PROCESSES

Automated design of multi-stage forging sequences for die forging

Yorck Hedicke-Claus[1] · Mareile Kriwall[1] · Malte Stonis[1] · Bernd-Arno Behrens[1,2]

Abstract

Forgings are produced in several process steps, the so-called forging sequence. The design of efficient forging sequences is a very complex and iterative development process. In order to automate this process and to reduce the development time, a method is presented here, which automatically generates multi-stage forging sequences for different forging geometries on the basis of the component geometry (STL file). The method was developed for closed die forging. The individual modules of this forging sequence design method (FSD method) as well as the functioning of the algorithm for the generation of the intermediate forms are presented. The method is applied to different forgings with different geometrical characteristics. The generated forging sequences are checked with FE simulations for the quality criteria form filling and freedom from folds. The simulation results show that the developed FSD method provides good approximate solutions for an initial design of forging sequences for closed die forging in a short time.

Keywords Forging · Forging sequences · CAD · Automated process design · Closed die forging

1 Introduction

Forgings are used in highly stressed areas such as engines or in vehicle construction due to their excellent mechanical properties (optimal fibre direction) [1]. The forging is produced in several forming steps (so-called forging sequence). Due to the advancing trend of e-mobility, new types of forgings are emerging for which many companies still have no or few reference processes. The effort for the design of an efficient forging sequence is very high. FE simulations are often used for the design of forging sequences, as they are a very good tool for determining and analysing quality criteria such as form filling, folds, forming forces and temperature curves. A disadvantage of FE simulations is the opposite direction of the simulation to the design direction of forging sequences [2]. The design direction of a forging sequence is backwards from the forging. This makes the design of efficient forging

sequences a complex and iterative development process that can take several days or weeks, depending on the complexity of the forging. The quality of a forging sequence and the corresponding intermediate forms is characterised by a good mass distribution in relation to the finished forging. Therefore, the following requirements arise for an automated and backward design of forging sequences for closed die forging based on the forging geometry:

– The forging sequence must be designed in a short time in a process-safe manner (< 60 min)
– 3D multi-stage forging sequence for different forging geometries
– No need for existing reference processes and empirical knowledge

2 State of the art

In addition to FE simulations, the mass distribution diagram according to MATHIEU [3] is a well-known tool for the design of forging sequences [4], which is applied to long forms according to the shape order of SPIES [5]. Most forged parts belong to this shape class (see Fig. 1) [5] and have a high potential for flash reduction according to Herbertz [6]. By means of the mass distribution diagram, the individual

✉ Yorck Hedicke-Claus
hedicke-claus@iph-hannover.de

[1] IPH–Institut für Integrierte Produktion Hannover gemeinnützige GmbH, Hollerithallee 6, 30419 Hannover, Germany
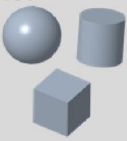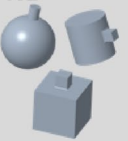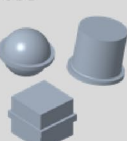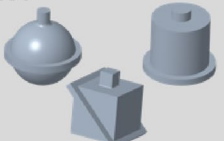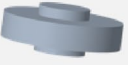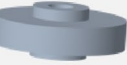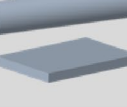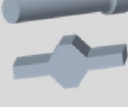
[2] Institute of Forming Technology and Machines (IFUM), Leibniz Universität Hannover, An der Universität 2, 30823 Garbsen, Germany

preforms can be generated by the global material impact along the longitudinal axis and optimized for different components [7]. To determine the forging sequences for forgings backwards from the forging, inverse material flow simulations were used by Wienströr et al. However, the approach was only developed for rotationally symmetric parts in 2D space without an extension of the approach to non-rotationally symmetric parts in 3D space [8].

Various stochastic and analytical methods are used for preform optimization and process design in die forging. The development of approaches to automated design of forging sequences has been the subject of research for a long time. Many of the approaches are limited to explicit components and geometries such as the computer-aided method for intermediate form design for axisymmetric forgings by Biswas et al. [9] or the approach by Subramanian et al. which is limited to specific cross-sectional shapes of forgings [10]. Furthermore, Biswas et al. have developed an approach for the design of preforms for long forms, but no multi-stage forging sequences are generated and the approach is valid for open die forging and not for closed die forging [11]. Mirsadeidi et al. [12] and Alimirzaloo et al. use Lagrange

interpolations for the design of preforms [13]. Knust et al. have developed a method for preform optimization for rotationally symmetric cross-wedge rolled preforms based on evolutionary algorithms [14, 15]. Numthong and Butdee developed an experience-based forging process design system for which they combined FE simulations with a case-based reasoning algorithm [16]. There are several Artificial Neural Network (ANN) based approaches to reduce the effort of FE simulations in the context of process design of forming processes. ANN based predictions of forming forces and die filling have been performed by Chan [17] and Tang [18]. Ciancio has investigated combinations of ANN and optimization methods to improve the prediction and optimization of manufacturing processes [19, 20]. Shao presents an approach to predict the influence of preform geometry on the forging process for a turbine blade using genetic algorithms and particle swarm optimization [21, 22]. Naeimi et al. optimized the 3D preform for rotationally symmetric components by using neural networks to predict the material flow for different preforms and compare them to determine the most suitable variant [23]. Guan et al. determined a single suitable 3D preform for multi-stage sequences using

Fig. 1 Shape order according to SPIES [5]

response surface methods [24]. Torabi et al. used an ANN of the type "NSGA II" to improve the preform geometry of a turbine blade [25]. Seu et al. used an ANN to optimize the geometry of an engine piston [26]. Caspari et al. describe in [27] an approach to optimize a preform by an inverse shape finding approach, based on the work of Chenot [28]. Liu et al. use generic algorithms to generate an optimized preform from a forging in combination with a Fourier analysis and FE methods [29]. The procedure follows a conformal mapping approach to generate a 3D preform [30]. Biba et al. investigated methods for the preform of forgings by a Laplace equation based on a comparison of the isothermal surfaces between the geometries blank and finished part [31, 32]. The presented investigations have in common that they are limited to simple, mostly rotationally symmetric, components or explicitly apply to only one component and only one preform is designed. In many of the presented works, FE results are used. An application for the design of a multi-stage forging sequence for different forging part geometries is not known.

3 Development of the method for automated forging sequence design

To fulfil the requirements of an automated forging sequence design, a method was developed that can generate multi-stage forging sequences based on the 3D geometry of the component. In order to be applicable to the widest possible range of forgings, the method has been developed to take into account all geometric features of forgings as they exist in the SPIES shape order (see Fig. 1 main axis curved, bifurcations, long forms, secondary shape elements, compact shape, disc shape). All forgings can be composed of these characteristics, some may occur more frequently, e.g., several secondary shape elements, several curved axes. The aim of the method is to provide an approximate solution for the first draft of a forging sequence. Explicit geometric characteristics such as draft angles and fillet edges are not considered, as these are only necessary in the detailed design of the forging sequence. The structure of the developed forging sequence design method (FSD method) is shown in Fig. 2. The input of the method is the STL file of a forging for a die forging process. Depending on the forging geometry, certain specifics for the design of the forging sequence are to be derived. For this purpose, the shape order according to SPIES is used, which divides the forgings into different shape classes (see Fig. 1 SC3 = long forms; SC2 = disc shapes, SC1 = compact shape) [5]. In the first step of the method, the module for classifying the forging is used. A classification tool was developed that uses ANNs to classify the forgings into the shape classes according to SPIES and identifies information relevant for the design, such as the cutting

axis, the geometric dimensions, and characteristics such as bifurcations and bending [33]. The main functionality of the classification tool is presented in detail and discussed in [33]. After a pre-recognition of the main shape class (SC1, SC2, SC3) by analysing the spatial dimensions in the three spatial directions, the forging is passed on to ANNs specialised in the main shape classes and the classification into the specific sub-shape class according to SPIES takes place. As a result, the shape class with the highest probability after classification is output, as well as the automatically determined cutting axis.

The output values are then subsequently passed to further modules of the FSD method that require this information (see Fig. 2). In the next step, a module is used to determine a complexity score for the forging. The complexity of a forging influences both the flash ratio [15] and the number of forming steps in the design of forging sequences. Therefore, a complexity score model was developed that determines a complexity score K_{Ges} of the forging based on the STL file of the forging [34]. The main functions of this module for determining the complexity score are presented and discussed in detail in [34]. The complexity of the geometry is evaluated based on the course of the centre of gravity line, the changes in the cross-section and the calculation of an envelope volume. The calculated complexity score is passed on to further modules to calculate the number of stages with fuzzy logic and to the point-shift-algorithm for generating the intermediate forms.
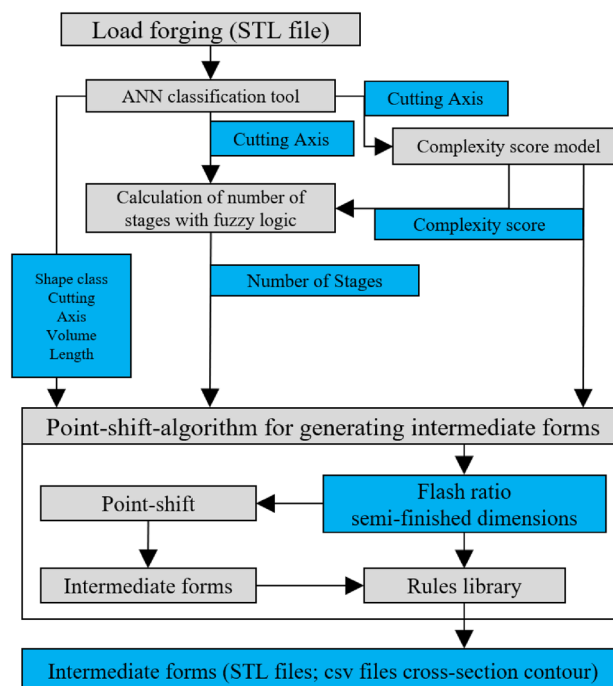

Fig. 2 Flowchart FSD method (grey = modules of the method, blue = output and input parameters)

The next step of the FSD method is to run a module to calculate the number of stages in the forging sequence. The number of forming stages depends on various design parameters, such as the batch size, the available installation space for the tools or the available forming machines and the necessary process forces. A direct calculation of the number of stages is not possible, as the design process for forging sequences is experience-based and company-specific. However, correlations and dependencies of relevant design parameters can be established, which enables the determination of the number of stages by means of fuzzy logic. Fuzzy logic allows fuzzy correlations to be mapped via a set of rules and a discrete solution value to be output through defuzzification.

The fuzzy system is composed of the input parameters "forming force", "complexity", "max. forming per stage", and "batch size" and provides a suggestion for the number of stages as an output value (Fig. 3). In the set of rules of the fuzzy system, rules such as "Small batch size & low complexity & low forming force = low number of stages" are mapped out.

The batch size and maximum forming per stage are entered manually by the user. The input parameter for the complexity of the forging is transferred from the complexity score model (K_{Ges} cf. [34]). To be able to conduct the approximate determination of the forming force F based on the forging geometry; the projection surface of the forging is determined from the STL file. Using the projection surface A_d and the resistance to deformation at the end of forming k_{we}, the required forming force F can be calculated [4, 35]:

$$F = A_d * k_{we}. \tag{1}$$

The projection surface corresponds to the mean component cross-section perpendicular to the forming direction. The projection surface can be determined using the slicer-algorithm (Sect. 3.1). The resistance to deformation at the
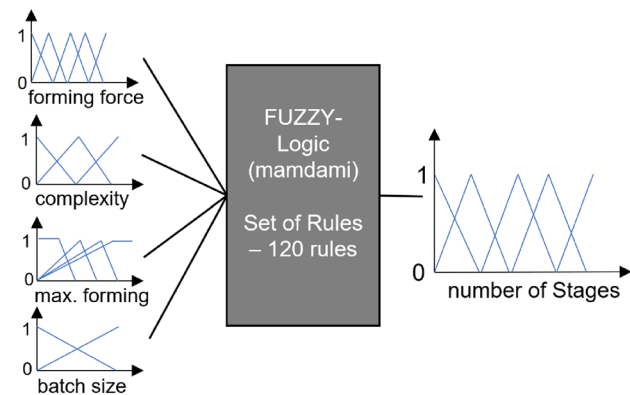
end of forming k_{we} is calculated via the yield stress k_f and a shape factor that takes the workpiece shape into account [35]. The yield stress depends on the material and the corresponding material data are stored in the method and the data sets can be extended accordingly. The number of steps calculated by the fuzzy logic system is passed on to the point-shift-algorithm to generate the intermediate forms. With the determined parameters (cutting axis, shape class, complexity score, and number of stages), a developed algorithm first calculates the necessary flash ratio and the mass distribution around the center of gravity line of the forging [36]. Step by step, the mass distribution of the forging and the semi-finished product is then adjusted, and the corresponding intermediate forms are exported as an STL file. In the following, the division of the forging into cutting planes as well as the functioning of the algorithm for the generation of the intermediate forms through the cross-section adjustments and the volume compensation through a length change via the forging sequence are presented in detail.

3.1 Slicing the forging geometry

The basis for the method is the division of the forging (in STL format) into cross-sections (slices) by a slicer-algorithm. The number of cutting planes n is specified by the user of the method. The forging is divided in n cross-sections at a constant distance d along the cutting axis (can be any of the three spatial directions) of the forging. Figure 4 shows the slicing of a 2-cylinder crankshaft into cross-sections. Geometric information such as surface area, center of
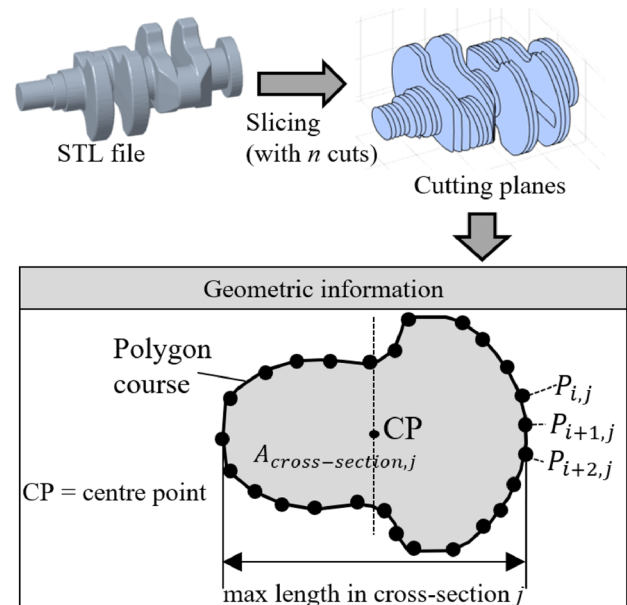

Fig. 4 Slicing a 2-cylinder crankshaft to obtain cross-sectional information

Fig. 3 Fuzzy-Logic for calculating the number of stages of the forging sequence

gravity and polygon chain of the cross-sectional area can be determined from the sectional planes.

Every point of the polygon chain is described through its position vector $\vec{x}$ and satisfies the following equation in which $\vec{p}$ is a support vector and $\vec{n}$ a normal vector to the plane:

$$(\vec{x} - \vec{p}) * \vec{n} = 0. \tag{2}$$

After slicing the component, a graph model of the component is created to save the neighborhood relations of the points. This is necessary for the later creation of the intermediate forms and checking the intermediate forms for forming rules. In the first step, all faces of the STL file are read in and the neighborhood relationships of the faces are determined. Then the cross-section points $P_{i,j}$ for n cuts are read in. For each cross-section point $P_{i,j}$ a face is searched in which the point is located. A tracing map is created for this purpose. For performance reasons, the search is aborted after too many levels. The search is conducted using the breadth-first search (BFS). In contrast to the depth search, all nodes that can be reached directly from the start node are searched first. A data structure in the form of a graph object is available, which makes it possible to export the intermediate forms as STL objects later in the method. Slicing provides relevant geometric information such as the area of the cross-sections and centres of gravity.

3.2 Point-shift-algorithm for generating the intermediate forms

To generate the intermediate forms of the forging sequence backwards from the forging to the semi-finished product, an algorithm was developed that shifts the mass distribution and center of gravity line of the forging step by step in the direction of the semi-finished product geometry [36]. For this purpose, forming factors f_i are defined, which indicate what percentage of the total forming takes place within a stage of the forging sequence. The following must apply to the entire forging sequence: $\sum_{i=1}^{n} f_i = 1$. The cross-section adjustment is done by adjusting the polygon courses of the cross-section contours. The cross-sectional contours of the forging are determined in all cutting planes A_j using the slicer algorithm presented in Sect. 3.1. The contour is described by a polygon consisting of the corner points P_1 to P_{m-1}. It is $[P_1, P_2] \cup [P_2, P_3] \cup \ldots \cup [P_{m-1}, P_m]$ and thus there is a closed polygon (see Fig. 4). For each point P_i the target coordinates on the target contour (semi-finished cross-section) are determined for displacement in the direction of the semi-finished product. The semi-finished product can be round or rectangular. Starting from the center of gravity of the forging and the semi-finished product, beams are generated through the polygon points
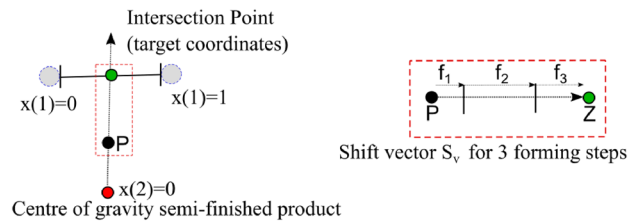
Fig. 5 Target coordinates for determining the shift vector
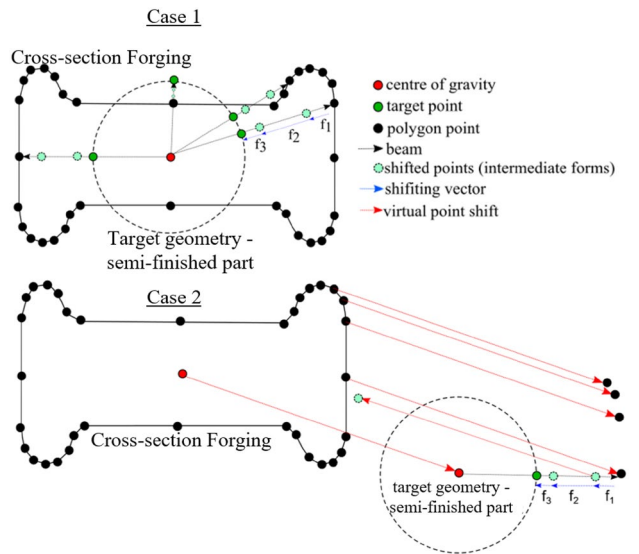
Fig. 6 Point-shift-algorithm for the cases centre of gravity for forged part = centre of gravity semi-finished product (Case 1) and forged part ≠ centre of gravity semi-finished product (Case 2)

P_i in the cross-section. The geometry of the semi-finished product is described by means of individual lines. For this purpose, a defined number of points is evenly distributed on the semi-finished product geometry and a line segment is defined between two points. To calculate the target coordinates for the point P_i, the intersection point between the beam and line segment on the target contour is calculated by solving a linear system of equations. The system of equations has a solution on the straight line between two points on the target geometry (semi-finished product) and on the beam starting from the center of gravity to the point P_i (see Fig. 5):

$$x(1) \leq 1 \cup x(1) \geq 0 \cup x(2) \geq 0 \tag{3}$$

A shift vector S_v is defined between the target coordinates determined in this way and the polygon point P_i.

Along this shift vector (S_v) the point P_i is shifted stepwise according to the number of forming steps and the forming factors f_i (see Fig. 6). If the centres of gravity of

the semi-finished product and the forging are not identical in the cross-section, first all points of the polygon chain of the cross-section of the forging are virtually shifted by the offset of the centres of gravity of the semi-finished product and the forging (see Fig. 6 Case 2). Then the cross-section contour is adjusted according to the displacement vectors. Finally, the points corresponding to the virtual displacement at the beginning are displaced back.

The forming factors f_i can be entered individually for cross-section classes depending on the cross-section size (e.g., cross-sections larger than semi-finished product diameter or smaller than semi-finished product diameter). The number of classes x can be entered manually and the cross-sections are evenly sorted into the selected classes. This allows the mass distribution in the intermediate forms to be set more finely and individually. The point clouds of the intermediate forms (*IMF*) over all iterations (s) of the forging sequence, result from the sum of the multiplication of all shift vectors S_v for all points P_i in all cross-sections A_j with the forming factor matrix:

$$IMF,\ iter(1,\ldots,s) = \sum_{j=1}^{n} S_v(A_j) * \begin{pmatrix} f_{1,1} & f_{\ldots} & f_{1,s} \\ f_{2,1} & f_{\ldots} & f_{2,s} \\ f_{x,1} & f_{\ldots} & f_{x,s} \end{pmatrix}. \quad (4)$$

The number of shift vectors in a cross-section A_j results from the amount of points P_i in the cross-section.

To take different geometry-specific features into account when designing the forging sequence with the point-shift-algorithm, special cases of the algorithm were defined, which are selected at the beginning based on the ANN-based geometric analysis and classification (cf. [33]). Figure 7 shows the different special cases for the point-shift-algorithm. In a first query, it is checked whether there are gaps in the component. If there are gaps, the gaps or bifurcations are closed by a thin material layer between the two cross-sections in a cutting plane so that there is only one continuous cross-section per cutting plane (CloseGap- algorithm in Fig. 7). The second query checks whether the component is a rotationally symmetrical component (e.g., flanged shaft, upright forged). If this is the case, the generation of the intermediate forms is only conducted by observing the cross-section in the center of the component in 2D. Subsequently, the cross-section contour is rotated by 360 degrees to obtain a 3D contour of the intermediate forms. If the part is not rotationally symmetrical, it is queried whether it is a long form or a compact shape or a disc shape. In the case of a bended long form, a variant of the point-shift-algorithm is called up in which a bending operation is integrated. In the bending operation, only the centers of gravity of the cross-sections are shifted and there is no adjustment of the cross-section contour. This bending line is determined by a
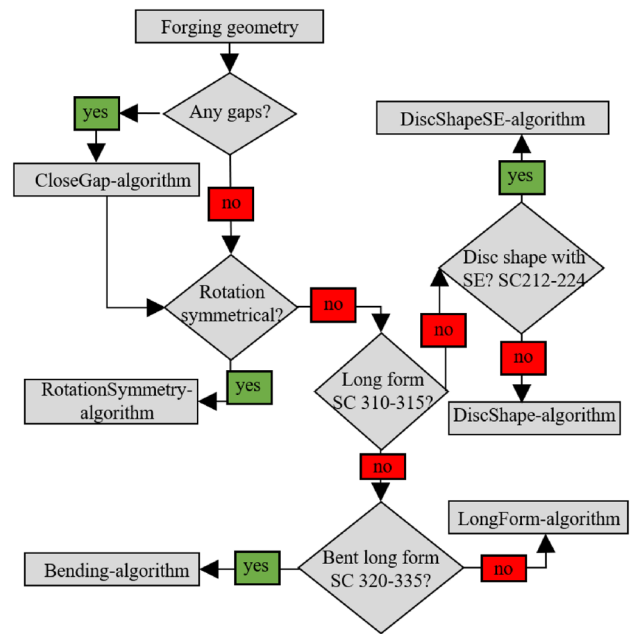

Fig. 7 Queries for the appropriate point-shift-algorithm depending on geometry-specific features and the shape class

linear regression. The differences in calling the point-shift-algorithm for long forms and disc shapes lie in the orientation of the semi-finished product and the length changes over the forging sequence.

3.3 Change in length of the intermediate forms over the forging sequence

By means of the point-shift-algorithm and the resulting change in cross-section (change in surface area), the material flow is mapped transversely to the forming direction. In order to equalise a possible volume difference due to the cross-section changes, a length change of the intermediate forms must be carried out and thus material flow in the longitudinal direction to the forming direction must be considered. The boundary condition for a length change is the volume constancy valid over the entire forging sequence for the closed die forging process, so that the following applies for each volume $V_{iter,i}$ of the intermediate forms:

$$V_{iter,i} = V_{semifinished} = V_{forging} + V_{flash} \quad (5)$$

Knust et al. have developed a method for the calculation of an adaptive flash ratio for the preform optimisation of rotationally symmetric and cross-wedge rolled preforms [15]. This approach was extended to calculate the flash ratio for an entire forging sequence for closed die forging. An

additional factor m was integrated into the calculation and the complexity value calculated by the complexity score model [34] was used used to calculate the flash ratio V_{flash,A_j}:

$$V_{flash,A_j} = m + 0.0059 * e^{5.6249*K_{Ges}}. \tag{6}$$

The factor m is calculated as a function of the shape class [5] as follows:

$$m = \begin{cases} 0.05 * \left(1 + 0.3 * e^{K_{Ges}}\right) & if \ SC = 1 \cap 2 \\ 0.1 * \left(1 + 0.3 * e^{K_{Ges}}\right) & if \ SC = 3 \end{cases}. \tag{7}$$

An exception to the volume constancy in the method is a calibration stroke before the final form, in which only very little forming takes place $(f_1 \leq 10\%)$. Here it is assumed that a large part of the material has already flowed into the flash in the previous stages. Therefore, the volume of the forging without flash is used as the reference volume for the volume adjustment in the calibration stroke. The reference volume over the iterations of the forging sequence is given by:

$$V_{Iter,i} = \begin{cases} V_{forging} & if \ f_i \leq 0.1 \cup i = 1 \\ V_{semifinished} & if \ f_i > 0.1 \cup i > 1 \end{cases} \tag{8}$$

In the presented method, only die forging processes are considered and only intermediate forms are generated for die forging. Therefore, the following conditions for long forms (SC 3 according to SPIES) apply to the length changes over the forging sequence:

$$l_{forging} \geq l_{Iter,1} \geq \dots \geq l_i \geq l_{semifinished}. \\ \Delta l < f_l \tag{9}$$

f_l is defined as the elongation factor in a stage. Values between 0 and 10% can be specified here. The limit value of 10% results from expert interviews with project partners, since a length change of more than 10% in a forming step during die forging occurs only very rarely in practice. However, a change of the limit value is feasible with little effort. To perform the length change over the forging sequence, the distance between the cutting planes generated by the slicer-algorithm is changed. The initial distance d is calculated as follows:

$$d = \frac{l_{forging}}{n}. \tag{10}$$

By comparing the volume after the execution of the point-shift-algorithm with the volume of the previous iteration, the volume difference that has to be compensated by a length change can be determined. Using the cross-sectional areas $A_{j,iter}$ and the distances $d_{i,iter}$ between the cross-sectional areas, the volume segments and the volume difference can be calculated:

$$V = \left|V_i - V_{i+1}\right| = \left|\sum_{j=1}^{n-1} A_{j,iter,i} * d_{j,i} - \sum_{j=1}^{n-1} A_{j,iter,i+1} * d_{j,i+1}\right|. \tag{11}$$

A variable length change of the distance $d_{i,iter}$ in different areas of the component leads to an improved mass distribution and a better form filling behaviour than a uniform adjustment over all cross-sections [37]. The individual volume segments are divided into six (number can be varied) different classes (V++, V+, V+=, V−=, V−, V−) according to their volume (Fig. 8). Segments of class V++ have the largest volume and segments V− the smallest volume. A volume segment V_i is formed by a cut i, the cut $i + 1$ and the distance d_i. For each volume segment, a volume change factor V_L is calculated:

$$V_{L,i} = \frac{V_i}{d_i} \tag{12}$$

In order to carry out small length changes with a simultaneous significant change in volume, large cross-sections (V++ and V+) in the forging are suitable. Large length changes with small volume adjustment are carried out with small cross-sections (V− and V−).

In addition to the ΔV to be compensated, the maximum permissible length change f_l in the forming stage is another boundary condition for the length change. In order to bring ΔV to zero and at the same time to maintain and achieve the maximum length change in a stage, first a change in the distances in V− (low $V_{L,i}$) is made to achieve the length change, then the volume is balanced via V++ (high $V_{L,i}$). If the volume constancy is not achieved or the changed length of the intermediate form does not correspond to the boundary conditions ($l_{iter} \leq l_{iter-1}$ and $\Delta l \leq f_l$), the adjustment is repeated with the next volume segments (V+ and V−, then V+= and V−=). For forgings from the disc shape class, the change in length is defined differently, since in this case the forging has the shortest
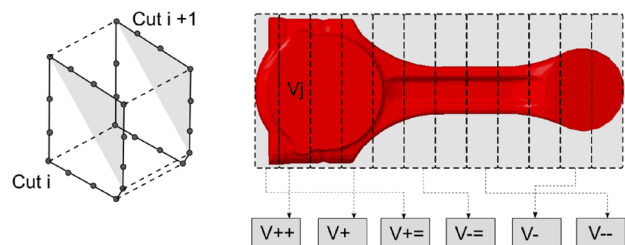

Fig. 8 Division of a connecting rod into volume segments and classification into the six volume classes

Table 1 Rule Categories

Rule category	Explanation	Example
Rules for points	Review of individual points	Fold criterion [38] Undercuts
Rules for areas	Checking component cross-sections	Maximum cross-section change
Rules for volume	Review of the overall shape	Volume constancy Compression ratio

Fig. 9 Checking a cross-section of an intermediate form for undercuts (areas marked in red show an undercut)

length (or the lowest height) and the semi-finished product is significantly longer. Here, a maximum compression ratio is taken into account when balancing the volume to achieve volume constancy.

3.4 Rules library

In order to ensure the process reliability of the generated forging sequences, a rule library was developed and implemented in the FSD method. This allows design rules and forming boundary conditions to be implemented in formulas in the method. Three rule classes were defined: Point rules, area rules and volume rules (see Table 1). A grammar for the creation of rules was developed to enable the users to integrate their own rules into the FSD method.

An executable tree generator is used for parsing (checking whether the rules can be mapped with the grammar). After an intermediate form has been generated by the point-shift-algorithm, the intermediate form is checked for the rules activated in the rule library. It is possible to activate and deactivate the rules. If a rule is not respected, an adjustment is made, e.g., by shifting the points to a target point by which the rule is respected. For this purpose, tolerance ranges are defined for the point shift.

In this way, process-related boundary conditions can be checked and adhered to. This prevents, for example, undercuts from occurring in a cross-section contour. Figure 9 shows an example of how to check a cross-section for undercuts. Each cross section is divided into four quarters (Q1 and Q2 above the plane of division, Q3 and Q4 below the plane of division) and it is checked whether an undercut is present. The condition for point P_2 in Q3 that there is no undercut is as follows:

$$x(P_1) \leq x(P_2) \leq x(P_3) \leq x(P_4) \tag{13}$$

If this condition is violated, point P_2 is moved to the position of point P_2* in Fig. 9 and the undercut is resolved. This check after each iteration in each cross-section ensures that no faulty intermediate forms can be generated.

In addition to undercuts, potential forming faults such as folds are also checked. For this purpose, the criterion
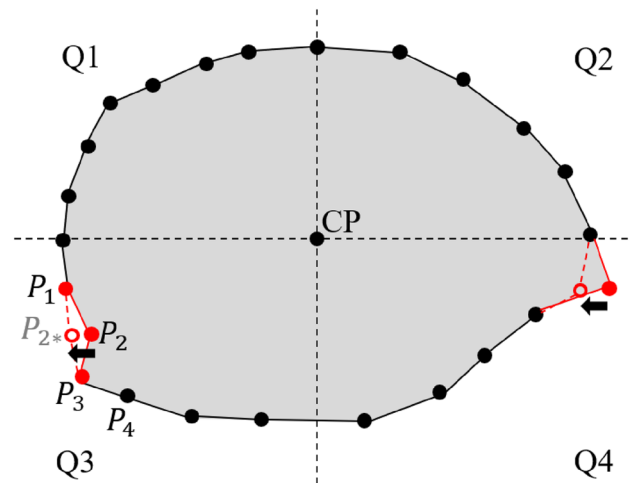
developed by BEHRENS ET AL. for checking folds in mesh structures [38] is adapted to the developed FSD method and transferred to the corresponding grammar. Within the control loop for checking the rules, rule category 1 is checked first. Here, for example, undercuts and potential folds [34] are checked and corrected. In the second step, the surfaces are checked (e.g., compliance with a maximum cross-section change) and the surface is scaled without changing the cross-section contour. In the third step, the component volume is checked with a change in length, if necessary, to adjust the volume. Formula 13 shows an example of the structure of a rule, as entered in the method:

$$iter = 1 | A(iter - 1, cut) < A(iter, cut) * 3 \tag{14}$$

The example rule describes the cross-section change over the iterations (intermediate forms). Here it is checked that a cross-sectional change may not be greater than a factor of three.

4 Results of the FSD method

In order to validate the method for the automated design of multi-stage forging sequences, the developed method is applied to various forgings. Following the shape order according to SPIES, forgings with different geometric characteristics are selected for validation. The selected forgings and the corresponding geometric characteristics are listed in Table 2. The selected geometries contain all geometric characteristics of the SPIES shape order from which the individual shapes are composed. This ensures that they can be applied to different forgings.

Table 2 Selected forgings for the validation of the FSD method

Forging	Characteristics
A	Shape class 313: pronounced complex mass distribution, pronounced secondary shape elements (l=249 mm, V=2.7 × 10^5 mm^3)
B	Shape class 312: characteristic H-profile cross-section in the centre of the forging (l=217 mm, V=2.18 × 10^5 mm^3)
C	Shape class 315: long form with secondary shape element, bifurcation on forging (l=212 mm, V=1.4 × 10^5 mm^3)
D	Shape class 104: disc-shaped element no secondary shaping elements, upright forged, rotationally symmetrical (max diameter=320 mm, height 180 mm, V=9.9 × 10^6 mm^3)
E	Shape class 322: bended, characteristic H-profile cross-section in in the centre of the component (l=465 mm, V=4.65 × 10^6 mm^3)
F	Shape class 312:very thin-walled in the middle, complex mass distribution (l=187 mm, V=1.61 × 10^5 mm^3)

Table 3 Result parameters from automated stage planning (complexity score, forming force and shape class)

Forging	Complexity	Shape class	Forming force
A	0.47	321–wrong (313 corrected manually)	954.6 tonnes
B	0.35	312/314	689.7 tonnes
C	0.41	315	900.2 tonnes
D	0.27	104	2520 tonnes
E	0.46	322/321	642.9 tonnes
F	0.44	312/315	742 tonnes

Table 4 Result parameters from automated stage planning (Number of stages, flash ratio and forming factors)

Forging	Number of stages	Flash ratio (%)	Forming factors
A	3	23.2	[9%, 40%, 51%]
B	3	18.7	[9%, 51%, 40%]
C	3	20.3	[9%, 51%, 40%]
D	3	9.6	[10%, 40%, 50%]
E	4	22.8	[9%, 71%, 20%, Bending]
F	3	21.7	[10%, 60%, 30%]

The FSD method is implemented in MATLAB. In this study, the method is applied to a Windows computer with 16 GB RAM, an SSD and an I7 processor. The selected forgings were loaded and processed in all presented modules of the method (cf. chapter 3). The design parameters determined by the FSD method are listed in Tables 3 and 4. The used material in this study is 1.7225. The detection of the shape class for forging A gave an incorrect result and was corrected manually.

The detection of the shape class for forging A gave an incorrect result and was corrected manually.

All forging sequences contain a calibration stroke, which means $f_1 \leq 10\%$. The semi-finished product diameters are selected manually. Boundary conditions are the maximum permissible change in length that can be achieved over the entire forging sequence, as well as the proposed flash ratio. In the selection, a diameter is aimed for that is as close as possible to the largest cross-section of the forging. This corresponds to the procedure for design in practice. Table 5 shows the generated forging sequences and the information on the selected semi-finished product.
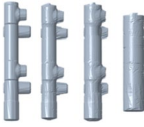
The runtime of the method is dependent on both the number of facets of the STL file and the number of cuts, as these two parameters significantly influence the number of points to be shifted by the point-shift-algorithm. The runtimes for the forging sequences (Table 5) is visualized in Fig. 10.

For all forgings ($n = 60$), the forging sequence was generated in less than one hour. The process is particularly fast for part D, as the points for the rotationally symmetrical forging only need to be moved in one cross-section. For a sufficient quality of the result export, however, points are added during the generation of forging sequences if the initial geometry was not sufficiently finely meshed.

5 Validation of the FSD method with FE simulations

The generated forging sequences, with the respective intermediate forms and semi-finished product, are checked by means of FE simulations in order to verify quality criteria such as form filling and freedom from folds. The material modelling in the simulations is done with the

Table 5 Forging sequences generated by the method for the validation forgings A–F

Forging	Forging sequence	Semifinished product
A		D = 45 mm L = 200 mm V = 3.2 × 10⁵ mm³ Runtime = 57 min
B		D = 45 mm L = 162 mm V = 2.57 × 10⁵ mm³ Runtime = 38 min
C		D = 38 mm L = 146 mm V = 1.65 × 10⁵ mm³ Runtime = 37 min
D		D = 240 mm L = 245 mm V = 1.1 × 10⁷ mm³ Runtime = 0.1 min
E		D = 43 mm L = 391 mm V = 5.7 × 10⁶ mm³ Runtime = 39 min
F		D = 44 mm L = 131 V = 1.99 × 10⁵ mm³ Runtime = 38 min

Fig. 10 Runtime and number of facets of the forgings A–F

Table 6 Regression coefficients of Hensel-Spittel equation of 1.7225

Coefficient	Value
Dependence of temperature	$m_1 = -0.00289$
Dependence of strain hardening	$m_2 = -0.1123$
Dependence of equivalent strain rate	$m_3 = 0.14368$
Dependence of equivalent strain	$m_4 = -0.04879$
Solidity	$A = 1872 N/mm^2$

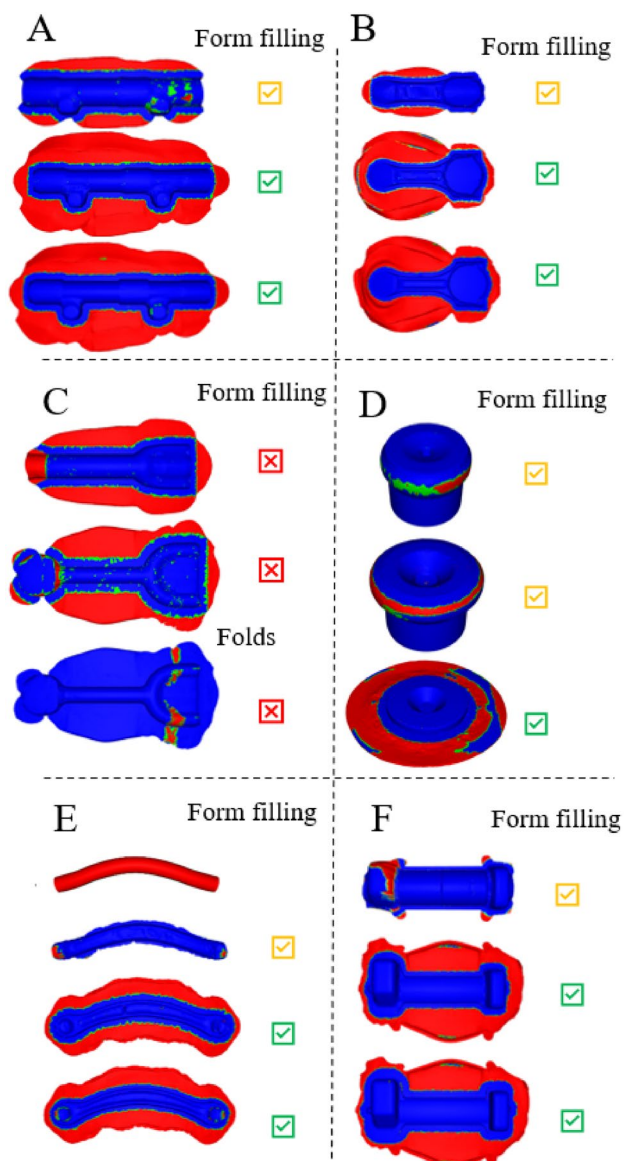**Table 7** Simulation parameter in FORGE NXT 3.2

Simulation parameters	Value
Material	1.7225
Billet temperature	1250 °C
Die temperature	250 °C
Mesh-size	2 mm
Press kinematics	Hydraulic press
Ambient temperature	50 °C
Friction μ	0.3
Forming velocity	200 mm/s

A, m_1, m_2, m_3, m_4. The individual regression coefficients of the Hensel-Spittel equation of 1.7225 are shown in Table 6.

The parameters from Table 7 were used for the simulations for validation.

The evaluation of the FE simulations is summarized in Fig. 11. The quality criteria form filling and freedom from folds are checked. The form filling was analyzed via the contact between the workpiece and the form. The blue areas represent areas with form filling. For the forgings A, B, D, E and F, form filling can be achieved in the final form, but often complete form filling is not achieved in the first stage. The forging F was simulated with a flash barrier in stage 1. This was done to adapt the forging sequence closer to reality, since an initial mass distribution is usually achieved by specific preforming processes such as stretch rolling or cross wedge rolling, which are significantly more material-efficient than a closed-die forging process. Here, other length changes can also be mapped with process reliability. For the forging C, no form filling was achieved in the first two intermediate forms. Folds also occurred in the process, which suggests that the algorithm, which places a thin layer of material in the bifurcation, still needs to be adapted and further rules are necessary to ensure freedom from folds for this class of forgings. In the other forging sequences, no folds occurred.

The results for forging E show that the FSD method can also process bended forgings and realistically represents bending operations in the forging sequence. Besides long forms, other shape classes (stocky part D) can also be processed and formable approximate solutions for forging sequences are

Hensel-Spittel equation implemented in FORGE® NXT 3.2 [39]. With this equation, the forming behavior is determined by means of a function dependent on the temperature T, the effective strain ε, the flow stress σ_f, the strain rate $\dot{\varepsilon}$, and the coefficients A and m_{1-4}. The Hensel-Spittel equation used by FORGE® NXT 3.2 is represented in the following equation:

$$\sigma_f = A e^{m_1 T} \varepsilon^{m_2} e^{m_4/\varepsilon} \dot{\varepsilon}^{m_3}. \tag{15}$$

The flow curve used for hot bulk metal forming processes is sufficiently precise by means of the coefficients

Fig. 11 Simulation results for the validation forgings A–F to check the quality criterium form filling by the contact in the FE simulations

generated. Overall, the simulation results show that the design parameters determined automatically by the FSD method (number of stages, flash ratio, etc.) lead to process-reliable forging sequences for a wide range of forgings. The results are generated in a short time and represent a very good approximation solution for an initial design of the forging sequences.

6 Summary and outlook

A method was presented that enables the automated generation of multi-stage forging sequences for die forging based on the forging geometry (FSD method). The FSD method is composed of different modules (ANN classification tool,

complexity score model, fuzzy design, intermediate form generation algorithm), which were developed to be able to fulfil the requirements of a method for automated forging sequence design. It was shown that the FSD method generates multi-stage forging sequences for different forgings based on the 3D model (STL file) of the geometry in a short time (less than 60 min). The individual intermediate forms of the forging sequence are exported as STL files. Based on the STL files, initial FE simulations can be set up to check quality criteria of the forging sequences (freedom from folds, form filling). The presented simulation results for the validation of the developed method show that the quality criteria are very well met and the suggestions for the forging sequence provide a very good approximate solution for design support. With the current method, there are still limitations in the applicability of the method for strongly bended geometries (SC 331 and higher). Such parts can currently only be processed segment by segment. This means that the forging would have to be divided into segments with a curved main axis for the method and the partial results would then have to be joined together. In future development work, this limitation is to be solved through adjustments, e.g., by calculating the neutral fiber of the forging and slicing along the neutral fiber to determine the cross sections. For complex forgings with bifurcations, adjustments are still necessary to obtain process-reliable forging sequences. In a next development step, preforming processes could be implemented in the FSD method (e.g., stretch rolling and cross wedge rolling) to further increase the realism of the results. This would allow new boundary conditions for the geometry generation, such as significantly greater length changes or process-specific cross-sectional shapes. In addition, the forging sequences could be designed to be even more material-efficient if preforming processes are considered.

References

1. Behrens B-A, Nickel R, Müller S (2009) Flashless precision forging of a two-cylinder crankshaft. Prod Eng 3(4–5):381–389

2. Langner J, Malte S, Bernd-Arno B (2016) Investigation of a moveable flash gap in hot forging. J Mater Proc Technol 231:199–208

3. Mathieu H (1991) Ein Beitrag zur Auslegung von Stadienfolgen beim Gesenkschmieden mit Grat. Fortschritt-Berichte VDI Reihe 2: Fertigungstechnik Nr. 213, VDI-Verlag, Düsseldorf

4. Doege E, Bernd-Arno B (2016) Massivumformung. Handbuch Umformtechnik. Springer Vieweg, Berlin, Heidelberg, pp 455–693

5. Spies K (1957) Eine Formenordnung für Gesenkschmiedestücke. Werkstattstechnik und Ma-schinenbau 47:201–205

6. Herbertz R, Licht W, Fuss F (2011) Materialeffizienz in der Massivumformung. In: Schmieden Journal

7. Movrin D et al (2010) Optimization and design of multistage hot forging processes by numerical simulation and experimental verification. J Technol Plast 35(1–2):75–88

8. Wienströer M (2004) Prozesssimulation der Stadienfolge beim Schmieden mittels Rückwärtssimulation. Diss. Hannover: Universität

9. Biswas S, Knight W (1974) A computer aided design of axisymmetric hot forging dies. 15th International Machine Tool Design and Research Conference

10. Subramanian TL, Akgerman K, Altan T (1977) Computer aided preform design for precision isothermal forging, 5th NAMRC

11. Biswas S, Knight W (1976) A computer aided preform design for long hot forgings. 17th International Machine Tool Design and Research Conference

12. Mirsaeidi M et al. (2009) Optimum forging preform shape design by interpolation of boundary nodes. In: Proceedings of the World Congress Engineering, 2, 1766–1771, WCE 2009, London, UK

13. Alimirzaloo V, Biglari FR, Sadeghi MH, Keshtiban PM, Sehat HR (2019) A novel method for preform die design in forging process of an airfoil blade based on lagrange interpolation and meta-heuristic algorithm. Int J Adv Manuf Technol 102(9–12):4031–4045

14. Knust J et al (2016) Preform optimization for hot forging processes using genetic algorithms. Intell J Adv Manuf Technol 89(5–8):1623–1634

15. Knust J, Stonis M, Behrens B-A (2016) Preform optimization for hot forging processes using an adaptive amount of flash based on the cross section shape complexity. Prod Eng 10(6):587–598

16. Numthong C, Butdee S (2012) The knowledge based system for forging process design based on case-based reasoning and finite element method. Appl Sci Eng Progress 5(2):45–54

17. Chan WL, Fu MW, Lu J (2008) An integrated FEM and ANN methodology for metal-formed product design. Eng Appl Artif Intell 21(8):1170–1181 (**Elsevier Verlag**)

18. Tang YC, Zhou X-H, Chen J (2008) Preform tool shape optimization and redesign based on neural network response surface methodology. Finite Elements Analysis Design 44(8):462–471 (**Elsevier Verlag**)

19. Ciancio C et al (2014) Heuristic techniques to optimize neural network architecture in manufacturing applications. Neural Computing Appl 27(7):2001–2015 (**Springer Verlag**)

20. Ciancio C et al (2015) Design of a high performance predictive tool for forging operation. Procedia CIRP 33:173–178 (**Elsevier Verlag**)

21. Shao Y, Ou H, Guo P, Yang H (2019) Shape optimization of preform tools in forging of aerofoil using a metamodel-assisted multi-island genetic algorithm. J Chin Inst Eng 42(4):297–308

22. Shao Y, Yan L, Guo P, Yang H, Shi F, Feng D (2019) A comprehensive study on fitness approximation techniques in shape optimization of aerofoil forging preform tools. Metals 9(6):617

23. Naeimi A, Mousavi ML, Eftekhari A (2014) Optimum designing of forging preform die for the H-shaped parts using backward deformation method and neural networks algorithm. J Modern Processes Manuf Prod 3(3):79–96 (**Islamic Azad University**)

24. Guan Y, Bai X, Liu M, Song L, Zhao G (2015) Preform design in forging process of complex parts by using quasi-equipotential field and response surface methods. Int J Adv Manuf Technol 79(1–4):21–29

25. Torabi SHR, Alibabaei S, Barooghi Bonab B, Sadeghi MH, Faraji G (2017) De-sign and optimization of turbine blade preform forging using RSM and NSGA II. J Intell Manuf. https://doi.org/10.1007/s10845-015-1058-0

26. Seungro L, Luca Q, Donghwi P, Inwoo K, Juhyun S, Naksoo K (2021) A New approach to preform design in metal forging processes based on the convolution neural network. MDPI Appl Sci 11:7948

27. Caspari M, Landkammer P, Steinmann P (2018) A non-invasive node-based form finding approach with discretization-independent target configuration. Adv Model Simul Eng Sci 5(1):11

28. Chenot J-L, Massoni E, Fourment L (1996) Inverse problems in finite element simula-tion of metal forming processes. Eng Computations 13:190–225

29. Liu C et al (2021) Optimal design of preform shape based on EFA-FEM-GA integrated methodology. Int J Mater Form 14:1043–1056. https://doi.org/10.1007/s12289-021-01620-0

30. Liu C et al (2021) Pre-forging shape design using conformal mapping method. In: Daehn G, Cao J, Kinsey B, Tekkaya E, Vivek A, Yoshida Y (eds) Forming the future. The minerals, metals & materials series. Springer, Cham. https://doi.org/10.1007/978-3-030-75381-8_241

31. Biba N et al (2020) Closed die forging preform shape design using isothermal surfaces method. Procedia Manuf 47:268–273. https://doi.org/10.1016/j.promfg.2020.04.219. (**ISSN 2351-9789**)

32. Biba N et al (2021) Preform design in axial hot closed die forging by isothermal surface method. Part 1 overview of preform design methods. Theoretical aspects and algorithm of preform shape design. Blanking Prod Mech Eng (press forging, foundry and other productions). https://doi.org/10.36652/1684-1107-2021-19-5-214-220

33. Hedicke-Claus Y, Langner J, Stonis M, Behrens B-A (2019) Klassifizierung von Schmiedeteilen mittels KNN, wt-online 11/12-2019, pp. 822–827

34. Hedicke-Claus Y, Roe C, Kriwall M, Stonis M, Behrens B-A (2021) Komplexitäts-Scoremodell für Schmiedeteile. In: WT WERK-STATTSTECHNIK, VDI Fachmedien GmbH, BD. 111, Jg., Nr.6, S. 458–463. https://doi.org/10.37544/1436-4980-2021-06-102

35. Dietrich J (2018) Praxis der Umformtechnik–Umform- und Zerteilverfahren, Werkzeuge, Maschinen. 12., überarb. und erw. Aufl., Springer Vieweg, Wiesbaden https://doi.org/10.1007/978-3-658-19530-4

36. Hedicke-Claus Y, Langner J, Stonis M, Behrens BA (2018) Innovativer Ansatz zur effizienten Stadienplanung. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 113(10):668–672

37. Hedicke-Claus Y, Kriwall M, Langner J, Stonis M, Behrens B-A (2021) Validation of automatically generated forging sequences by using FE simulations. In: Daehn G, Cao J, Kinsey B, Tekkaya E, Vivek A, Yoshida Y (eds) Forming the future. The minerals, metals & materials series. Springer, Cham. https://doi.org/10.1007/978-3-030-75381-8_238

38. Behrens B-A, Röhr S, Schäfer F, Hundertmark A (2007) Untersuchungen zur numerischen Ermittlung von Schmiedefalten. UTF-Science, 2

39. Hensel A, Spittel T (1978) Kraft- und Arbeitsbedarf bildsamer Formgebungsverfahren, Deutscher Verlag für Grundstoffindustrie