for the initial and optimized geometries. A randomized Latin LHS of 200 observations is applied, where the melting temperature, mold surface temperature, packing pressure, and injection speed are varied using a normal distribution; the mean and standard deviation (STD) of which are shown in Table IV.

The results are summarized in Fig. 17, where the normalized distributions of the average change in the surface normal angle during the molding process, $\bar{\varphi}_w$, (Eq. (13)) are shown. To quantify the results, specific values for the mean, standard deviation, maximum and minimum are denoted. These results demonstrate that the variances for the optimized geometries are considerably smaller than those for the initial geometry. Of the optimized geometries, the SBO optimization results showed the smallest variance.

Next, the developed optimization procedure is applied to a real industrial part.

Application to an Industrial Part

The practicability of the present optimization approach is verified for a real industrial part with dimensions of 260 mm × 225 mm × 125 mm. Figure 18 presents the initial geometry of the mold cavity, the two area sections for the thickness variations, the predefined edge nodes for the parameterization, the gate location, and the control node for adjusting the holding pressure time.

For each section, a polynomial degree of $n_p = 2$ and upper and lower bounds of –0.7 and 0.7 mm, respectively, are defined for changes in the thickness of each design variable. These settings result in a design vector $\mathbf{x}$ with a dimension of 18. During the subsequent assembly process, the housing undergoes a welding process in which a second part is attached to the upper rim of the housing. Therefore, the warpage of the upper rim should be as low as possible and is selected as the objective function for the optimization (Fig. 18).

ABS Terluran GP-35 was employed as the polymer material. Table V presents the process conditions that were determined for the initial geometry for the cavity. The predetermined mesh size of 2% of the space diagonal of the geometry corresponds to a finite element mesh consisting of approximately 24,500 triangle elements, which leads to a low processor operating grade with a computational time of approximately 5 min for one warpage evaluation.

For comparison purposes, the three optimization algorithms, the COBYLA, SBO, and GA, are applied using the same settings as to the warpage demonstrator.

RESULTS AND DISCUSSION

Preprocessing Results

The mesh parameterization of section 1 is illustrated in Fig. 19. Floater's mean value approach achieves a parameterization that nearly preserves the shape of the triangular elements. This result leads to a low distortion in the applied thickness distribution.

Optimization Results

The resulting optimization histories of the three different optimization algorithms are shown in Fig. 20, where the objective function for the best result is plotted against the number of function evaluations and the number of generations. The

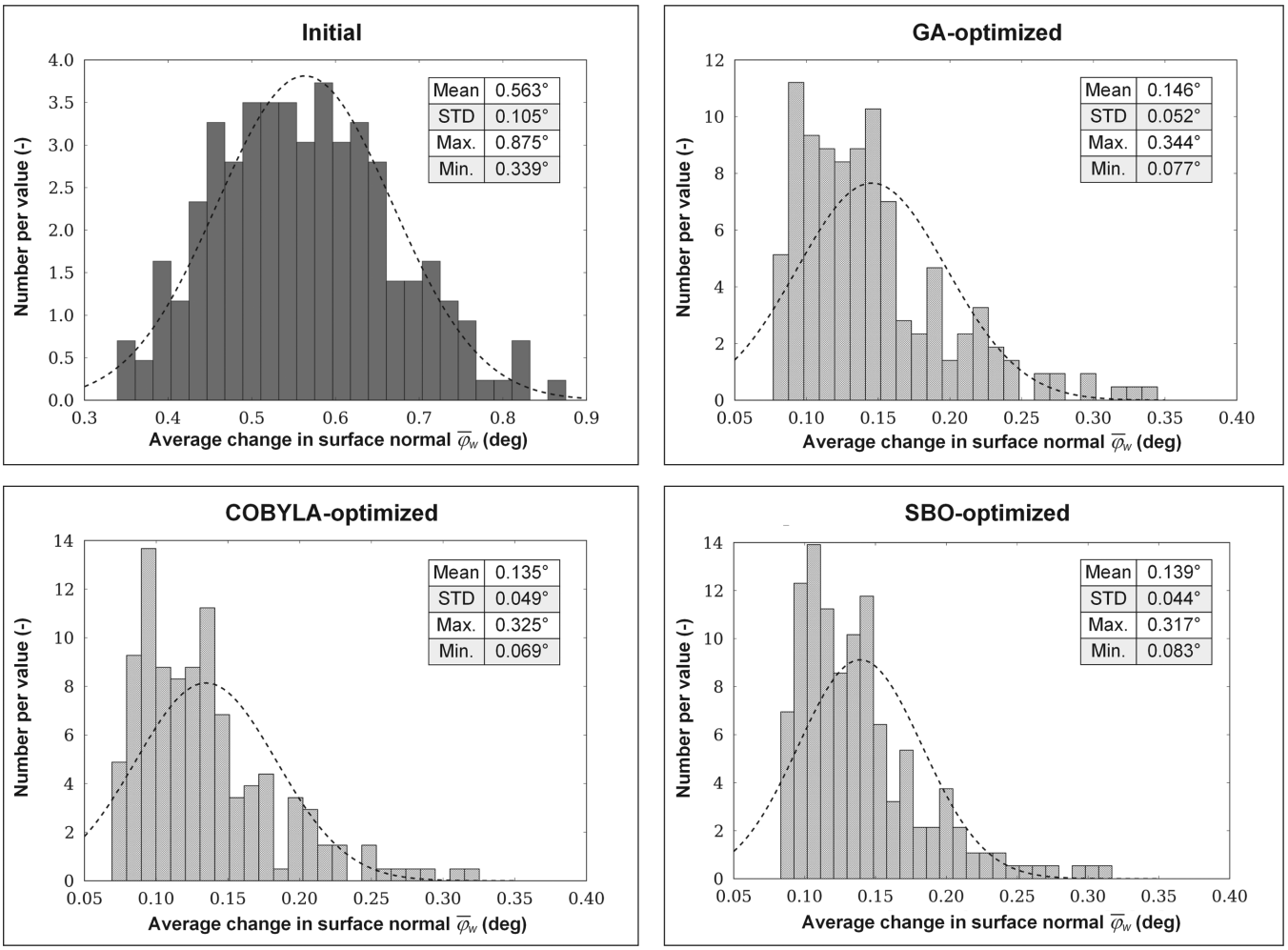

FIGURE 17. Comparison of the process robustness between the initial and the optimized designs for the warpage demonstrator.

plotted curves reveal that the best solution using the COBYLA is achieved after approximately 50 function evaluations, whereas the GA exhibits improvements over the entire optimization history. For the SBO, the primary optimization with the Gaussian process begins after a LHS with a size of 190 observations and terminates after 99 additional function evaluations.

Table VI presents the number of function calls required to reach termination and the best objective function values. The GA produces the best objective function values. The trend for the optimization history of the GA (Fig. 20) suggests an even better solution for larger populations than the applied 8 individuals. The GA almost halves the initial value of 2.0 for the objective function.

Figure 21 illustrates the resulting warpage for the best individual for the GA and the initial geometry. The figure illustrates a large reduction in the warpage at the upper rim, which fulfills the objective of the optimization. In particular, the incidence of the housing wall at the upper left side has been nearly eliminated. Table VII illustrates that the maximum warpage at the rim is reduced from 0.492 to 0.224 mm. These results correspond well to the self-defined out-of-plane deformation with a reduction in the maximum change of the surface normal from 0.552° to 0.228°.

TABLE V
Process Conditions for Housing

Process Variable	Value
Injection speed (cm^3/s)	128.3
Postfill time (cooling and packing) (s)	25
Packing time (s)	15
Packing pressure (MPa)	60
Melting temperature (°C)	240
Surface temperature of cavity (°C)	27

Figure 22 presents the changes in the wall thickness resulting from the optimization procedure, where the wall thickness of the initial design is compared with the design of the best individual. Thus, the algorithm forces the wall thickness to increase in area section 1 at the bottom of the housing and decrease at the upper side walls. The wall thickness increases by approximately +0.64 mm at the bottom area, where the gate is located and decreases at the upper side walls by between –0.62 and –0.54 mm. In area section 2, there is a small reduction in the wall thickness at the bottom area, which increases toward the upper edge of the part to a final value of −0.62 mm.

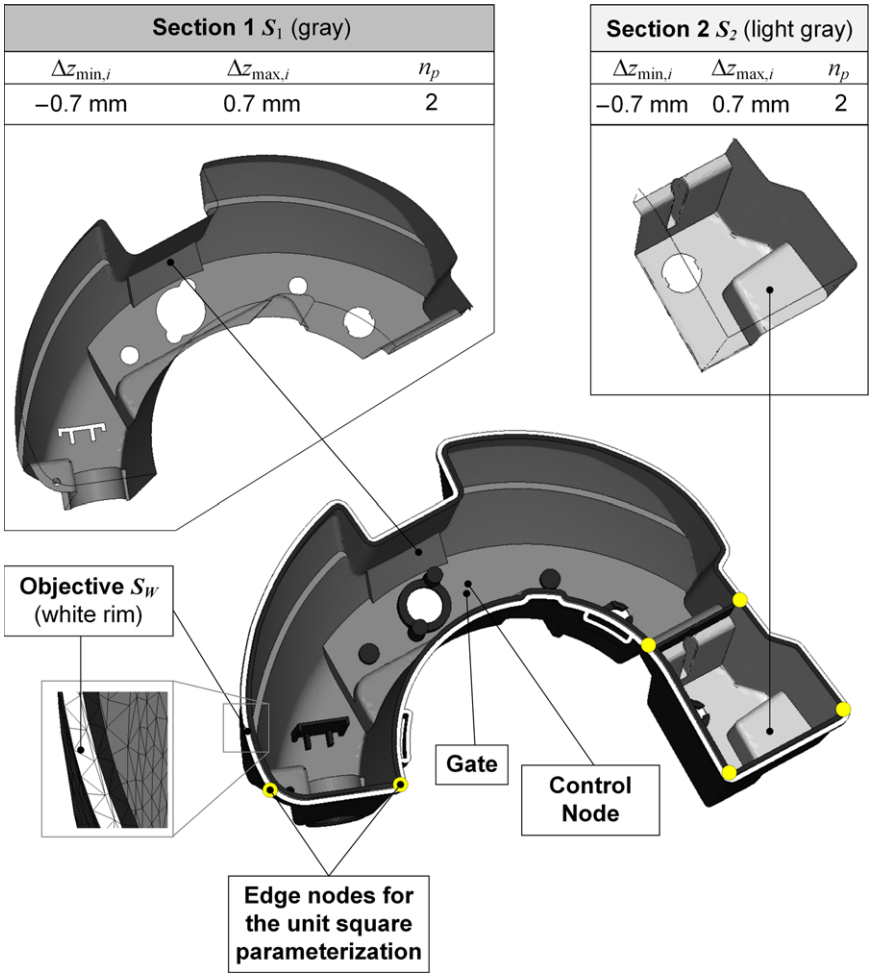

FIGURE 18. Initial cavity showing the position of the gate and control node, selected area sections for variations in thickness and boundary conditions, edge points for parameterization, and area section for warpage optimization.

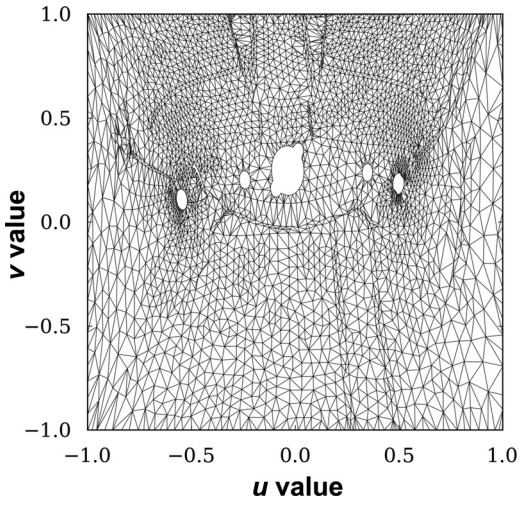

FIGURE 19. Mesh parameterization for section 1 of the housing.

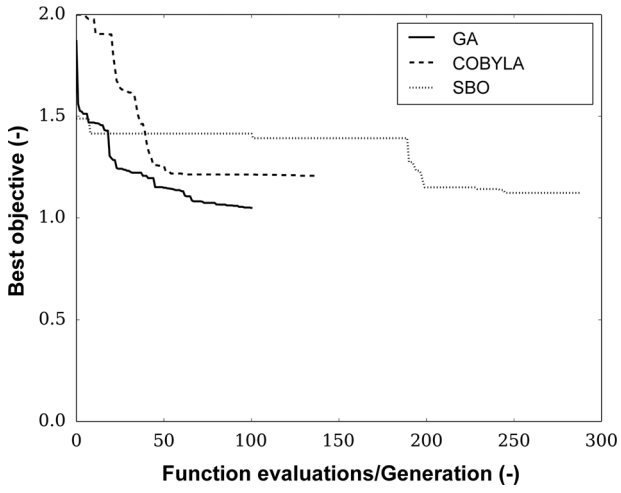

FIGURE 20. Optimization history of the different optimization algorithms for housing.

TABLE VI
Comparison of the Housing Optimization Results

Parameter	GA	COBYLA	SBO
Best objective (-)	1.049	1.207	1.123
Function calls (-)	800	137	289

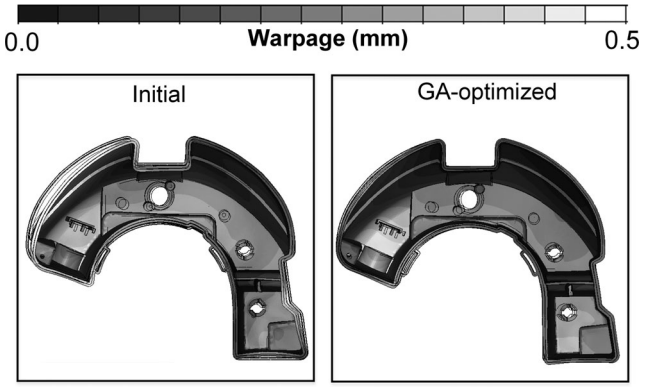

FIGURE 21. Comparison of the warpage between the initial and optimized designs for a geometrical scaling with a scaling factor of 20.

TABLE VII
Discrete Optimization Results for Housing

Parameter	Initial	GA
Maximum warpage at rim (mm)	0.492	0.224
Average change of surface normal (°)	0.194	0.123
Maximum change of surface normal (°)	0.552	0.228
Postfill time (s)	25	35.1
Packing time (s)	15	26.2

The changes in the wall thickness in both of the area sections are physically reasonable and correspond well with the results of the warpage demonstrator. (1) The increase in the wall thickness at the bottom of the housing in area section 1 can be justified as a correction of the design error in the initial geometry, where the wall thickness at the bottom is 0.5 mm smaller than the thickness of the surrounding side walls. Increasing the wall thickness improves the pressure transmission at the side walls. Thus, the shrinkage potential and the existence of sink marks decrease. (2) The decrease in the wall thickness at the upper side walls is accompanied by a decrease in the shrinkage potential at locations far from the gate where there is poor pressure transmission and a smaller solidification layer is present. The same explanation holds for the decrease in the thickness of area section 2 toward the upper edge.

Although no special restrictions are defined, the algorithms tend to a final wall thickness distribution without undercuts, which ensures smooth ejection out of the mold. However, this result does not hold in general and depends on the geometry of the part and its gate location. The avoidance of undercuts could be integrated into the present optimization algorithm and is a subject for future work.

Owing to the increase in wall thickness in the bottom area where the gate is located, the optimized geometry leads to an increase in holding pressure and post fill time (Table VII). However, the objective of the optimization is to achieve an optimal warpage.

Once the geometry for the cavity is optimized, the next step is to apply the modifications from the resulting Stereolithography mesh to the original solid geometry. For this purpose, several software tools can be used to generate a solid model consisting of several nonuniform rational basis splines (NURBSs). Currently, five-axis computerized numerical control (CNC) milling

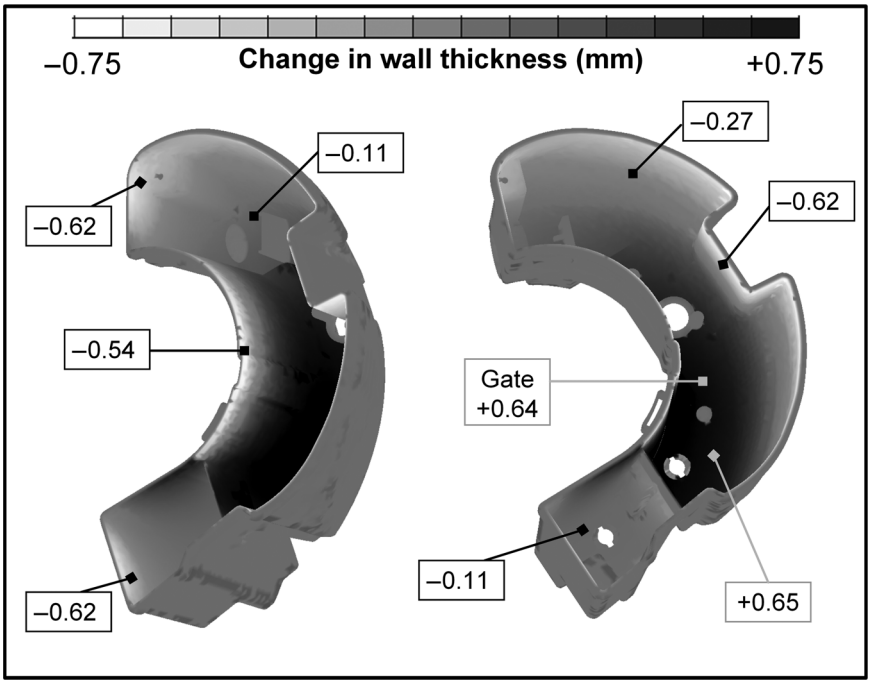

FIGURE 22. Change in the wall thickness for the housing during optimization.

machines are the state of the art and guarantee that more complex shapes can be realized.

Conclusions

In this study, an optimization procedure is developed to minimize the warpage of molded parts by optimizing their wall thickness distributions. This procedure is applied to a warpage demonstrator and a real industrial part, demonstrating the practical applicability of the procedure because of a low user input, computational times on a state-of-the-art personal computer (Intel i7, 3.5 GHz, 16 GB of RAM) below one day, warpage reductions of up to 80%, and a highly plausible final wall thickness distribution.

A comparison of three different derivative-free optimization algorithms that can incorporate constraints demonstrates that for a design vector with up to nine design variables, the direct search method COBYLA and the SBO exhibit the highest efficiency in terms of function calls and the best objective function value. The metaheuristic method represented by a GA is the best choice if the design vector increases to a dimension of 18.

The optimization procedure presented in this paper has been successfully applied to other industrial parts with similar topologies. Nevertheless, there could be cases where this method is not effective, especially for the optimization of complex high-aspect-ratio geometries. The optimization of such geometries requires a high number of finite elements for accurate discretization, which may be difficult in a limited time frame when using state-of-the-art personal computers.

The presented methodology will be developed further. Major research topics will include the prevention of possible undercuts, the automated selection of the unit square edge points on the parameterization plane, and the consideration a nonuniform wall temperature and environmental issues, such as minimizing material usage.

Acknowledgments

The authors wish to thank Simcon Corporation and the open-source projects Python, Scipy Pyevolve, Matplotlib, and DAKOTA for making their software available for this study.

References

1. Kennedy, P. K. Practical and Scientific Aspects of Injection Moulding Simulation; Technical University Eindhoven, The Netherlands, 2008.
2. Michaeli, W.; Niggelmeier, P. Kunststoffe 1999, 89, 5.
3. Michaeli, W.; Baranowski, T. J Polym Eng 2010, 30, 14.
4. Möbius, T.; Eilbracht, S.; Rudolph, N.; Osswald, T. A. Z Kunststofftech 2013, 9, 7.
5. Yin, F.; Mao, H.; Hua, L.; Guo, W.; Shu, M. Mater Des 2011, 32, 6.
6. Shi, H. Z.; Gao, Y. H.; Wang, X. C. Int J Adv Manuf Technol 2010, 48, 7.
7. Deng, Y. M.; Zhang, Y.; Lam, Y. C. Mater Des 2010, 31, 5.
8. Farshi, B.; Gheshmi, S.; Miandoabchi, E. Mater Des 2011, 32, 9.
9. Kurtaran, H.; Ozcelik, B.; Erzurumlu, T. J Mater Process Technol 2005, 169, 5.
10. Ozcelik, B.; Sonat, I. Mater Des 2009, 30, 8.
11. Huang, M. C.; Tai, C. C. J Mater Process Technol 2001, 110, 5.
12. Kong, W. Y.; Kim, J. K. IEEE Trans Compon Packag Manufact 2003, 26, 7.
13. Ozcelik, B.; Erzurumlu, T. Mater Des 2006, 27, 8.
14. Wang, G. L.; Zhao, G. Q.; Li, H. P.; Guan, Y. J. China Mech Eng 2009, 20, 4.
15. Gao, Y. H.; Wang, X. C. Int J Adv Manuf Technol 2008, 37, 7.
16. Zhai, M.; Xie, Y. Int J Adv Manuf Technol 2010, 49, 6.
17. Pandelidis, I.; Zhou, Q. Polym Eng Sci 2004, 30, 9.
18. Zhai, M.; Shen, C. Y. J Reinf Plast Compos 2005, 24, 7.
19. Zhai, M.; Lam, Y. C.; Au, C. K.; Liu, D. S. Polym Plast Technol Eng 2005, 44, 13.
20. Lam, Y. C.; Jin, S. J Injection Molding Technol 2001, 5, 12.
21. Lam, Y. C.; Britton, G. A.; Liu, D. S. Int J Adv Manuf Technol 2004, 24, 6.
22. Lee, B. H.; Kim, B. H. Polym Plast Technol Eng 1995, 34, 18.
23. Lee, B. H.; Kim, B. H. Polym Plast Technol Eng 1997, 36, 16.
24. Osswald, T. A.; Turng, L. S.; Gramann, P. Injection Molding Handbook; Carl Hanser Verlag: München, Germany, 2007.
25. Farin, G.; Hoschek, J.; Kim, M. S. Handbook of Computer Aided Geometric Design; Elsevier Science: Amsterdam, Nederlands, 2001.
26. Floater, M. S. Comput Aided Geom Design 1997, 14, 19.
27. Yoshizawa, S.; Belayev, A.; Seidel, H. P. Proc Shape Modeling Applications 2004, 8.
28. Filz, P. F.; Kutschera, H.; Stöckmann, H.; Welter, F.; Webelhaus, K. Eur. Patent 1,385,103, 2006.
29. Hele-Shaw, H. S. Proc Royal Inst 1899, 49.
30. Besel, P. J.; McKay, N. D. IEEE Trans Pattern Anal Mach Intell 1992, 14, 17.
31. Hemker, T. Derivative Free Surrogate Optimization for Mixed-Integer Nonlinear Black Box Problems in Engineering; Technical University Darmstadt, Darmstadt, Germany, 2008.
32. Holland, J. Adaption in Natural an Artificial Systems; The University of Michigan Press: Ann Arbor, MI, 1975.
33. Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning; Addison-Wesley: Reading, MA, 1989.
34. Filho, J. L. R.; Treleaven, P. C. Computer 1994, 27, 15.
35. Powell, M. J. D. Advances in Optimization and Numerical Analysis; Kluwer: Dordrecht, The Netherlands, 1994, p. 16.
36. Queipo, V. N.; Haftka, R. T.; Shyy, W.; Goel, T.; Vaidyanathan, R.; Tucker, P. K. Prog Aerospace Sci 2005, 41, 28.
37. Matheron, G. Econ Geol 1963, 58, 20.
38. Sacks, J.; Schiller, S. B.; Welch, W. J. Technometrics 1989, 31, 6.
39. Mockus, J.; Tiesis, V.; Zilinskas, A. Towards Global Optimisation; Amsterdam, The Netherlands, 1978; Vol. 2.
40. Jones, D. R.; Schonlau, M.; Welch, W. J. J Global Optim 1998, 13, 37.
41. Nissen, V. Einführung in Evolutionäre Algorithmen; Vieweg und Teubner Verlag; Germany, 1997.
42. Chang, Y.; Liu, C. S.; Huang, S. T.; Huang, C.; Chen, M.; Yang, W. SPE-ANTEC Tech Papers 2009, 55, 3096.