PARAMETER OPTIMISATION OF STRESS-STRAIN CONSTITUTIVE EQUATIONS USING GENETIC ALGORITHMS

Y. Y. Yang (y.yang@shef.ac.uk), M. Mahfouf, and D. A. Linkens

IMMPETUS
Department of Automatic Control and Systems Engineering
University of Sheffield
Mappin St., Sheffield S1 3JD, UK

Abstract: The accuracy of numerical simulations and many other material design calculations (such as the rolling force, rolling torque, etc.) depends on the description of strain-stress relationship of the deformed materials. One common method of describing the strain-stress relationship is via constitutive equations. The unknown parameters in the constitutive equations are typically fitted by experimental data obtained via plane strain compression (PSC) using different optimisation techniques. Due to the highly nonlinear behaviour of the constitutive equations and the noise included in the stress-strain data, determination of the unknown parameters is not an easy task. In this paper, genetic algorithms are exploited to determine the optimal parameters for the constitutive equations and available PSC data. The original PSC data were processed to generate the required stress-strain data to determine the parameters for the selected constitutive equations. Data pre-processing is critical to remove the process noise contained in the original PSC data, leading to improved stress-strain data for the parameters optimisation. Several genetic optimisation schemes have been investigated, with different coding schemes and different genetic operators for selection, crossover and mutation. It was found that the real-value coded genetic algorithms converge much faster and are more efficient since there is no need for chromosome encoding and decoding.

Keywords: Genetic algorithms, parameter optimisation, aluminium alloy, flow stress, constitutive equations.

1. Introduction

Stress-strain relationships are important in many numerical simulations of materials processing and other design calculations (such as mill rolling force and rolling torque). Strain hardening functions relating the yield stress to the temperature, strain and strain rate are commonly used in finite element (FE) models [1]. How to obtain an accurate strain-stress relationship becomes critical for the correct calculation of the FE model. There has been much research into determining the stress-strain relationships by using plane strain compression (PSC) experiments, physical knowledge and some form of regression techniques, leading to the so-called constitutive equations [2, 3]. The PSC experimental data need to be interpreted adequately in order to deliver the data required for fixing the constitutive equations, typically the true stress, strain rate and stress data from the original load displacement PSC data. Much research has been conducted on this aspect, with correction for sample geometry, friction behaviour, isothermal deformation, etc [4, 5]. Also, optimization techniques have been used in determining the unknown parameters for the constitutive equations [6]. Neural networks have also been used for stress-strain modelling [7-9], including internal state variable models considering the dislocation density and other internal states as extra inputs [10, 11]. However, the accurate modelling of the stress-strain relationship demands a large number of PSC tests, because the existing knowledge of stress-strain relationship is not effectively used. Due to the highly nonlinear behaviour of the constitutive equations and the significant noise included in the stress-strain data, determination of the unknown parameters is not an easy task. Many different techniques have been developed for determining these parameters, such as simplex optimisation, Monte-Carlo simulation, etc. Unfortunately, such methods can often suffer from the “local minimum” problem due to the nature and size of the search space. This motivates us to search for the optimal parameters for the constitutive equations using genetic algorithms (GA) [12]. The advantage of GA is that they do not require any gradient information, neither the continuity assumption in searching for the best parameters, since the search is guided by the natural fitness survival mechanism.

In this paper, parameter optimisation was conducted based on GA search for the selected constitutive equations, with the stress-strain data obtained via a series of PSC tests on AA5052 grade aluminium. The original PSC data (containing the load-displacement data) was processed with the Terrapin programme and other data processing to produce the equivalent stress-equivalent strain data required for fixing the parameters of the constitutive equations. The geometric parameters of the PSC samples and the friction conditions have been considered in the data processing. The paper is organised in the following way: a brief description of the PSC tests and the processing of the stress-strain data are given in Section 2; basic concepts of GA-based optimisation and its application in parameter estimation are given in Section 3, illustrated with the determination of parameters for constitutive equations with different coding
schemes and genetic operators for parameter optimisation being introduced; implementation of the GA-based optimisation for determining the parameters of the constitutive equations are given in Section 4, together with analysis and discussions; finally conclusions are outlined in Section 5.

2. PSC Tests and Data Processing

In order to determine the parameters of the constitutive equations, a series of PSC tests with different temperatures, strains, and strain rates, have been conducted. Specimens (taken from aluminium alloy AA5052) were prepared, with a nominal width of 50mm, and length of 100mm. Three different thicknesses i.e., 10mm, 5mm, and 2.5mm, have been prepared for the PSC specimens to investigate the geometrical factors on the flow stress behaviour. An aluminium specimen was first heated in the heating furnace until it reached the specified temperature, then it was moved into the PSC apparatus that conducts the pressing with the prescribed nominal strain rate. A graphite lubricant was applied to the specimens to reduce the friction during the PSC test. Tests were carried out at the University of Sheffield using a 500 kN servo-hydraulic machine with computer control, at constant equivalent strain rates of 0.5, 3 and 20 s\(^{-1}\) to a total equivalent strains \(\varepsilon = 1.5\). The nominal testing temperatures were 300 and 400° C. Load, displacement and temperature were recorded as a function of time during the tests. The load and displacement data were converted from raw data to equivalent stress–equivalent strain curves and also pressure–equivalent strain using an analysis program, with some necessary corrections for the geometry change of the specimen in three dimensions [4]. Instantaneous strain rate was first computed from the original PSC data, and was then processed to obtain a smoothed strain rate to avoid the oscillation noises introduced by the combined effect of vibration and differentiation. Instantaneous temperature was computed using a finite difference thermal model of the plane strain compression test developed by Foster [13] because it was found that thermocouples did not always remain in the centre of the deformation of work and other electromagnetic disturbances can affect the temperature measurement. A total of 18 SPC tests have been conducted for investigating the stress-strain relationship of the AA5052. To reduce the redundant PSC data for the small strain rate test, a maximum of 200 data points was used with uniform sampling of the overall stress-strain curve. Fig. 1 shows the equivalent stress-strain data, where the dynamic specimen temperatures and instant strain rates were used. It can be seen that the PSC training data contain significant noise, even after careful data processing, especially for the equivalent strain rate.

![Fig. 1 Overview of the AA5052 PSC data](image)

Various researchers showed that the selection of the function describing stress–strain curves is crucial for accuracy of the constitutive equations. Several types of equations used as constitutive laws are not suitable of describing stress-strain behaviour of the aforementioned PSC tests. After carefully investigation, the constitutive functions proposed by Shi et al are adopted for the flow stress calculation [3]:

\[
\begin{align*}
Z & = \dot{\varepsilon} e^{Q_{\text{def}}/(RT)} \\
\sigma_0 & = \alpha_0 \sinh^{-1}(Z/A_0)^{1/\alpha_0}; \\
\sigma_r & = \alpha_r \sinh^{-1}(Z/A_r)^{1/\alpha_r}; \\
\varepsilon_r & = C_1 + (C_2 \sigma_r)^3 \\
\sigma & = \sigma_0 + (\sigma_r - \sigma_0)(1 - e^{-h/\varepsilon_r})^{0.5}
\end{align*}
\]

where \(R\) is the gas constant, \(Q_{\text{def}}\) is the activation energy for deformation, \(T\) is the deformation temperature in \(^\circ\)K, \(\varepsilon\) is the equivalent strain, \(\dot{\varepsilon}\) is the strain rate, \(Z\) is the Zener-Hollomon parameter, \(\varepsilon_r\) is the relaxation or transient strain.
constant, $\sigma_0$ is the initial flow stress, $\sigma_s$ is the saturation or steady state flow stress, and $\sigma$ is the equivalent flow stress. $\alpha_0$, $\alpha_s$, $A_1$, $A_2$, $n_0$, $n_s$, $C_1$, $C_2$ are model parameters depending on the alloy contents and deformation conditions. To simplify the discussion, a model parameter vector $\beta$ is introduced and equation (1) can be represented by:

$$y(k) = f(\beta, x(k))$$

(2)

where $y(k)$ is the output prediction (flow stress) of the constitutive model. $\beta = [Q, \alpha_0, \alpha_s, A_1, A_2, n_0, n_s, C_1, C_2]$ is the model parameter vector, and $x = [\varepsilon, T, \varepsilon]$ is the input vector consisting of strain, temperature and strain rate.

3. Genetic Algorithms for Parameter Optimisation

Genetic algorithms (GA) were proposed by Holland and his team during the late 1960s and in the early 1970s [14], and research interests in GA have been growing rapidly since then. Essentially, GA is a kind of search algorithm based on the mechanics of natural selection and natural genetics [12, 14]. GA combines “survival of the fittest” among string populations with a structured yet randomised information exchange to form a search algorithm. At each generation, a new set of populations is created by the process of selecting individuals according to their fitness level defined in the problem domain and then the selected populations breed using operators such as crossover and mutation borrowed from natural genetics. This process leads to the evolution of the populations that are better suited than their parent generation, and eventually leads to the optimal solution. The GA technique has been successfully applied to many optimisation problems, e.g., game playing, scheduling, transportation problems, optimal control, adaptive control, neural network structure, machine learning, etc [15].

GA works on the chromosomes of the population, not on the decision variables. Individuals in the population are coded as strings, known as chromosomes, so that genotypes (chromosome values) are uniquely mapped onto the decision variables (known as phenotypes). There are many different schemes for chromosome encoding, such as binary coding, real-value coding, tree coding, etc. Examination of a chromosome string in isolation usually yields no information about the problem to be solved. A chromosome needs to be decoded into its phenotypic values and the objective function (fitness) of the chromosome can then be evaluated. The search process in GA operates on the encoding of the decision variables, rather than the decision variables themselves, except in cases where the genotypes are identical to the phenotypes, as in the case of a real-value coding. Fig. 2 shows how GA works for a typical optimisation problem. It starts by generating chromosomes representing the initial population with a specific number of individuals. The fitness values of the individuals are evaluated based on the objective function of the optimisation problem. From here on, the genetic evolution proceeds from generation to generation. In producing a new generation, the parents needed for breeding a new child are first selected according to their fitness values and put into the mating pool. The selection will ensure that high-fitted individuals have a high probability to be selected. Recombination takes place to produce children from the selected parents, often via some kind of crossover operations. The generated child can go through mutations, with some bits in the chromosome being mutated. The purpose of mutation, which is generally a background operator with small probability, is to prevent premature convergence of the population. The GA sequence ends when either the maximal generations have been produced or other predefined termination conditions are satisfied. The average performance of individuals in the population is expected to increase during the evolution of the populations, as good individuals are preserved and bred with one another and the less-fit individuals die out.
**Population Evolution:** \( \text{gen} = \text{gen} + 1 \) 
Select \( P(\text{gen}) \) from \( P(\text{gen}-1) \) Based on Fitness \( (\text{gen}-1) \) 
Recombination: \( P(\text{gen}) \) Crossover, \( P(\text{gen}) \) Mutation 
Fitness Evaluation: \( \text{Fitness}(\text{gen}) = \text{Evaluate} (P(\text{gen})) \) 
Termination? 
**End** 

**Start GA**s: Initialisation 
\( \text{gen} = 0 \), Generate Ppopulation \( P(\text{gen}) \) 
\( \text{Fitness}(\text{gen}) = \text{Evaluate} (P(\text{gen})) \) 

**Basic GA search for Epoch Generations** 
Selection, Crossover, Mutation & Insertion 
End Iteration? 
**End** 

**Start Iterative** GA: Epoch = 300 
\( \text{gen} = 0 \), Initial Population \( P(\text{gen}) \) 
Select Fitness (Performance) Index 
Set selection, crossover, mutation operators 
Select crossover, mutation, insertion rates 
Set other GA parameters & options 
Basic GA search for Epoch Generations 
Selection, Crossover, Mutation & Insertion 
End Iteration? 
**End** 

**Fig. 2 Block diagram of a simple GA sequence** 
**Fig. 4 Flowchart of the iterative GA search procedure**

In the current optimisation problem, the decision variable is the unknown parameter vector \( \mathbf{\beta} \), which consists of 9 parameters. Different encoding schemes have been adopted for generating the chromosomes, and Fig. 3 shows the chromosome structures of binary encoding [12] and real-value coding.

(a) Binary coded chromosomes

```
0110010...100 1100101...101 0010101...001 1110111...111 ...
```

Gene #1 -> Q  Gene #2 -> \( \alpha_1 \)  Gene #3 -> \( \alpha_s \)  ... ... ... ... Gene #9 -> \( C_2 \)

(b) Real-value coded chromosomes

```
<table>
<thead>
<tr>
<th></th>
<th>Q</th>
<th>( \alpha_1 )</th>
<th>( \alpha_s )</th>
<th>( A_1 )</th>
<th>( A_s )</th>
<th>( n_0 )</th>
<th>( n_s )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.65 ( \times 10^3 )</td>
<td>25.9</td>
<td>45.45</td>
<td>1.19 ( \times 10^{11} )</td>
<td>1.90 ( \times 10^{11} )</td>
<td>6.7</td>
<td>6.0</td>
<td>0.074</td>
<td>2.74 ( \times 10^{-5} )</td>
</tr>
</tbody>
</table>
```

**Fig. 3 Chromosome structures**

The resolution of a binary code is limited by the number of the binary digits used to encode the genotype, and can be determined by:

\[
dx = (x_{\text{max}} - x_{\text{min}}) / 2^n
\]

(3)

where \( x_{\text{min}} \) and \( x_{\text{max}} \) are the minimal and maximal values of the decision variable \( x \), \( n \) is the number of digits of the binary code for \( x \), and \( dx \) is the resolution of encoding. \( n \) can be decided by the requirement of encoding resolution \( dx \).

For the parameters of the constitutive equations, an encoding resolution of \( dx = 0.0001 \) is desirable for normalised parameters with a range of \([1, 100]\), which leads to a 20-bit binary encoding for a model parameter. The resulting chromosomes using binary coding will have a length of 180 bits (20 bits for each of the 9 model parameters). In a real-value coding a parameter is naturally represented by a real-variable. The resulting chromosomes are much smaller (9 real-value variables) compared to the binary coding, and the expression power of the real-value code is better than the binary coding. Another advantage of real-value coding is that no encoding and decoding are required for the model parameter vector \( \mathbf{\beta} \). From the nature of the parameter optimisation problem for the constitutive equations, real-value coding appears to be more suitable.
4. Implementation of GA Search and Results

For the optimisation of the constitutive equations given in (1-2), the minimal and maximal values of the unknown parameters for the search are listed in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( x_{\text{min}} )</th>
<th>( x_{\text{max}} )</th>
<th>Parameter</th>
<th>( x_{\text{min}} )</th>
<th>( x_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q )</td>
<td>10000</td>
<td>1000000</td>
<td>( n0, ns )</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>( \alpha_0, \alpha_s )</td>
<td>1</td>
<td>100</td>
<td>( C_1 )</td>
<td>0.01</td>
<td>1</td>
</tr>
<tr>
<td>( A_0, A_s )</td>
<td>( 1 \times 10^{10} )</td>
<td>( 1 \times 10^{12} )</td>
<td>( C_2 )</td>
<td>( 1 \times 10^{-6} )</td>
<td>( 1 \times 10^{-4} )</td>
</tr>
</tbody>
</table>

The aim of the GA optimisation is to determine the unknown model parameters \( \beta \) such that the errors between the predictions of the constitutive equations and the real measurements are minimised. The root mean square error (RMSE) is used here as the objective function for optimisation:

\[
J(\beta) = \frac{\sqrt{\sum_{k=1}^{N} (y^d(k) - f(\beta, x(k)))^2}}{N}
\]

where \( \{x(k), y^d(k)\}, \ k = 1,2,\ldots,N \) are the available stress-strain data.

Three different GA coding schemes, binary coding, gray-coding and real-value coding, have been used to solve the parameter optimisation. Gray-coding has some advantage over the standard binary coding, with a small perturbation caused by simple mutations. An iterative GA procedure is designed to guide the search processing, with the overall search process divided into a number of fixed generations. At the beginning of a continued new GA search, the option is given to select new selection schemes as well as the crossover and mutation operators. The advantage of the iterative GA procedure is its flexibility and the ability to change the critical GA parameters based on the stage of the searching. Also, the termination of the GA can be governed by monitoring the iterative searching process. The iterative GA procedure is able to give better solutions due to the interaction between GA search and human judgement. Fig. 4 shows the flowchart of the iterative GA search procedure. For both the binary coding and the real-value coding, different genetic searches have been tested to find the suitable GA parameters, such as the selection and crossover methods, the crossover and mutation rates, the population sizes, the fitness scaling factors, the termination criteria, the variation of GA parameters and options during the iterative search procedures, etc. The GA parameters were then set up and extensive GA searches were carried-out to find the optimal decision variables \( \beta^* \). In the following, typical GA results obtained by the iterative GA optimisation based on gray-coding and real-value coding will be presented, with the initial population size of 50, and an intermediate generation of 300.

For gray coding GA, stochastic universal sampling based on ranking fitness evaluation has been used for reproduction, and the fitness values are scaled such that the fitness value for the minimal objective function equals twice the fitness value for the mean objective function. A single point crossover and a double point crossover have been used alternatively with crossover rate = 0.7, followed up by a random mutation with a mutation rate of 0.035 for breeding the child generation. The GA searching profile is given in Fig. 5a. The final model prediction error is RMSE = 15.02 MPa, and model parameter vector fixed at \( \beta^* = [17.513467, 26.388574, 56.851392, 100.336461, 36.625307, 137.019889, 3.605400, 44.891117, 56.656746] \).
For the real-value coding, the selection algorithm and fitness calculation are the same as for binary coding. For convenience of comparisons, the population size and total number of generation are kept the same as in binary coding. Special crossover and mutation algorithms need to be designed according to the nature of real-value coding. A discrete recombination scheme is used for crossover, which exchanges variable values between the two parents to produce two children, with a crossover rate of 1. For each variable the parent who contributes its variable to the offspring is chosen randomly with equal opportunity. A real-value mutation algorithm is then used for mutation, with an equivalent (binary) mutation rate of 0.005. The GA searching profile is given Fig. 5b. The final model prediction error is RMSE = 14.42 MPa, and model parameter vector fixed at $\mathbf{b}^* = [16.413942, 26.090100, 61.028597, 1.507852, 9.270131, 100.000000, 3.550106, 48.230574, 100.000000]$. The evolution of some model parameters during the real-value code GA searching is shown in Fig. 6a, with most parameters stabilised after approximately 400 generations. Fig. 6b shows the prediction results of the developed constitutive equations for the AA5052 PSC testing data which have not been used for GA optimisation. The results are satisfactory, considering the noise contained in the PSC data.

Both the gray coding GA and real-value coding GA achieve similar modelling performance (with the latter being a slightly better). Comparing the binary code and the real-value code GA, it was found that the latter is much faster in finding the decision variables: real-value coding GA converged after 300 generations while binary coding GA took approximately 1000 generations. The mean objective function in real-value coded GA shows a more consistent behaviour and converged gradually to its best objective value (see in Fig. 5b), while the mean objective function in the binary code GA shows significant oscillations and does not converge (see Fig. 5a). Furthermore, in real-value code GA, there is no need for chromosome encoding and decoding, which saves a considerable computation time, leading to a much faster search per-generation than that of the binary-code counterpart. To visualize the behaviour of the developed constitutive equations, the corresponding model response surfaces are presented in Fig. 7 where the remaining model inputs take the mean nominal. The constitutive equations have correctly caught the trend of the stress variation against the deformation temperature and the strain rate.
5. Conclusions

This study showed that GA was effective in determining the unknown parameters of the specified constitutive equations. Both the gray-coding and real-value coding GAs were successful in finding suitable unknown parameters which gave similar prediction performance, however, the real-value coding GA was more efficient for the current optimisation problem. The genotypes are identical to the phenotypes in the real-value coded GA scheme, hence there is no need for chromosomes’ encoding and decoding, which are often computationally demanding. Also, the convergence speed in finding the optimal solution was faster in the real-value coded GA. The resulting constitutive equations have been verified against unseen PSC data and the prediction of the flow stress is satisfactory, with prediction error at the same level as for those of the “training” PSC data. The iterative GA procedure proposed in this paper is very flexible and helps to tackle complicated optimisation by intelligent adjustment of the critical GA parameters according to the search behaviour, although it needs additional expert’s knowledge. Some kind of automated GA parameter adjusting might be desirable, such as the rule guided approach. This can be a topic for further research.

The parameter optimisation problem investigated here is essentially data driven after the function structure has been fixed. As is well known, the quality of the optimisation results are affected by the quality of the PSC data. Although considerable effort has been devoted to data processing for better and more consistent PSC data, there is still considerable noise in the strain rate, as is obvious from Fig. 1. The extreme vibration of the strain rate is partially caused by the limitation of the servo-hydraulic PSC machine and is difficult to remove by computer signal processing. A new thermo-mechanical compression (TMC) machine has recently been commissioned. It is a purpose-built powerful machine (covering a wide range of constant true strain rates up to 300 s$^{-1}$) designed to simulate metal forming processes, and represents the most advanced machine of its type. So the effect of strain rate noise (vibration) should not be significant in future investigation.

ACKNOWLEDGEMENT

The authors wish to thank Dr B Kowalski and Mr. A Lacey for their valuable discussions and providing the PSC data. Financial support from EPSRC is also acknowledged under GR/L50198 and GR/R70514/01.

REFERENCES


