

Optimization of forging processes using Finite Element simulations

A comparison of Sequential Approximate Optimization and other algorithms

Martijn H. A. Bonte · Lionel Fourment · Tien-tho Do ·
A. H. van den Boogaard · J. Huétink

Received: 22 October 2008 / Revised: 25 May 2010 / Accepted: 14 July 2010 / Published online: 12 August 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract During the last decades, simulation software based on the Finite Element Method (FEM) has significantly contributed to the design of feasible forming processes. Coupling FEM to mathematical optimization algorithms offers a promising opportunity to design *optimal* metal forming processes rather than just *feasible* ones. In this paper Sequential Approximate Optimization (SAO) for optimizing forging processes is discussed. The algorithm incorporates time-consuming nonlinear FEM simulations. Three variants of the SAO algorithm—which differ by their sequential improvement strategies—have been investigated and compared to other optimization algorithms by application to two forging processes. The other algorithms taken into account are two iterative algorithms (BFGS and SCIP) and a Metamodel Assisted Evolutionary Strategy (MAES). It is essential for sequential approximate optimization algorithms to implement an improvement strategy that uses as much information obtained during previous iterations as possible. If such a sequential improvement strategy is used, SAO provides a very efficient algorithm to optimize forging processes using time-consuming FEM simulations.

Keywords Sequential Approximate Optimization · Optimization algorithms · Forging processes · Finite Element simulations · Forging

M. H. A. Bonte · A. H. van den Boogaard (✉) · J. Huétink
University of Twente, P.O. Box 217,
7500 AE, Enschede, The Netherlands
e-mail: a.h.vandenboogaard@utwente.nl

L. Fourment · T.-t. Do
Mines ParisTech, CEMEF, CNRS, UMR 7635,
BP 207, 06904 Sophia Antipolis Cedex, France

1 Introduction

At the end of the previous century, the Finite Element Method (FEM) has become an important tool for designing feasible metal forming processes. More recently, several authors recognised the potential of coupling FEM simulations to mathematical optimization algorithms to design *optimal* metal forming processes instead of just *feasible* ones. Early 2D non-steady forging optimizations were presented by Fourment and Chenot (1996), Fourment et al. (1996), Zhao et al. (1997) and an early 3D extension was given by Laroussi and Fourment (2004). A critical issue in the application of optimization algorithms to forming processes is the long calculation time for one single function evaluation by FEM, typically of the order of one hour to several hours for 2D problems and several hours to several days on a parallel computer for 3D problems. Therefore, optimization algorithms should be used that require only a limited number of function evaluations. The efficiency of sequential approximate optimization algorithms, compared to some other methods applied to forging is the main focus of this paper.

A way of optimizing metal forming processes is using classical iterative optimization algorithms (Conjugate Gradient, BFGS, etc.), where each function evaluation means running a FEM calculation, see e.g. Naceur et al. (2001), Vielledent and Fourment (2001), Kleinermann and Ponthot (2003), Lin et al. (2003). These algorithms are well-known, but suffer from a number of disadvantages: function evaluations are inherently sequential, require difficult to obtain sensitivities, and may be trapped in local optima.

Several authors have tried to overcome these disadvantages by applying genetic or evolutionary optimization algorithms, see e.g. Castro et al. (2004), Schenk and Hillmann (2004), Fourment et al. (2005), Habbal et al. (2008). Genetic

and evolutionary algorithms look promising because of their tendency to find the global optimum and the possibility for parallel computation. However, they are known to require many function evaluations (Emmerich et al. 2002).

A third alternative is using approximate optimization algorithms such as Response Surface Methodology (RSM) or Kriging (DACE). Classical RSM is based on fitting a lower order polynomial metamodel through response points allowing for a random error, Kriging interpolates exactly through these response points. Approximate optimization algorithms allow for parallel computing, tend to find global optima and do not need sensitivities. However, the accuracy of the obtained optimum completely depends on the accuracy of the metamodel. Examples where approximate optimization algorithms are applied to optimize metal forming processes include Jansson et al. (2003, 2005), Naceur et al. (2004), Bonte et al. (2007).

This paper describes Sequential Approximate Optimization algorithms to optimize forging processes using time-consuming Finite Element simulations. The sequential improvement aims at achieving an accurate solution of the global optimum with the lowest possible number of FE simulations. Two advanced methods are considered in Section 2: ‘minimising a merit function’ according to Emmerich et al. (2002) and ‘maximum expected improvement’ according to Schonlau (1997). Subsequently, their performance is compared to that of other algorithms (two iterative algorithms and an evolutionary strategy) by application to two forging processes in Section 3.

2 Sequential Approximate Optimization algorithm

The Sequential Approximate Optimization (SAO) algorithm using time-consuming FEM simulations is presented in Fig. 1. The several stages will be explained one by one. Sections 2.1 through 2.4 shortly describe the initial (non-sequential) metamodel based optimization algorithm, as described in Bonte et al. (2007). The Sequential Approximate Optimization algorithm extends this approach with the sequential improvement strategies presented in Section 2.5. These dramatically increase the efficiency of the algorithm. In Section 2.6 three other algorithms are introduced that are used for comparison with the SAO approach in Section 3.

The optimization algorithm presented here is implemented in MATLAB and can be used in combination with any Finite Element code. It may also be applied to applications other than forging, for which performing many function evaluations is time-consuming or otherwise prohibitive. For the fitting of the DACE/Kriging metamodels, use was made of the MATLAB Kriging toolbox implemented by Lophaven, Nielsen and Søndergaard (Nielsen 2002; Lophaven et al. 2002a, b).

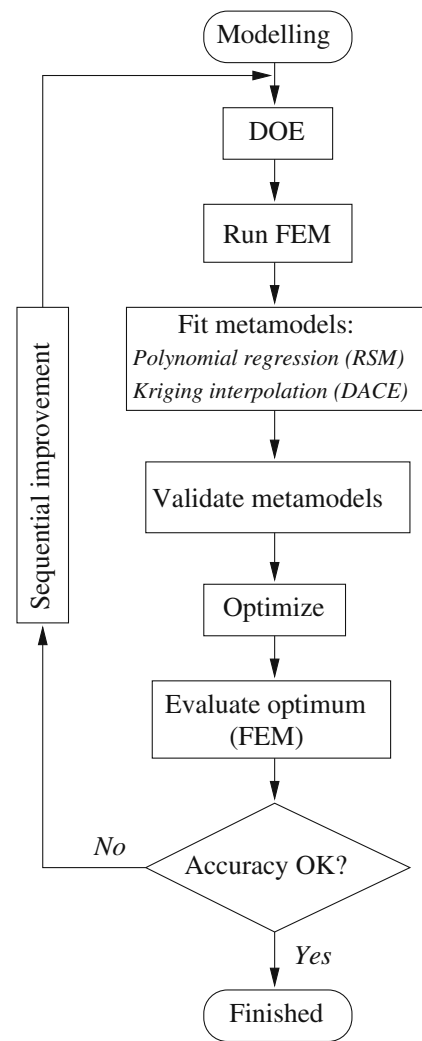


Fig. 1 Sequential Approximate Optimization algorithm

2.1 Modelling

The first step is to create the optimization model, i.e. quantifying objective function and constraints, and selecting the design variables. The combination of a structured modelling procedure and an optimization algorithm for solving the modelled optimization problem is referred to as an *optimization strategy*. In the remainder of this paper, the focus is on the solving part of such an optimization strategy rather than on the modelling part. A structured approach to the construction of an optimization model for metal forming processes can be found in Bonte (2007) and Bonte et al. (2008).

2.2 Design Of Experiments (DOE)

When the optimization problem has been modelled a number of design points is selected by a Design Of Experiments

(DOE) strategy. A space-filling Latin Hypercube Design (LHD) is a good and popular DOE strategy for constructing metamodels from deterministic computer experiments such as Finite Element calculations (McKay et al. 1979; Santner et al. 2003) and has been selected for the SAO algorithm. A typical size of an initial space-filling LHD for computer experiments exists of 10 times the number of design variables (Schonlau 1997).

However, when a metamodel is used for optimization, it is important that the metamodel gives accurate results in the neighbourhood of the optimum. Often, this optimum lies on the boundary of the design space. Therefore, an accurate prediction is needed on the boundary, which implies performing measurements on that boundary. An LHD will generally provide design points in the interior of the design space and not on the boundary. Therefore, the LHD is combined with a Resolution III or IV fractional factorial design, which puts DOE points in corners of the design space. This method was also proposed by Kleijnen and van Beers (2008).

2.3 Running the FEM simulations and fitting the metamodels

The responses (objective function and constraints) are evaluated at the design points given by the DOE strategy, running parallel FEM calculations. The next step is to fit metamodels for each response. Two metamodeling techniques are considered: Response Surface Methodology (RSM) and Kriging or Design and Analysis of Computer Experiments (DACE).

Using RSM, the response measurements \mathbf{y} are presented as the sum of a lower order polynomial metamodel and a random error term ϵ (Myers and Montgomery 2002). A metamodel based on RSM can be used to predict the response y_0 of an unknown design variable setting \mathbf{x}_0 :

$$\hat{y}_0 = \mathbf{x}_0^T \hat{\boldsymbol{\beta}} \tag{1}$$

where $\hat{\boldsymbol{\beta}}$ are the regression coefficients obtained by least squares regression. It is also possible to determine the variance at this location (Myers and Montgomery 2002):

$$\text{var}(\hat{y}_0) = \sigma^2 \mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0 \tag{2}$$

where σ^2 is the error variance. The variance at \mathbf{x}_0 is used in the sequential improvement strategies in Section 2.5.

Four possible shapes of RSM metamodels are commonly applied. They are in ascending complexity:

- linear
- linear + interaction

- pure quadratic or elliptic
- (full) quadratic

Sacks et al. (1989a, b) proposed Kriging or DACE to fit metamodels using deterministic computer experiments. A metamodel based on Kriging interpolates between the measurement points and exactly matches the calculated response at the design points. In this work, a Gaussian exponential correlation function is adopted. Gaussian exponential functions are intuitively attractive because they are infinitely differentiable. Moreover, Gaussian exponential functions are frequently used in literature (Santner et al. 2003) and have been found to give accurate results (Lophaven et al. 2002a).

Analogously to RSM, Kriging can be used to predict the response y_0 at an unknown location \mathbf{x}_0 , see e.g. Koehler and Owen (1996), Lophaven et al. (2002b), Santner et al. (2003), Martin and Simpson (2005):

$$\hat{y}_0 = \mathbf{x}_0^T \hat{\boldsymbol{\beta}} + \mathbf{r}^T \hat{\mathbf{R}}^{-1} (\mathbf{y} - \mathbf{X} \hat{\boldsymbol{\beta}}) \tag{3}$$

where $\hat{\boldsymbol{\beta}}$ are the regression coefficients, \mathbf{r} is the matrix containing the correlation between \mathbf{x}_0 and the DOE points, \mathbf{R} the matrix containing the correlation between the DOE points themselves, \mathbf{y} the response measurements, and \mathbf{X} the design matrix containing the DOE points.

A measure for the predicted variance at this location is given by the *Mean Squared Error (MSE)*, see e.g. Martin and Simpson (2005):

$$\text{MSE}(y_0) = \sigma_z^2 \left(1 - \begin{bmatrix} \mathbf{x}_0^T & \mathbf{r}^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{X}^T \\ \mathbf{X} & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{r} \end{bmatrix} \right) \tag{4}$$

where σ_z^2 is the process variance. The Mean Squared Error is, again, used in the sequential improvement strategy in Section 2.5.

Several Kriging metamodels can be distinguished based on the order of their trend functions:

- Kriging with a zeroth order trend function
- Kriging with a first order trend function
- Kriging with a second order trend function

Computer simulations such as the Finite Element Method are thought of as being deterministic (Sacks et al. 1989a, b), which is in favour of Kriging as a metamodeling technique. However, numerical noise due to for example finite element discretization, automatic mesh refinement or step size adjustment may be present, which pleads for using RSM or modified Kriging models for inaccurate data as presented in Sakata et al. (2007).

2.4 Metamodel validation, optimization and accuracy evaluation

Selecting the best metamodel for each response is done by metamodel validation. Metamodel validation for RSM is based on ANalysis Of VAriance (ANOVA) and residual plots, see e.g. Myers and Montgomery (2002). Metamodel validation for Kriging is based on cross validation. Using cross validation, one leaves out one, say the i^{th} , of the response measurements and fits the metamodel through the remaining response measurements. The difference between the measured value y_i and the value predicted by the metamodel at this location \hat{y}_{-i} is a measure for the accuracy of the metamodel. One can repeat this procedure for all say n measurement points and calculate the cross validation Root Mean Squared Error (RMSE_{CV}):

$$\text{RMSE}_{\text{CV}} = \sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_{-i})^2}{n}} \quad (5)$$

As RMSE_{CV} approaches 0, the metamodel becomes more and more accurate.

For each response (objective function and constraints) the metamodel outperforming the other six metamodels is selected. These best metamodels for objective function and constraints are subsequently optimized using a standard Sequential Quadratic Programming (SQP) algorithm. To avoid convergence to a local minimum, the SQP algorithm is restarted at all DOE points. Since the evaluation takes place on the metamodels—being explicit mathematical functions—the evaluation time remains small compared to a complete nonlinear FEM analysis.

The obtained approximate optimum is finally checked by running one last FEM calculation with the approximated optimal settings of the design variables. In addition to metamodel validation, the difference between the approximate objective function value and the value of the objective function calculated by the last FEM run is a measure for the accuracy of the obtained optimum. If the user is not satisfied with the accuracy, the Sequential Approximate Optimization algorithm allows for sequential improvement.

2.5 Sequential improvement

Sequential improvement implies adding new DOE points to the initial DOE. Goal of sequential improvement is to improve the accuracy of the metamodel and hence the accuracy of the approximate optimum. Three different sequential improvement strategies are considered:

- adding new DOE points in a space-filling way (SAO-SF),

- adding new DOE points by Minimising a Merit Function (SAO-MMF),
- adding new DOE points by Maximising Expected Improvement (SAO-MEI).

SAO-SF The first variant (SAO-SF) simply adds new DOE points in a space-filling way. The same space-filling Latin Hypercube Design that has been introduced in Section 2.2 is used for this.

In earlier work (Bonte et al. 2007) a zooming strategy was investigated. A new metamodel was created in a limited zone near the approximate optimum. However, ongoing research revealed two drawbacks for this strategy. Firstly, zooming is not trivial in case of multiple local optima and secondly, it is not efficient since time-consuming calculations performed in previous iterations are disregarded if they are outside the zoomed-in design space. For several analytical test functions this approach was found to be less efficient than a space-filling sequential improvement. Therefore, the following—advanced—sequential improvement strategies are compared to the straightforward space-filling improvement.

SAO-MMF The second sequential improvement strategy makes use of all the information obtained during previous iterations of the algorithm, i.e. metamodels of the objective function \hat{y} and its standard deviation \hat{s} .

To explain this, consider the Kriging metamodel of an objective function depicted in Fig. 2. The cross marks in the figure denote the response measurements from the previous iteration(s). At an untried design variable setting x_0 , the predicted objective function value is \hat{y} , using (1) or (3). The uncertainty at this location can be modelled as a normal

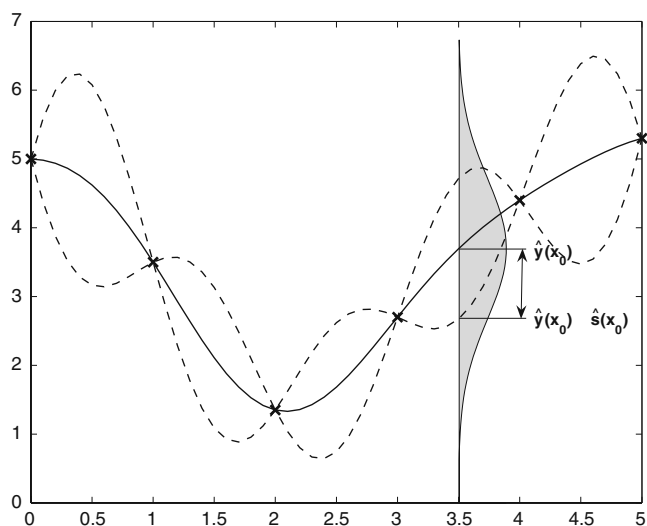


Fig. 2 Sequential improvement employing metamodel information of the previous iteration

distribution with standard deviation \hat{s} as shown in the figure. For RSM, \hat{s} equals the square root of the variance in (2), for Kriging it equals the square root of the Mean Squared Error (RMSE) in (4).

The second sequential improvement strategy (SAO-MMF) selects the new DOE points based on the merit function:

$$f_{\text{merit}}(\mathbf{x}) = \hat{y}(\mathbf{x}) - w \cdot \hat{s}(\mathbf{x}) \tag{6}$$

where \hat{y} and \hat{s} are for both RSM and Kriging given by meta-models from previous iterations of the algorithm. w is a weight factor.

Emmerich et al. (2002) use the same merit function for making evolutionary algorithms more efficient. They propose the ‘‘Metamodel-Assisted Evolution Strategy’’ (MAES) that is also included in the comparison in Section 3. MAES is shortly introduced further in Section 2.6.

Alternatively, Torczon and Trosset (1998) propose to minimise the merit function in (6). However, instead of the standard deviation in (6) they use the distance between a possible new candidate point and an already evaluated point as a measure for the error of the metamodel. Based on this approach, Buche (2004) and Buche et al. (2005) use the RMSE of Kriging-like Gaussian Processes instead of the distance. Buche determines the minimisation of the merit function by an evolutionary strategy.

Here, the SAO-MMF algorithm minimises the merit function of (6) with the multistart SQP algorithm introduced in Section 2.4. A remaining question is how to set the value of the weight factor w . If one selects $w = 0$, the new DOE points equal the optima of the metamodel \hat{y} . If $w \rightarrow \infty$, the new DOE points are added in a space-filling way. It was found that $w = 1$ provides a good compromise between both extreme cases.

To illustrate how SAO-MMF selects new DOE points, consider Fig. 3. The figure shows the same Kriging metamodel as the one in Fig. 2 (the dotted line). The metamodel of its merit function f_{merit} is also visualised as the dashed line. One can easily obtain the (in this case) five minima of the merit function by applying the multistart SQP algorithm. Note that only those minima of the merit function are taken into account that actually promise improvement with respect to the best objective function value f_{min} obtained in previous iterations of the algorithm. In this example, this saves running three time-consuming FEM simulations: only two FEM simulations have to be run instead of five in case all minima of the merit function would have been taken into account.

SAO-MEI A third sequential improvement strategy has been proposed by Schonlau (1997) and aims at maximising the expected improvement of an untried point. This method is also reported in Jones et al. (1998), Jones (2001),

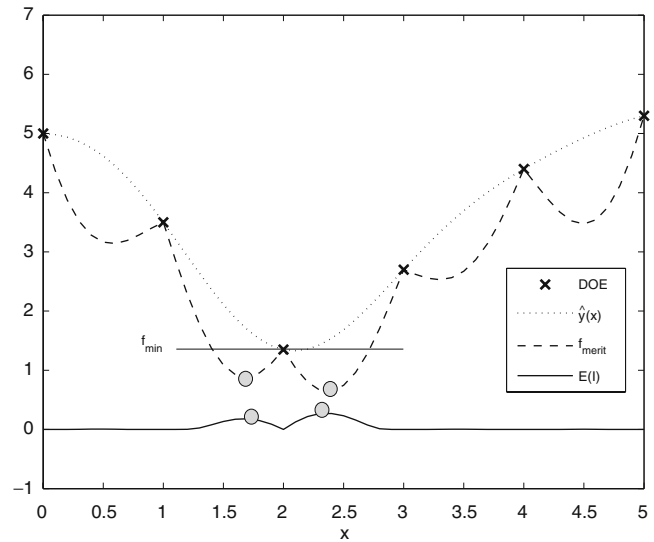


Fig. 3 SAO-MMF vs. SAO-MEI

Santner et al. (2003), Sasena et al. (2002) and—just as SAO-MMF—fully exploits all information available from previous iterations of the algorithm.

The method starts by defining *Improvement I* as Schonlau (1997):

$$I = \begin{cases} f_{\text{min}} - y & \text{if } y < f_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where f_{min} is the lowest objective function value obtained during earlier iterations and y is a possible new outcome of a function evaluation. Clearly, if $y < f_{\text{min}}$, the situation has improved. In general, the expected value of a stochastic variable X is defined as:

$$E(X) = \int_{-\infty}^{\infty} xp(x)dx \tag{8}$$

in which x is a possible value of X and $p(x)$ is the probability that X actually obtains this value. $p(x)$ is the probability density function. Assuming a normal distribution, combination of (7) and (8) yields the *Expected Improvement*:

$$E(I) = \int_{-\infty}^{f_{\text{min}}} (f_{\text{min}} - y)\phi(y)dy \tag{9}$$

where $\phi(y)$ is the normal probability density function. Now y can be replaced by the metamodel of the objective function \hat{y} and (9) may be rewritten to Schonlau (1997), Jones et al. (1998):

$$E(I) = (f_{\text{min}} - \hat{y}) \Phi\left(\frac{f_{\text{min}} - \hat{y}}{\hat{s}}\right) + \hat{s}\phi\left(\frac{f_{\text{min}} - \hat{y}}{\hat{s}}\right) \tag{10}$$

$$E(I) = 0 \quad \text{if} \quad \hat{s} = 0$$

where \hat{y} is the objective function metamodel and \hat{s} its standard deviation as depicted in Fig. 2. ϕ and Φ denote the probability density and the cumulative distribution functions of the standard normal distribution. Schonlau (1997) proposes to maximise $E(I)$ in (10) to yield the point promising the Maximum Expected Improvement (MEI). Only this point is subsequently evaluated.

For the SAO algorithm, it is proposed to exploit the possibility for parallel computing and to include all points that (locally) Maximise Expected Improvement. This algorithm is referred to as SAO-MEI. The Expected Improvement function of (10) is again maximised by the multistart SQP algorithm. Difficulties for maximising the Expected Improvement function are indicated by Jones et al. (1998): (i) the function can be extremely multimodal with flat planes in between, which makes it difficult to optimize; and (ii) the Expected Improvement is 0 by definition for DOE points that have already been calculated, thus the SQP algorithm cannot be started from the DOE points. These problems have been overcome by starting the SQP algorithm from a dense grid of newly generated design points.

Figure 3 also shows the expected improvement function of the Kriging metamodel introduced in Fig. 2 as the solid line. Note that the function is indeed multimodal with large flat planes, and that the function is zero at locations where simulations have already been performed.

SAO-MEI is similar to SAO-MMF introduced in the previous paragraph. Both make use of all information available from previous iterations. They both tend to select new DOE points in the region where the global optimum is predicted to be (\hat{y} is small). Additional points are selected where no points have been sampled before (\hat{s} is large). Note from Fig. 3 that the points obtained by SAO-MEI are at similar, but not the same locations as the points obtained by SAO-MMF.

One of the differences between both methods is that SAO-MEI includes the best value f_{\min} directly in the expected improvement function. However—as discussed earlier—for SAO-MMF it is recommended to run only the FEM simulations that promise to be better than f_{\min} , which reduces the difference between both methods.

2.6 Other algorithms

In the next section, SAO-SF, SAO-MMF and SAO-MEI will be compared to each other by application to forging. Other algorithms will also be taken into account. This section shortly introduces the other algorithms: two iterative algorithms (BFGS and SCIP), and an efficient evolutionary strategy.

BFGS The well-known BFGS algorithm (Broyden 1970) is usually the most efficient quasi-Newton method for optimization problems. It makes it possible to find satisfactory solutions within few iterations when the objective function is convex, for instance as in Vielledent and Fourment (2001) for 2D forging applications. It requires computing the gradient, which is not a simple issue. In the FEM code used for the forging applications in Section 3, the adjoint state method is used and the partial derivatives are calculated with a semi-analytical approach (Laroussi and Fourment 2004). When the optimization problem is more complex, the solution may depend on the starting point and the algorithm gets trapped into local optima.

SCIP In order to escape from local extrema, the method of moving asymptotes (Svanberg 1987) has been proposed. A convex envelope of the objective function is built during optimization iterations, using a convex approximation and a family of rational functions. The utilised SCIP algorithm (Zillober 2001) represents one of its variants derived from Sequential Convex Programming. It is particularly suited for constrained optimization problems.

MAES Evolutionary algorithms are regarded as the most robust with respect to local extrema, making it possible to solve the most complex optimization problems. Evolutionary algorithms typically consist of the selection-recombination-mutation process. This process in combination with FEM simulations as function evaluations is presented in Fig. 4.

Evolutionary Strategies (ES) are similar to Genetic Algorithms (GA), with slight differences. ES use real coding, mutation is the main genetic operator while recombination

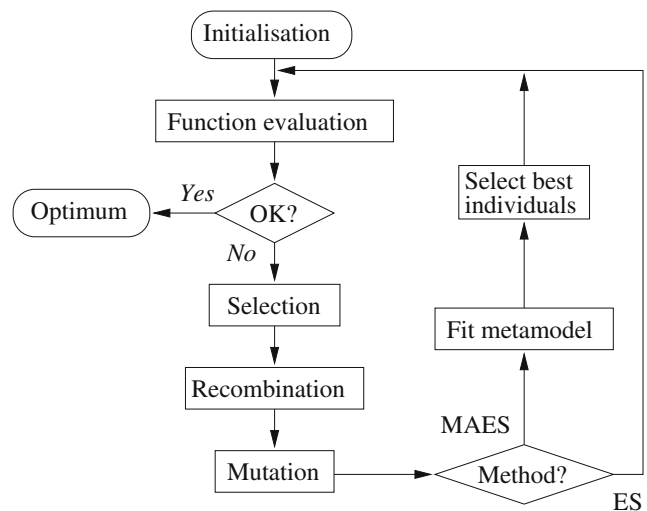


Fig. 4 Overview of metamodel assisted and direct evolutionary strategy

is not systematically used. The selection of parents is simpler, with only two strategies, the “plus” (parents are kept in the new generation) and the “comma” (parents do not survive) ones. In general, ES can find a solution more rapidly, whereas GA would find a more global extremum. More general information on Evolutionary Strategies and Genetic Algorithms can be found in e.g. Schwefel (1995), Bäck and Schwefel (1993), Beyer and Schwefel (2002).

However, as mentioned in the introduction, the costs of both ES and GA are usually high in terms of function evaluations, which is considered a serious disadvantage when each function evaluation is a time-consuming FEM simulation. The utilised Metamodel-Assisted Evolution Strategy (MAES) proposed by Emmerich et al. (2002) combines an Evolutionary Strategy (ES) with Kriging metamodels to reduce the number of function evaluations. Hence, one could refer to MAES as a hybrid algorithm combining two of the three groups of algorithms reviewed in Section 1: evolutionary algorithms and approximate optimization.

MAES is also depicted in Fig. 4. The specific algorithm that was used comprises a regular (2 + 10)-ES. It starts by randomly choosing an initial population of three times the number of design variables. After having run the FEM simulations for this initial population, the two best settings are selected, recombined and mutated to yield ten children. Before running these children settings, the results of the previously performed FEM calculations are used to fit a Kriging metamodel. Instead of running the FEM calculations for the ten children directly, the results are first predicted by the merit function of (6) where the weight factor w is taken equal to 1. Based on the prediction, the best 20% of the individuals are subsequently evaluated by running FEM simulations for the corresponding design variable settings. In this way, the application of metamodeling techniques saves eight time-consuming FEM calculations per generation with respect to a regular ES.

Subsequently, the procedure is repeated: the two best settings are selected again, recombined and mutated to yield ten new children, the Kriging metamodel is updated, and the performance of the new children is predicted by the new merit function. This procedure is continued until one is satisfied with the results.

3 A comparison between the optimization algorithms by application to forging

In this section, the three proposed Sequential Approximate Optimization algorithms will be applied to two industrial forgings: a spindle (Section 3.1) and a gear (Section 3.2) with realistic objective functions. The algorithms will be

compared to each other, and to the other optimization algorithms introduced in Section 2.6.

In both examples, finite element calculations are performed with the code Forge3. Elements with linear velocity and pressure interpolation are used with a bubble function enrichment of the velocity field. The material is modelled by an incompressible rigid-viscoplastic Norton–Hoff law:

$$\mathbf{s} = 2K \left(\sqrt{3} \dot{\epsilon}_{eq} \right)^{m-1} \dot{\epsilon} \tag{11}$$

where \mathbf{s} represents the deviatoric stress, $\dot{\epsilon}$ the strain rate tensor and $\dot{\epsilon}_{eq} = \sqrt{2/3} \dot{\epsilon} : \dot{\epsilon}$ the equivalent strain rate; K is the material consistency and m its strain rate sensitivity. As a first approach, the calculations consider isothermal conditions.

Both products, the spindle and the gear, are obtained in two forging steps, and the main issue of the process design regards the shape of the forming tools. While the die geometries for the last step are straightforwardly derived from the required product shape, the shape of the preform dies can have very varied forms depending on the designer and pursued objectives. They have strong influence on the product quality and the process efficiency. In particular, the preform shapes determine the correct filling of the die and surface defects like folding, as well as the energy consumption in the final forging step. Therefore, the shape of the preforms is the current subject of the optimization problems. The simulations were performed on a 2.4GHz Intel Pentium 4 desktop computer with 2GB memory.

In Section 3.3 the results of the comparison of the algorithms are discussed.

3.1 Spindle

The optimization problem The first case is presented in Fig. 5. It regards the hot forging of a spindle out of a cylindrical billet with a diameter of 50 mm and height of 90 mm

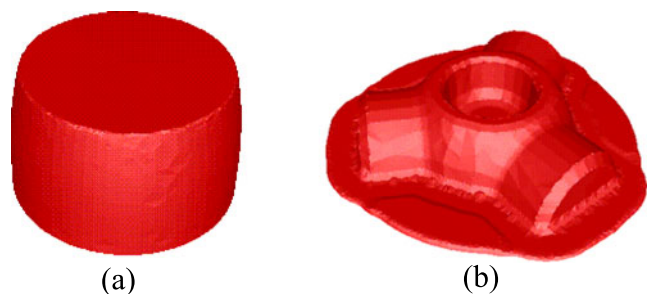
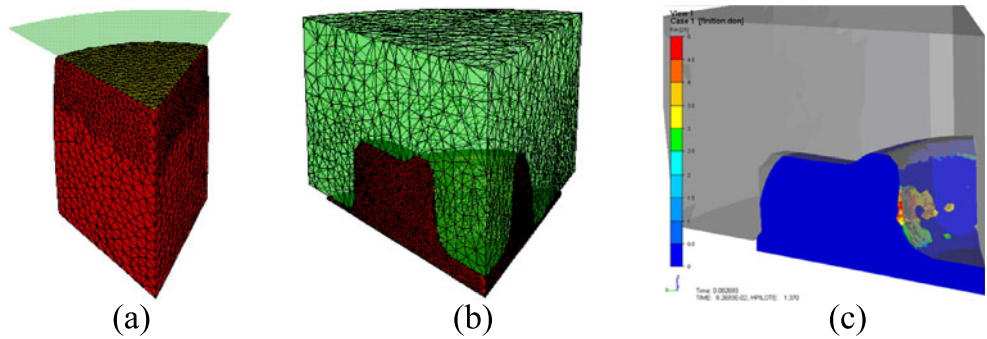


Fig. 5 The spindle: a The preform; b The final product

Fig. 6 (a) The FE model after upsetting; (b) The FE model after forging; (c) Folding



made of 38MnSi4 steel. The two process operations take place at a temperature of 1100 °C: a preform is first made by upsetting the cylinder between *simple* dies, before being forged into the final product.

To evaluate whether the final product can be made in the factory, a Finite Element (FEM) calculation was performed. The material parameters for the Norton–Hoff law are $K = 1809$ MPa and $m = 0.14$. Figure 6a and b show the FE models after the two operations. Use has been made of symmetry, so only one twelfth of the component was considered, using 6-fold symmetry in the plane and 2-fold symmetry along the horizontal plane. The two-stepped process simulation, took about four hours with the utilised computer. In Fig. 6c, one can observe that a folding defect occurs, which deteriorates the final product quality. The depicted quantity is the equivalent plastic strain rate at the free surface of the product, i.e. where no contact exists between the product and the die.

To overcome the folding defect, the geometry of the preform is optimized with as goal to minimise the equivalent plastic strain rate at the free surface during forging. This objective function was proposed by Fourment et al. (1998) and has been used successfully since. The objective function is formulated as follows:

$$\Phi_{\text{fold}} = \frac{1}{t_{\text{end}} - t_0} \int_{t=t_0}^{t_{\text{end}}} \left(\frac{1}{\Omega_{\text{ft,ref}}} \int_{\Omega_{\text{ft}}} \left(\frac{\dot{\epsilon}_{\text{eq}}}{\dot{\epsilon}_{\text{eq,ref}}} \right)^\alpha ds \right)^{\frac{1}{\alpha}} dt \tag{12}$$

t denotes the time, Ω_{ft} is the free surface of the discretized domain at time t and $\Omega_{\text{ft,ref}}$ the reference free surface at time $t = t_0$. $\dot{\epsilon}_{\text{eq}}$ and $\dot{\epsilon}_{\text{eq,ref}}$ are the equivalent strain rate and a reference equivalent strain rate and α is an amplification factor, which is selected to be equal to 10.

The geometry of the axi-symmetric preform die is modelled by the B-spline shown in Fig. 7 as the thick line. The B-spline is determined by the six control points $C_1 \dots C_6$ with radial coordinates of 0, 10, 20, 30, 35 and 40 mm respectively. The vertical position of the control points C_1 to C_4 , relative to C_5 and C_6 are parameterized by the three design

variables μ_1 to μ_3 as presented in Fig. 7. Control point C_1 is placed on the symmetry axis. Note that C_1 and C_2 will always have the same vertical position, leading to a horizontal tangent at the centre. All design variables are allowed to vary between -10 and 20 mm. In the initial preform die, all control points are on the same vertical position, 55 mm above the plane of symmetry. The stroke of 25 mm (in the symmetric model) is fixed to control points C_5 and C_6 . For this optimization problem, the material flow is rather simple and the forging force and die filling are not an issue, therefore no other constraints are present.

The total optimization problem can now be modelled as follows:

$$\begin{aligned} \min \Phi_{\text{fold}}(\mu_1, \mu_2, \mu_3) \\ \text{s.t.} \quad & -10 \text{ mm} \leq \mu_1 \leq 20 \text{ mm} \\ & -10 \text{ mm} \leq \mu_2 \leq 20 \text{ mm} \\ & -10 \text{ mm} \leq \mu_3 \leq 20 \text{ mm} \end{aligned} \tag{13}$$

Applying the optimization algorithms The proposed Sequential Approximate Optimization algorithm, as well as the optimization algorithms introduced in Section 2.6, are now applied to solve the optimization problem modelled in (13).

Table 1 presents the results. The table shows for all algorithms the number of the FEM calculation N_{opt} , that

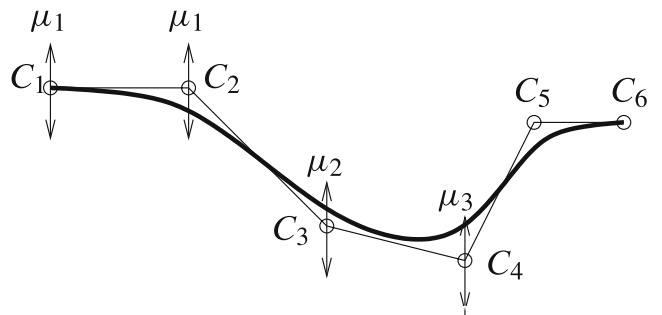


Fig. 7 The B-spline describing the geometry of the preform die for the spindle

Table 1 Results of optimizing the spindle

	N_{opt}/N_{tot}	μ_1 (mm)	μ_2 (mm)	μ_3 (mm)	Φ_{fold}	OK?
Initial	–	0	0	0	10.49	No
BFGS	3/12	7.02	2.92	6.55	10.25	No
SCPIP	10/10	–4.25	–6.01	–3.07	9.17	Yes
MAES	30/48	–7.26	–3.84	–10	8.35	Yes
SAO-SF	36/50	–1.69	–5.95	8.33	9.20	No
SAO-MMF	38/48	–10	–6.50	–9.82	8.06	Yes
SAO-MEI	22/50	–8.52	–6.40	–10	7.97	Yes

gave the optimal settings, as a fraction of the total number of simulations N_{tot} performed for a specific algorithm. Additionally, it presents the optimal design variable settings and corresponding objective function values and it answers the question whether the folding defect has been solved or not as verified visually after an analysis with a refined mesh. Figure 8 visualises the initial preform geometry and geometries after optimization using different algorithms. The convergence of the optimization algorithms is depicted in Fig. 9.

The results will be discussed in Section 3.3 together with the results of the second forging application.

3.2 Gear

The optimization problem The second case regards the warm forming of a steel gear at 600 °C out of a preform shape as presented in Fig. 10. Just as was the case for the spindle, forming the gear is a two step process, but this time,

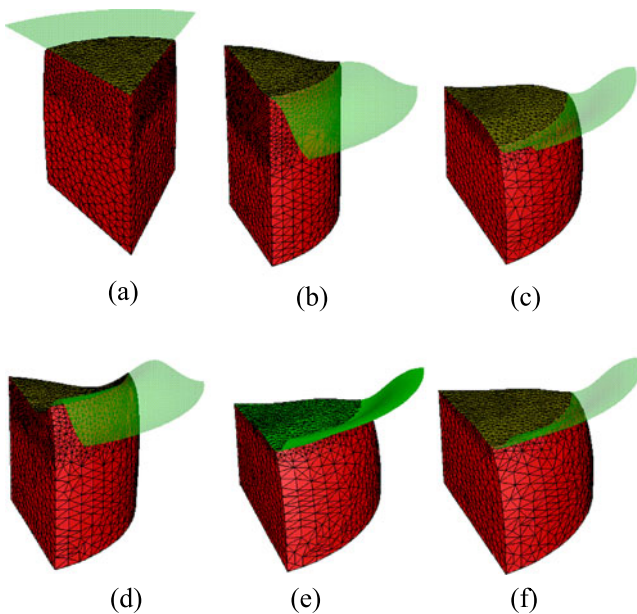


Fig. 8 Preforms of the spindle: **a** Initial; **b** BFGS; **c** MAES; **d** SAO-SF; **e** SAO-MMF; **f** SAO-MEI

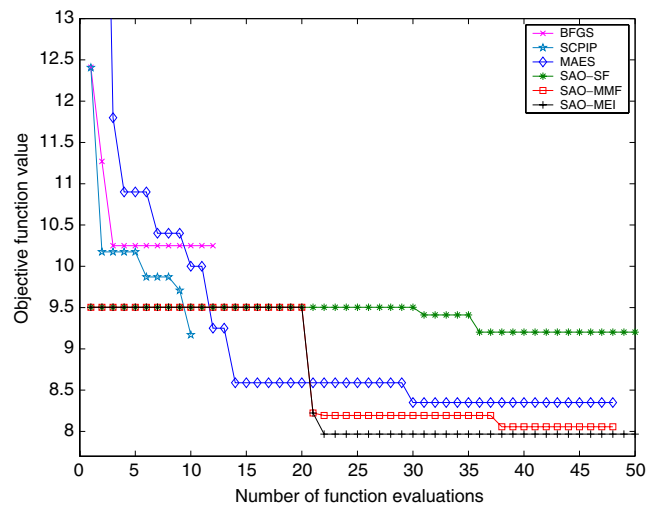


Fig. 9 Convergence of the algorithms for optimizing the spindle

the preform shape of the component rather than the shape of the preforming dies was regarded as the optimization objective. Consequently, it was not necessary to simulate the first forging operation. The resulting parts after both production steps are shown in Fig. 10.

To optimize the process, a Finite Element model was made. Figure 11a shows the FE model of the preform. Note that use has been made of the product symmetry, so only one twentieth of the component was considered. Initial calculations with a fine mesh and proper Norton–Hoff law took 12 hrs., which was considered too much for optimization purposes. To reduce the calculation time a relatively coarse mesh was used and the sensitivity index m in the Norton–Hoff law was set to 1.0, which modified the material flow in a not too significant manner, so that the optimization problem was not really changed. For this linearized case, the material consistency K does not affect the material flow and only acts as a multiplication factor for the forging force or energy. It does not influence the optimization results. With this setup one full simulation of the gear took about half an hour.

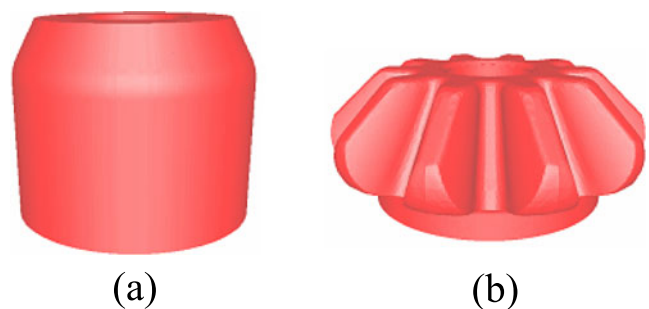


Fig. 10 The gear: **a** The preform; **b** The final product

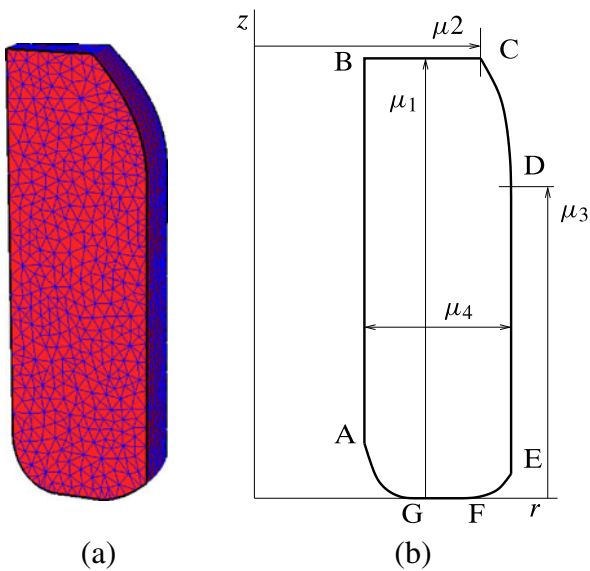


Fig. 11 (a) The 3D FE model of the gear; (b) The design variables describing the geometry of the preform of the gear

To limit the production costs and environmental pollution under warm forming conditions, it is essential to increase the tool life and consequently to minimise the forging force or the total energy required for forging the gear. An objective function is formulated as:

$$\Phi_{ene} = \int_{t=t_0}^{t_{end}} \left(\int_{\Omega_t} \sigma : \dot{\epsilon} dw + \int_{\Omega_{ct}} \tau \cdot v ds \right) dt \tag{14}$$

t denotes the time, Ω_t is the discretized domain at time t , σ and $\dot{\epsilon}$ are the stress tensor and strain rate tensor. In the second part of the equation, Ω_{ct} is the contact surface and τ and v are the surface shear stress and the relative velocity, respectively. To assure a final part without folding defects, the objective function for energy reduction is combined with the function already defined for the reduction of folding potential in (12) resulting in:

$$\Phi_{tot} = a \frac{\Phi_{fold}}{\Phi_{fold,ref}} + (1 - a) \frac{\Phi_{ene}}{\Phi_{ene,ref}} \tag{15}$$

where the weight factor a is chosen equal to 0.5. To make the two objective functions comparable to each other, they are normalised by $\Phi_{fold,ref}$ and $\Phi_{ene,ref}$.

The design variables μ_1, μ_2, μ_3 and μ_4 , describing the preform geometry are presented in Fig. 11b. Requiring a specific volume to fill the final die, μ_4 can be expressed as a function of the other three:

$$\mu_4 = f(\mu_1, \mu_2, \mu_3) \tag{16}$$

Thus, a set of three design variables μ_1, μ_2 and μ_3 remains. The inner radius (points A and B) is fixed to 11.25 mm and the radius of points G and F are 16 and 20 mm respectively. The z -coordinates of points A and E are 5.6 and 2.47 mm respectively. The curves between points A and G and between F and E are represented by quadratic functions in r and the curve between C and D is represented by a quadratic function in z , such that there is a horizontal tangent in F and G and a vertical tangent in D. No constraints have been formulated except the box constraints bounding the design variables. These box constraints are included in the following optimization model that is used for optimizing production of the gear:

$$\begin{aligned} \min \Phi_{tot}(\mu_1, \mu_2, \mu_3) \\ \text{s.t. } 40 \text{ mm} \leq \mu_1 \leq 46 \text{ mm} \\ 18 \text{ mm} \leq \mu_2 \leq 22 \text{ mm} \\ 30 \text{ mm} \leq \mu_3 \leq 34 \text{ mm} \end{aligned} \tag{17}$$

The initial preform design variable values are $\mu_1 = 44.60$ mm, $\mu_2 = 21.65$ mm and $\mu_3 = 32.33$ mm. These design variables result in an objective function value Φ_{tot} of 1.19. The amount of energy (Φ_{ene}) and folding (Φ_{fold}) obtained for these design variable values are set to 100%.

Applying the optimization algorithms Now, the different optimization algorithms are compared to each other by applying them to the optimization problem modelled in (17). The obtained results are also compared to the results of the initial preform.

Table 2 presents the results. The table again shows for all algorithms the number of the FEM calculation N_{opt} , that

Table 2 Results of optimizing the gear

	N_{opt}/N_{tot}	μ_1 (mm)	μ_2 (mm)	μ_3 (mm)	Φ_{ene}	Φ_{fold}	Φ_{tot}
Initial	–	44.60	21.65	32.33	100%	100%	1.19
BFGS	3/7	40.91	18.00	30.11	N/A	N/A	1.157
SCPIP	3/21	45.80	19.80	30.00	N/A	N/A	1.123
MAES	42/49	45.67	18.36	30.25	–9.7%	–7.6%	1.079
SAO-SF	51/51	45.25	18.00	30.58	–8.7%	–6.6%	1.091
SAO-MMF	31/49	45.25	18.18	30.65	–8.2%	–9.8%	1.076
SAO-MEI	49/54	45.35	18.00	30.41	–9.4%	–9.8%	1.068

gave the optimal settings, as a fraction of the total number of simulations N_{tot} performed for a specific algorithm. Additionally, it presents the optimal design variable settings and corresponding objective function values as well as the reduction in energy and folding potential obtained by optimization. Figure 12 visualises the shapes of the initial preform and after optimization using different algorithms. The convergence of the algorithms is depicted in Fig. 13.

3.3 Discussion

Concluding this comparison between optimization algorithms, the results are discussed. For both the spindle and the gear, the results show similar trends.

Iterative algorithms BFGS and SCPIP For both forging cases, the iterative BFGS and SCPIP algorithms yielded relatively small improvements in only few iterations. For the spindle, the improvement proved to be too small to solve the folding defect. SCPIP performed significantly better than BFGS, since it is able to escape from some local minima. It is, however, not global thus it can get stuck in other local minima. In the case of the spindle, it managed to solve the defect. Hence, the improvement proved to be large enough. A similar trend emerged for optimizing the gear, although the difference between the two algorithms is not as significant.

As a return for being local, the BFGS and SCPIP algorithms require relatively few time-consuming FEM calculations for obtaining their optima. This makes them useful when a quick, but not too large improvement is required. For a more global convergence, other algorithms could be studied, such as GCMMA (Svanberg 1995; Bruyneel et al. 2002). However, an important disadvantage of such algorithms is the necessity to calculate sensitivities. Sensitivities are provided by the Forge3 code as described in Laroussi and Fourment (2004). The additional computational cost they require is about 30% of the total time. Meanwhile, they cannot readily be obtained for any forging configuration, and are not available on most commercial FE codes, which pleads for developing non-gradient algorithms.

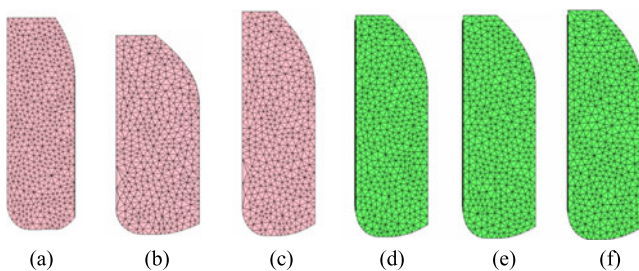


Fig. 12 Preforms of the gear: **a** Initial; **b** BFGS; **c** MAES; **d** SAO-SF; **e** SAO-MMF; **f** SAO-MEI

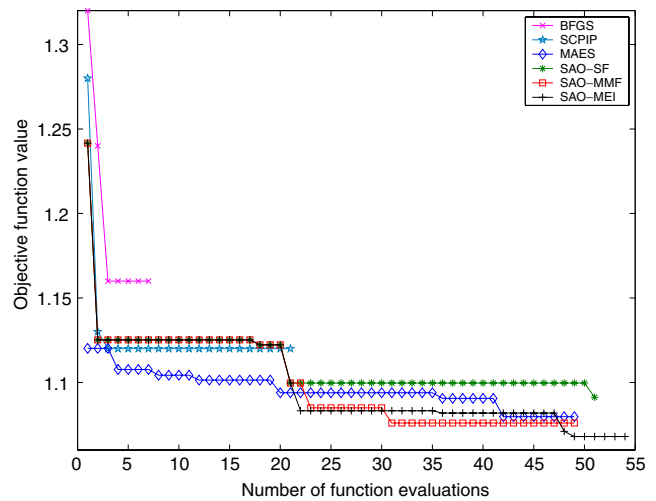


Fig. 13 Convergence of the algorithms for optimizing the gear

Metamodel Assisted Evolutionary Strategy (MAES) Being an evolutionary strategy, MAES is a global algorithm and use can be made of parallel computations. However, instead of running all 10 FEM calculations in this case, only the best 20% = 2 simulations are really performed. Hence, the possibility for parallel computing is not used to full extent. However, in return, the convergence plots in Figs. 9 and 13 show that the objective function improves quickly and greatly, the latter due to being a global algorithm.

MAES convincingly removed the folding defect from the spindle and reduced the forming energy and folding potential of the gear by 9.7% and 7.6%, respectively. It outperforms the iterative algorithms and shows that combining an evolutionary strategy with metamodeling techniques can overcome the disadvantage that genetic and evolutionary algorithms need many function evaluations in return for providing a global optimum and/or significant improvement.

Sequential Approximate Optimization (SAO) SAO-SF, SAO-MMF and SAO-MEI all started with 20 FEM calculations generated by the space-filling Latin Hypercube Design introduced in Section 2.2. This explains the coinciding convergence behaviour of the three variants of SAO during the first 20 FEM calculations in Figs. 9 and 13. Subsequently, the different sequential improvement strategies further improved the results in a number of batches. Using SAO-SF each new batch consisted of 10 new FEM calculations. The optimum of the metamodel of the previous batch was included each time. SAO-MMF and SAO-MEI themselves determine how many new simulations are required in a next batch (this depends on the number of minima found for the merit function and the number of maxima found for the expected improvement function, respectively).

For both forging cases, the convergence plots in Figs. 9 and 13 show very slow convergence behaviour for SAO-SF,

which simply adds new DOE points in a space-filling way. The final objective function value improvement is generally better than that obtained by the local BFGS and SCIP algorithms, but worse than MAES. Within 50 FEM calculations, SAO-SF did not manage to remove the folding defect for the spindle.

In contradiction to SAO-SF, the SAO-MMF and SAO-MEI algorithms exploit all information available from previous iterations (mean and standard deviations) and perform much better. SAO-MMF and SAO-MEI did solve the folding problem convincingly and they improved objective function values for the gear by about 10%, see Table 2. SAO-MMF and SAO-MEI even performed better than the MAES algorithm in both forging cases. For both the spindle and the gear, SAO-MEI performed slightly better than SAO-MMF, although the difference is small.

An advantage of SAO-MMF over SAO-MEI is that it is easier to optimize the merit function than the expected improvement function, as clearly seen in Fig. 3. The somewhat arbitrary selection of the weight factor w for SAO-MMF is seen as a disadvantage compared to SAO-MEI where arbitrary assumptions are not needed.

A final comparison Concluding this comparison, it can be stated that applying any of the optimization algorithms yielded better results than the initial situations. For the spindle, it was shown that the folding defect can be solved using optimization. For the gear, both the folding susceptibility and the energy consumption could be decreased by about 10% with respect to the initial forging process.

It has been found that global algorithms perform better than local (or quasi-local) algorithms. MAES and the three variants of the SAO algorithm generally yield superior results compared to the BFGS and SCIP algorithms. The local algorithms are less capable of solving the folding defect for the spindle, whereas most global algorithms do solve this forging defect. The exception is SAO-SF, which shows slow convergence behaviour, and less improvement than MAES, SAO-MMF and SAO-MEI.

In the end, it can be concluded that MAES, SAO-MMF and SAO-MEI all are very good algorithms for the optimization of forging processes. They have been shown to eliminate the folding defect for the spindle and have reduced both the folding susceptibility of the gear and the energy consumption needed for forging this part by approximately 10%. The fact that the difference between MAES, SAO-MMF and SAO-MEI is small is demonstrated by Tables 1 and 2: for both the spindle and the gear the optimal design variable settings are approximately the same for all three algorithms. This is visualised by the Figs. 8 and 12: the optimal preform shapes are very similar for MAES, SAO-MMF and SAO-MEI. Comparing the performance of these three algorithms to each other, SAO-MEI proves to be the best

algorithm, followed closely by SAO-MMF. Both SAO variants performed slightly better than MAES in both forging cases.

4 Conclusions

In this paper, the performance of Sequential Approximate Optimization (SAO) algorithms for optimizing forging processes using time-consuming Finite Element simulations is considered. Response Surface Methodology and Kriging are incorporated as metamodeling techniques. The best metamodels for objective function and constraints are included in the optimization model and optimized using a multistart Sequential Quadratic Programming (SQP) algorithm. SAO allows for sequential improvement if the acquired optimum is not accurate enough.

Three variants of the SAO algorithm have been investigated. They differ by the sequential improvement strategies used. The first variant puts in new DOE points in a space-filling way (SAO-SF). The second and third variants exploit all information already obtained during previous iterations. New DOE points are selected based on Minimising a Merit Function (SAO-MMF) and Maximising Expected Improvement (SAO-MEI).

These three variants of the SAO algorithm have been compared to each other and to other optimization algorithms by application to two forging processes: a spindle and a gear. The other algorithms taken into account are two iterative algorithms (BFGS and SCIP) and a Metamodel Assisted Evolutionary Strategy (MAES).

It is concluded that it is essential for sequential approximate optimization algorithms to implement a sequential improvement strategy that uses as much information obtained during previous iterations as possible. SAO-MEI and SAO-MMF have performed much better than the SAO-SF algorithm.

If an efficient sequential improvement strategy is used, SAO provides a very good algorithm to optimize forging processes using time-consuming FEM simulations: it outperforms the iterative algorithms which get stuck in a local optimum. Only the MAES algorithm stays close.

The potential of the proposed SAO algorithm has been demonstrated by two forging cases: it solved a folding defect for the spindle and decreased both the energy consumption and folding susceptibility for the gear with about 10%.

Finally, this paper shows the capacity of metamodel algorithms to solve actual 3D forging optimization problems within a limited and quite reasonable number of expensive FEM simulations. Sound results, if not globally optimal, are provided, which can offer a great aid in the design of complex forging sequences.

Acknowledgments This work has been conducted partly within the framework of project MC1.03162, Optimization of Forming Processes, which is part of the Strategic Research Programme of the Materials Innovation Institute (M2i) in The Netherlands and also partly within the framework of the OPTIMAS project (Optimization of Material Forming Processes), a project of the French RNTL network for Software Technologies. We would also like to thank the Institute for Mechanics Processes And Control Twente (IMPACT) for financially supporting the first author's internship at CEMEF, Ecole des Mines de Paris. The contributions of Michael Emmerich from the Dortmund University (Germany) within the 'APOMAT' COST 526 project, M2i and its industrial partners and of the Ascoforge Safe company are also gratefully acknowledged.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Bäck T, Schwefel HP (1993) An overview of evolutionary algorithms for parameters optimization. *Evol Comput* 1:1–23
- Beyer HG, Schwefel HP (2002) Evolution strategies: a comprehensive introduction. *Natural Computing* 1:3–52
- Bonte MHA (2007) Optimisation strategies for metal forming processes. PhD thesis, University of Twente, Enschede, The Netherlands, ISBN 978-90-365-2523-7
- Bonte MHA, van den Boogaard AH, Huétink J (2007) A metamodel based optimization algorithm for metal forming processes. In: Banabic D (ed) *Advanced methods in material forming*. Springer Verlag, Heidelberg, pp 55–72
- Bonte MHA, van den Boogaard AH, Huétink J (2008) An optimization strategy for industrial metal forming processes: modelling, screening and solving of optimization problems in metal forming. *Struct Multidiscipl Optim* 35(6):571–586
- Broyden CG (1970) The convergence of a class of double-rank minimization algorithms. *J Inst Math Appl* 6:222–231
- Bruyneel M, Duysinx P, Fleury C (2002) A family of MMA approximations for structural optimization. *Struct Multidiscipl Optim* 24:263–276
- Büche D (2004) Multi-objective evolutionary optimization of gas turbine components. PhD thesis, ETH Zürich, Zürich, Switzerland
- Büche D, Schraudolph N, Koumoutsakos P (2005) Accelerating evolutionary algorithms with gaussian process fitness function models. *IEEE Trans Syst Man Cybern* 35(2):183–194
- Castro CF, António CAC, Sousa LC (2004) Optimization of shape and process parameters in metal forging using genetic algorithms. *J Mater Process Technol* 146:356–364
- Emmerich M, Giotis A, Özdemir M, Bäck T, Giannakoglou K (2002) Metamodel-assisted evolution strategies. In: *Proceedings of the international conference on parallel problem solving from nature*, 07–11 September 2002, pp 361–370
- Fourment L, Chenot JL (1996) Optimal design for non steady-state metal forming processes—I shape optimization method. *Int J Numer Methods Eng* 39:33–50
- Fourment L, Balan T, Chenot JL (1996) Optimal design for non steady-state metal forming processes—II application of shape optimization in forging. *Int J Numer Methods Eng* 39:51–66
- Fourment L, Vieilledent D, Chenot JL (1998) Shape optimization of the axisymmetric preform tool in forging using a direct differentiation method. *Int J Form Process* 1:399–423
- Fourment L, Do TT, Habbal A, Bouzaiane M (2005) Gradient, non-gradient and hybrid algorithms for optimizing 2D and 3D forging sequences. In: *Proceedings of ESAFORM*. Cluj-Napoca, Romania
- Habbal A, Fourment L, Do TT (2008) Optimization globale par un algorithme hybride. Application en forgeage. *Rev Eur Mec Numer* 17:303–322 (in French)
- Jansson T, Nilsson L, Redhe M (2003) Using surrogate models and response surfaces in structural optimization—with application to crashworthiness design and sheet metal forming. *Struct Multidiscipl Optim* 25(2):129–140
- Jansson T, Andersson A, Nilsson L (2005) Optimization of draw-in for an automotive sheet metal part—an evaluation using surrogate models and response surfaces. *J Mater Process Technol* 159:426–234
- Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J Glob Optim* 21:345–383
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13:455–492
- Kleijnen JPC, van Beers WCM (2008) Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European J Oper Res* 186:1099–1113
- Kleinermann JP, Ponthot JP (2003) Parameter identification and shape/process optimization in metal forming simulation. *J Mater Process Technol* 139(1–3):521–526
- Koehler JR, Owen AB (1996) Computer experiments. In: Ghosh S, Rao CR (eds) *Handbook of statistics*. Elsevier Science, New York, pp 261–308
- Laroussi M, Fourment L (2004) The adjoint state method for sensitivity analysis of non-steady problems. Application to 3D forging. *Int J Form Process* 7(1–2):35–64
- Lin Z, Juchen X, Xinyun W, Guoan H (2003) Optimization of die profile for improving die life in the hot extrusion process. *J Mater Process Technol* 142(3):659–664
- Lophaven SN, Nielsen HB, Søndergaard J (2002a) Aspects of the MATLAB Toolbox DACE. Technical Report IMM-REP-2002-13, Technical University of Denmark—Department of Informatics and Mathematical Modelling, Lyngby, Denmark
- Lophaven SN, Nielsen HB, Søndergaard J (2002b) DACE—A MATLAB Kriging Toolbox. Technical Report IMM-TR-2002-12, Technical University of Denmark—Department of Informatics and Mathematical Modelling, Lyngby, Denmark
- Martin JD, Simpson TW (2005) Use of Kriging models to approximate deterministic computer models. *AIAA J* 43:853–863
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21:239–245
- Myers RH, Montgomery DC (2002) *Response surface methodology: process and product optimization using designed experiments*, 2nd edn. Wiley, New York
- Naceur H, Guo Y, Batoz J, Knopf-Lenoir C (2001) Optimization of drawbead restraining forces and drawbead design in sheet metal forming process. *Int J Mech Sci* 43(10):2407–2434
- Naceur H, Ben-Elechi S, Knopf-Lenoir C, Batoz JL (2004) Response surface methodology for the design of sheet metal forming parameters to control springback effects using the inverse approach. In: *Proceedings of NUMIFORM*. Columbus, OH
- Nielsen HB (2002) DACE, A MATLAB Kriging toolbox. <http://www.imm.dtu.dk/~hbn/dace/>. Accessed 26 July 2010
- Sacks J, Schiller SB, Welch WJ (1989a) Design for computer experiments. *Technometrics* 31:41–47
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989b) Design and analysis of computer experiments. *Stat Sci* 4:409–423
- Sakata S, Ashida F, Zako M (2007) On applying Kriging-based approximate optimization to inaccurate data. *Comput Methods Appl Mech Eng* 196:2055–2069

- Santner TJ, Williams BJ, Notz WI (2003) *The design and analysis of computer experiments*. Springer-Verlag, New York
- Sasena MJ, Papalambros P, Goovaerts P (2002) Exploration of meta-modeling sampling criteria for constrained global optimization. *Eng Optim* 34:263–278
- Schenk O, Hillmann M (2004) Optimal design of metal forming die surfaces with evolution strategies. *Comput Struct* 82:1695–1705
- Schonlau M (1997) *Computer experiments and global optimization*. PhD thesis, University of Waterloo, Waterloo ON, Canada
- Schwefel HP (1995) *Evolution and optimum seeking*, 6th edn. Wiley, New York
- Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. *Int J Numer Methods Eng* 24:359–373
- Svanberg K (1995) A globally convergent version of mma without line search. In: Olhoff N, Rozvany GIN (eds) *First world congress of structural and multidisciplinary optimization*, pp 9–16
- Torczon V, Trosset MW (1998) Using approximations to accelerate engineering design optimization. ICASE 98-33, NASA, Hampton VI, USA
- Vielledent D, Fourment L (2001) Shape optimization of axisymmetric preform tools in forging using a direct differentiation method. *Int J Numer Methods Eng* 52:1301–1321
- Zhao G, Wright E, Grandhi RV (1997) Preform design in metal forming using an optimization method. *Int J Numer Methods Eng* 40:1213–1230
- Zillober C (2001) Global convergence of a nonlinear programming method using convex approximations. *Numer Algorithms* 27(3): 256–289