A novel evolutionary algorithm for determining unified creep damage constitutive equations

B. Li[a], J. Lin[b, *], X. Yao[c]

[a] *School of Mechanical Engineering, Southwest Jiaotong University, Chendu 610031, People's Republic of China*
[b] *School of Manufacturing and Mechanical Engineering, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK*
[c] *School of Computer Science, The University of Birmingham, Birmingham B15 2TT, UK*

Abstract

The determination of material constants within unified creep damage constitutive equations from experimental data can be formulated as a problem of finding the global minimum of a well defined objective function. However, such an objective function is usually complex, non-convex and non-differentiable. It is difficult to be optimised by classical gradient-based methods. In this paper, the difficulties in the optimisation are firstly identified. Two different objective functions are proposed, analysed and compared. Then three evolutionary programming algorithms are introduced to solve the global optimisation problem. The evolutionary algorithms are particularly good at dealing with problems which are complex, multi-modal and non-differentiable. The results of the study shows that the evolutionary algorithms are ideally suited to the problem. Computational results of using the algorithms to determine the material constants in a set of physically based creep damage constitutive equations from experimental data for an aluminium alloy are presented in the paper to show the effectiveness and efficiency of the three evolutionary algorithms.© 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Constitutive equations; Creep damage; Evolutionary algorithms; Global optimisation

1. Introduction

There is a common requirement to describe the creep behaviour of different materials using physically based unified creep damage constitutive equations. In parallel with this, constitutive equations used to describe the creep behaviour are becoming more complex as additional state

---

* Corresponding author. Tel.: +44-0121-414-3317; fax: +44-0121-414-3515.
E-mail address: j.lin@bham.ac.uk (J. Lin).

Nomenclature

ε	creep strain
ε^t	computed (theoretical) strain
ε^m	experimental (measured) strain
σ	stress (MPa)
t	time (h)
$A,\ B,\ h,\ H^*,\ K_c,\ D$	material constants
H	primary state variable
$\phi,\ \omega_2$	damage state variables
t^t	computed (theoretical) time (h)
t^m	experimental (measured) time (h)
$f(X),\ F(X)$	objective functions
d	the shortest distance from the computed to experimental data
$f_g(x)$	Gaussian density function
$f_t(x)$	Cauchy density function

variables are introduced to accurately describe the deformation and damage mechanisms [1–3]. The determination of the values of material constants in such constitutive equations is often problematic, even when high quality experimental creep data is available [3,4]. This difficulty has been discussed by many researchers [1,4,5] and optimisation techniques are normally used to determine the material constants on the basis of minimising the sum of the errors between experimental and computed data.

Lin and Hayhurst [6] developed an optimisation technique and successfully determined the material constants for the constitutive equations where stress can be explicitly expressed as a function of strain. However, for a set of unified creep damage or general viscoplastic constitutive equations, numerical integration is needed for solving the equation sets. It has been shown that the objective functions for the optimisation are highly non-linear [3–5]. Hence, traditional gradient descent methods tend to fail because it is very difficult to choose starting values for the constants to obtain the global minimum. Thus, extra care should be taken to choose starting values of the constants for optimisation. Kowalewski et al. [3] developed a three-step method to determine the starting values of the constants in a set of creep damage constitutive equations for final optimisation. Perrin and Hayhurst [5] developed an optimisation method to determine damage constitutive equations based on a technique of transformation of creep constitutive equations. However, it is still difficult to choose starting values of the constants. Zhou and Dunne [7] proposed a four-stage method to determine material constants for a particular set of the viscoplastic constitutive equations for superplastic alloys. The starting values of the material constants have to be chosen carefully at each stage of the optimisation, otherwise convergence to a local minimum is unavoidable and poor material constants would be obtained.

The techniques developed above are highly problem dependent and usually suitable for particular sets of constitutive equations. Significant knowledge of material science, mathematics and computational mechanics is required. Hence, they are not suitable for industrial users. In order to solve the problem, Lin and Yang [4] proposed a genetic algorithm (GA) based optimisation technique, which is suitable for both creep damage models and general unified viscoplastic constitutive equations.

This technique has generic features, overcomes the difficulty of choosing starting values for the constants, and provides a better chance to converge to the global minimum. However, the method used was based on the classical binary GA [8]. It is inefficient in solving continuous problems, such as those considered in this paper. Hence the computing resources required for solving the optimisation problems are significant. In this paper, three new algorithms based on evolutionary programming are introduced to solve the continuous optimisation problem. They were shown to be much more efficient than binary GAs for continuous problems [9].

This paper is organised as follows. First a set of unified creep damage constitutive equations for aluminium alloys are introduced in Section 2. In order to avoid the traditional problem of the inconsistency in experimental and computed lifetimes [3], two new objective functions are formulated for this type of optimisation and the behaviour of the objective functions is then analysed in Section 3. Three evolutionary algorithms for minimising the two objective functions are described in detail in Section 4. Finally, the computational results and conclusions are presented.

2. Physically based constitutive equations

Continuum damage mechanics (CDM) with two damage variables was justified and used by Kowalewski et al. [1,3] to model tertiary creep softening in an aluminium alloy caused by (i) grain boundary cavity nucleation and growth, and (ii) ageing of particular microstructures. Primary creep is also described by the model. The form of the constitutive equations proposed for uni-axial conditions is given by the following set of equations:

$$\frac{d\varepsilon}{dt} = \frac{A}{(1-\omega_2)^n} \sinh\left(\frac{B\sigma(1-H)}{1-\phi}\right), \tag{1}$$

$$\frac{dH}{dt} = \frac{h}{\sigma}\frac{d\varepsilon}{dt}\left(1 - \frac{H}{H^*}\right), \tag{2}$$

$$\frac{d\phi}{dt} = \frac{K_c}{3}(1-\phi)^4, \tag{3}$$

$$\frac{d\omega_2}{dt} = \frac{DA}{(1-\omega_2)^n}\sinh\left(\frac{B\sigma(1-H)}{1-\phi}\right), \tag{4}$$

where A, B, h, H^*, K_c and D are material constants, and n is given by

$$n = \frac{B\sigma(1-H)}{1-\phi}\coth\left(\frac{B\sigma(1-H)}{1-\phi}\right).$$

Eq. (1) describes the creep evolution for the aluminium alloy. The steady state creep is related to the constants A, B and H^*. Eq. (2) describes primary creep using variable H, which varies from 0 at the beginning of the creep process to H^*, where H^* is the saturation value of H at the end of primary period and subsequently maintains this value until failure.

The equation set contains two damage variables to model tertiary softening mechanisms. The first damage variable ϕ, which is described by Eq. (3), is defined from the physics of ageing to lie within the range 0–1 for mathematical convenience. The second damage variable ω_2, defined by Eq. (4), describes grain boundary creep constrained cavitation, the magnitude of which is strongly sensitive to alloy composition and to processing route. Creep constrained cavitation can either be nucleation or growth controlled. Here only nucleation control is considered and the failure criterion is $\omega_2 = \frac{1}{3}$ [3].

Material constants which appear in this model may be divided into three groups, i.e.,

 (i) constants h and H^* which describe primary creep;
 (ii) constants A and B which characterise secondary creep; and
(iii) constants K_c and D responsible for damage evolution and failure.

The six material constants need to be determined from experimental data so that the creep damage behaviour for the material can be accurately predicted using Eqs. (1)–(4).

3. Features of objective functions for optimisation

3.1. Formulation of objective functions

Optimisation techniques for determining the material constants arising in the constitutive equations are based on minimising the sum of the squares of errors between the computed and experimental data. The objective function is normally defined, for example [3], by the sum of the squares of the differences between the computational and experimental creep strains for the same time and stress levels. If the predicted lifetime is longer than the corresponding experimental lifetime, some of the experimental data cannot be involved in the error estimation. In order to compensate these errors, an extra correction term is introduced, which is detailed by Kowalewski et al. [3]. In this work, two methods are proposed to define the objective functions for the optimisation.

(a) *Shortest distance method*: In this method, the errors are defined by the shortest distance between computational and experimental data. In theory it is an ideal form of defining an objective function. But it is difficult to compute in practice. Here, a simple and computationally inexpensive method is introduced to calculate the shortest distance of an experimental data point $(t_{ij}^m, \varepsilon_{ij}^m)$, ith data point of the jth experimental creep curve, to the corresponding numerically integrated creep curve. This includes three steps: (i) identify two adjacent closest integrated points on the creep curve to the experimental data point; (ii) determine a straight line through the two integrated points; (iii) calculate the distance of the experimental data point to the straight line. The point determined on the computed creep curve is noted as $(t_{ij}^t, \varepsilon_{ij}^t)$. For creep constitutive equations a situation may occur in the optimisation, in which the shortest distances almost keep unchanged if the computed lifetime is much longer than the experimental lifetime. In order to increase the sensitivity, one extra term is introduced. Thus the objective function, $f(X)$, is defined by

$$f(X) = \sum_{j=1}^{n_1} \sum_{i=1}^{m_j} w_{ij} d_{ij}^2 + \sum_{j=1}^{n_1} W_j (t_{m_j j}^t - t_{m_j j}^m)^2, \tag{5}$$

where X $(X = (x_1, x_2, \ldots, x_s))$ represents the material constants and s is the number of constants to be optimised. n_1 is the number of experimental creep curves. m_j is the number of the experimental data for the jth creep curve. d_{ij} represents the shortest distance from the ith data point of the jth experimental creep curve to the corresponding computed creep curve. w_{ij} is the relative weight for the ith data point of the jth experimental creep curve. W_j is the relative weight for stress level j. $(t^t_{m_j} - t^m_{m_j})$ is the error between the computational and the experimental times for the same creep strain and the same stress level j. In order to increase the sensitivity of the objective function and keep the units consistent, the following transformation is introduced to define the shortest distance d_{ij}:

$$d^2_{ij} = \alpha(\varepsilon^t_{ij} - \varepsilon^m_{ij})^2 + \beta(t^t_{ij} - t^m_{ij})^2, \tag{6}$$

where, for the material under consideration, we choose $\alpha = 100$ and $\beta = 0.01$ (h^{-2}). $(t^t_{ij} - t^m_{ij})$ is the error in time and $(\varepsilon^t_{ij} - \varepsilon^m_{ij})$ is the error in strain. Eq. (5) is known as objective function I (OFI).

(b) *Errors in time only*: In order to avoid the correction term introduced in Eq. (5), another form of objective function is introduced and given below.

$$F(X) = \sum_{j=1}^{n_1} \sum_{i=1}^{m_j} w_{ij}(t^t_{ij} - t^m_{ij})^2, \tag{7}$$

where $(t^t_{ij} - t^m_{ij})$ is the time difference between the computational and experimental data for the same creep strain ε^m_{ij}. n_1, m_j and w_{ij} have the same meanings as defined before. Eq. (7) is called objective function II (OFII).

3.2. Monotonicity analysis of creep strain and objective functions

From Eqs. (1)–(4), it can be seen that the creep strain ε is a function of the seven variables A, B, h, H^*, K_c, D and t. For the sake of convenience, the strain ε is written as $\varepsilon = \varepsilon(A, B, h, H^*, K_c, D, t)$. As time t increases while keeping other variables constant, the absolute value of the creep strain ε increases. This means that the strain ε monotonically increases with time t [10–12]. The result can also be obtained from the condition $\mathrm{d}\varepsilon/\mathrm{d}t > 0$ [10]. Thus, the following characteristics of Eqs. (1)–(4) can be obtained: (i) ε monotonically increases with respect to t. (ii) The boundary conditions do not change. That is when $t = 0$, there are always $\varepsilon = 0$, $\omega_2 = 0$, $H = 0$ and $\phi = 0$. (iii) As one of the material constants changes and the others are constant, the creep strain curve must change. For example, the point of $\varepsilon(A, B, h, H^*, K_c, D, t) = \varepsilon(A + \Delta, B, h, H^*, K_c, D, t)$ does not exist except for $t = 0$. The two curves do not cross except for at the point $t = 0$.

When $t = 0$, Eq. (1) can be rewritten as:

$$\left.\frac{\mathrm{d}\varepsilon}{\mathrm{d}t}\right|_{t=0} = A \sinh(B\sigma). \tag{8}$$

In Eq. (8), as the material constant A increases, e.g., from A to $A + \Delta$, while keeping others constant, $\mathrm{d}\varepsilon/\mathrm{d}t|_{t=0}$ increases, i.e., $\mathrm{d}\varepsilon/\mathrm{d}t|_{t=0,A=A+\Delta} > \mathrm{d}\varepsilon/\mathrm{d}t|_{t=0,A=A}$. Thus at $t = \Delta t$, $\varepsilon(A + \Delta, B) > \varepsilon(A, B)$. This result can be extended to the whole equation set, that is, $\varepsilon(A + \Delta, B, h, H^*, K_c, D, t) > \varepsilon(A, B, h, H^*, K_c, D, t)$ for $t > 0$. This indicates that creep strain ε increases with the

Table 1
Material constants used for Figs. 1 and 2

A	B	h	H^*	K_c	D
4.04e-15	0.1126	2.95e4	0.1139	1.82e-4	2.75

Table 2
Values of derivatives calculated using the constants listed in Table 1 for OFI

$\partial f / \partial A$	$\partial f / \partial B$	$\partial f / \partial h$	$\partial f / \partial H^*$	$\partial f / \partial K_c$	$\partial f / \partial D$
8.23e15	5.97e3	1.21e-3	2.78e2	5.69e-6	9.54e-1

increment of A. This phenomenon is known as the monotonicity of the creep strain with respect to A. The same results can be obtained for the other material constants. The results can be summarised as $\varepsilon = \varepsilon(A^+, B^+, h^-, H^{*-}, K_c^+, D^+, t^+)$, where the superscript "+" indicates that creep strain ε increases as the value of the constants increase, i.e., ε monotonically increases with respect to the constants, and "−" represents that ε decreases as the constants increase, i.e., ε monotonically decreases with respect to the constants.

3.3. Characteristics of the objective functions

The creep experimental data reported by Kowalewski et al. [1] for an aluminium alloy at $150°C$ for four stress levels of 241.3, 250.0, 262.0 and 275.0 MPa are used in the optimisation and analysis. If five of the material constants listed in Table 1 are used and only one constant is allowed to vary, the behaviour of the two objective functions with the variation of individual constants are given in Fig. 1. The thin lines in the figure are calculated using OFI and the thick lines are computed with OFII. Fig. 1 shows that the two objective functions have similar behaviour. Only one minimum point is observed for each material constant, although there are plateau regions where $\partial f / \partial x_i \approx 0$. Fig. 2 shows the variation of OFI with the material constant B for different values of constant A. The values of the other material constants given in Table 1 are used for the computations. It can be seen that the curve shifts with the variation of the value A. Similar features of the OFII are observed. The minima occur within a very narrow region for all A values. As the values of B are greater than the values B_1, B_2 and B_3 for the corresponding three A values, the residuals remain almost constant. This indicates that the values of A and B are too big and the computed lifetimes of the creep curves tend to zero. Thus, the sum of the errors remains constant. Fig. 2 also indicates that different combinations of constants A and B give different values of minima and only one minimum point exists for each combination.

In multiple variable optimisations, the differentiation of an objective function respect to individual variables should have similar values. Otherwise, it is difficult to obtain a balanced result. Table 2 gives the values of derivatives for individual variables which are calculated using the constants listed
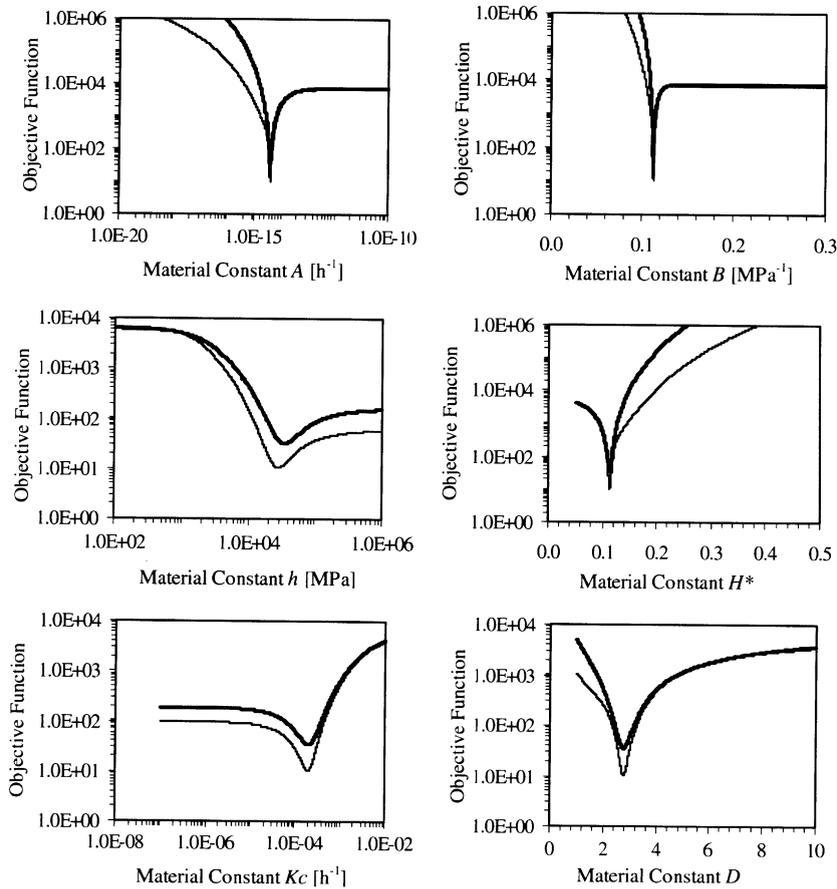
Fig. 1. Variation of objective functions with individual material constants. The thin lines are obtained using OFI and the thick lines are computed using OFII.

in Table 1. A huge variation is observed. For example, the largest value of the derivatives is 8.23e15 for $\partial f / \partial A$ while the lowest value is 5.69e-6 for $\partial f / \partial K_c$. This behaviour of the objective function is very "bad" and needs to be avoided. When the traditional optimisation methods are used to solve the problem, iterations are very often stopped at almost the starting points for the variables, which have low values of derivatives.

According to the above analysis, there are following three intrinsic difficulties for the optimisation.

(i) Variation of objective functions for some of the material constants, such as constants A, B, h and K_c (cf. Fig. 1) contains large plateau regions, that is $\partial f / \partial x_i \approx 0$. If the starting points are given within these regions, a gradient-based optimisation system will treat those points to be minima. Hence the optimisation fails.

(ii) The difference among the derivatives for the material constants is too large for traditional optimisation algorithms to work well. The constants having high values of derivatives play major roles in the optimisation and those with low values of derivation would be ignored.
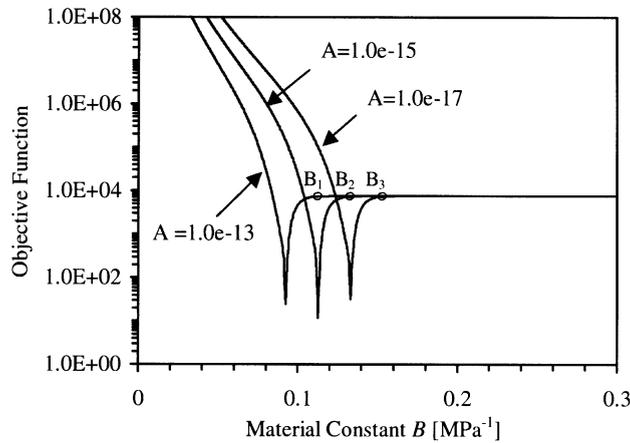
Fig. 2. Variation of OFI with the material constant B for different values of constant A.

(iii) The contours of the objective functions are very narrow, and a steep sided valley is normally formed for many constants, as shown in Figs. 1 and 2. Such functions have already been shown to be difficult to optimise. For example, the Rosenbrock's function, which is a well-known test function with such narrow ridges, has only two variables but is difficult to optimise by traditional gradient-based optimisation algorithms.

The second problem could be overcome by introducing a normalisation method. The normalised material constants z_i $(i = 1, \ldots, s)$ are defined as

$$z_i = z_{gi} + \frac{z_{ui} - z_{gi}}{x_{ui} - x_{gi}}(x_i - x_{gi}), \quad i = 1, \ldots, s, \tag{9}$$

where x_i $(i = 1, \ldots, s)$ are the original constants which need to be optimised. z_{gi}, z_{ui}, x_{gi} and x_{ui} are the lower and upper bounds of z_i and x_i, $i = 1, \ldots, s$, respectively. Here, z_{gi} and z_{ui}, $i = 1, \ldots, s$, are set to 1.0 and 10.0.

Unfortunately, the first and third problems cannot be solved easily by any traditional optimisation techniques. One method for dealing with the first problem might be to divide the domains of individual material constants into small regions. Starting values of the constants are selected by considering all possible combinations of the small regions for all the constants. Then traditional optimisation methods can be used to search the optimal values of the constants for each set of starting values. However, this is impractical since too much CPU time would be required to solve the problem. Another approach to solving the optimisation problem is to adopt new algorithms that are good at dealing with plateaus and long and narrow ridges. In this paper, three evolutionary algorithms are introduced for solving the optimisation problem. They can handle functions with plateaus and ridges much better than traditional gradient-based algorithms. They are also more efficient than binary GAs. All three algorithms enjoy global convergence.

4. Evolutionary optimisation methods

The three evolutionary optimisation algorithms used in this paper are classical evolutionary programming (CEP) [8], fast evolutionary programming (FEP) [13,14], and improved fast evolutionary programming (IFEP) [14,15]. This section describes implementation details of these three algorithms.

4.1. Classical evolutionary programming (CEP)

The CEP was first proposed as an approach to artificial intelligence. It has been recently applied with success to many numerical and combinatorial optimisation problems. Optimisation by CEP can be summarised into two major steps:

(i) mutate the solutions in the current population;
(ii) select the next generation from the mutated and the current solutions.

These two steps can be regarded as a population-based version of the classical generate-and-test methods, where mutation is used to generate new solutions (offspring) and selection is used to test which of the newly generated solutions should survive to the next generation. The classical evolutionary programming is implemented as follows [14].

Step 1: Generate the initial population of μ individuals, and set the initial iteration $k = 1$. Each individual is taken as a pair of real-valued vectors, (X_i, η_i), $\forall i \in \{1, \ldots, \mu\}$, where X_i's are objective variables (or material constants) and η_i's are standard deviations for Gaussian mutations (also known as strategy parameters in self-adaptive evolutionary algorithms).

Step 2: Evaluate the fitness score for each individual (X_i, η_i), $\forall i \in \{1, \ldots, \mu\}$, of the population based on the objective function, $f(X_i)$.

Step 3: Each parent (X_i, η_i), $i = 1, \ldots, \mu$, creates a single offspring (X_i', η_i') by: for $j = 1, \ldots, n$:

$$x_i'(j) = x_i(j) + \eta_i(j)N_j(0, 1), \tag{10}$$

$$\eta_i'(j) = \eta_i(j)\exp(\tau'N(0, 1) + \tau N_j(0, 1)), \tag{11}$$

where $x_i(j), x_i'(j), \eta_i(j)$ and $\eta_i'(j)$ denote the jth component of the vectors X_i, X_i', η_i and η_i', respectively. $N(0, 1)$ denotes a normally distributed one-dimensional random number with mean zero and standard deviation one. $N_j(0, 1)$ indicates that the random number is generated anew for each value of j. The factors τ and τ' are commonly set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$.

Step 4: Calculate the fitness of each offspring $(X_i', \eta_i'), \forall i \in \{1, \ldots, \mu\}$.

Step 5: Conduct pairwise comparison over the union of parents (X_i, η_i) and offspring (X_i', η_i'), $\forall i \in \{1, \ldots, \mu\}$. For each individual, q opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is no smaller than the opponent's, it receives a "win".

Step 6: Select the μ individuals out of (X_i, η_i) and (X_i', η_i'), $\forall i \in \{1, \ldots, \mu\}$, that have the most wins to be parents of the next generation.
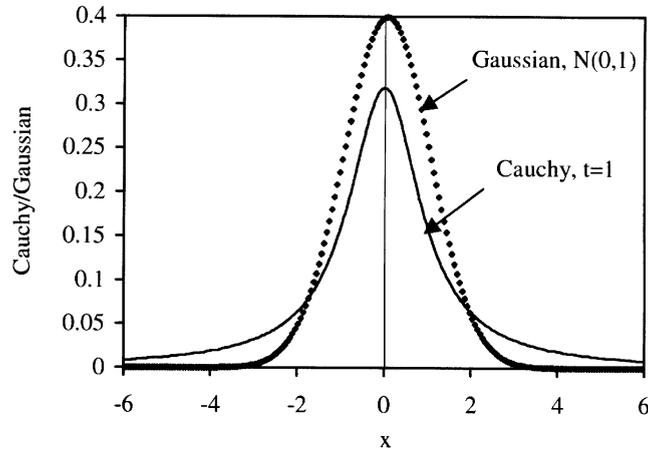
Fig. 3. Comparison of Cauchy and Gaussian density functions.

Step 7: Stop if the halting criterion is satisfied; otherwise, $k = k + 1$ and go to Step 3.

The Gaussian density function with variance 1 (cf. Fig. 3) centred at the origin is defined by

$$f_g(x) = \frac{1}{\sqrt{2n}} e^{-(x^2/2)}, \tag{12}$$

where $-\infty < x < \infty$.

4.2. Fast evolutionary programming (FEP)

The FEP is exactly the same as the CEP described in Section 4.1 except for that Eq. (10) is replaced by the following:

$$x_i'(j) = x_i(j) + \eta_i(j)\delta_j, \tag{13}$$

where δ_j is a Cauchy random variable and is generated anew for each value of j. The one-dimensional Cauchy density function centred at the origin is defined by

$$f_t(x) = \frac{1}{\pi} \frac{1}{1 + x^2}, \tag{14}$$

where $-\infty < x < \infty$.

The shape of $f_t(x)$ resembles that of the Gaussian density function but approaches the axis so slowly that its expectation does not exist. As a result, the variance of the Cauchy distribution is infinite. Fig. 3 shows the difference between Cauchy and Gaussian density functions by plotting them in the same scale. It is clear from Fig. 3 that Cauchy mutation is more likely to generate an offspring further away from its parent than Gaussian mutation due to its long flat tails. It is expected to have a higher probability of escaping from a local optimum or moving away from a plateau, especially when the "basin of attraction" of the local optimum or the plateau is large relative to the mean step size. On the other hand, the smaller hill around the centre in

Fig. 3 indicates that Cauchy mutation spends less time in exploiting the local neighbourhood and thus has a weaker fine-tuning ability than Gaussian mutation ($N(0,1)$) in small to mid-range regions.

4.3. Improved fast evolutionary programming (IFEP)

IFEP uses the idea of mixing search biases to mix Cauchy and Gaussian mutations. Unlike some switching algorithms which have to decide when to switch between the different mutations during search, IFEP does not need to make such decision and introduces no additional parameters. It differs from FEP and CEP only in Step 3 of the algorithms described in Sections 4.1 and 4.2. Instead of using Eq. (10) (for CEP) or Eq. (13) (for FEP) alone, IFEP generates two offspring from each parent, one by Cauchy mutation and the other by Gaussian. The better one is then chosen as the surviving offspring. The rest of IFEP is exactly the same as FEP and CEP.

5. Determination of material constants using the evolutionary approach

An optimisation software package has been developed using C++ based on the three evolutionary algorithms. The unified creep damage constitutive equations are numerically integrated using the fourth order Runge–Kutta method and the time step for the integration is automatically controlled. This ensures that the creep curves can be accurately computed. Detailed studies have been carried out for the two formulated objective functions using the three evolutionary algorithms.

Fig. 4 shows the evolution of objective functions with respect to the number of generations for the two objective functions and the three algorithms. For OFI shown in Fig. 4(a), the residual decreases steadily for all three algorithms in the first 200 generations. Then the decrease slows down. This indicates that all three algorithms were able to find a good solution in the search space quickly (i.e., in the first 200 generations or so). As analysed theoretically and empirically [14], FEP and IFEP performed slightly better than CEP in terms of convergence speed. It is important to note that all three algorithms were randomly initialised. Yet they all converged to similar values around 200 generations. The results show that the evolutionary algorithms are much more robust than previous methods which are highly dependent on initial values. There is no need to search for a good initial value manually through a tedious trial and error process.

Similar observations can be made from the experimental results using objective function II, which are shown in Fig. 4(b). Although a different objective function was used here, the evolutionary algorithms were still able to find similarly good solutions from random initial values. This illustrates another aspect of the robustness of evolutionary algorithms—they work well for different objective functions.

The optimised material constants are summarised in Table 3 for the two objective functions and the three evolutionary methods. The residuals are also given for the same evolutionary methods with different objective functions for comparison. From the residuals given in Table 3, it can be seen that IFEP performs better than the other two although the difference is small.

Figs. 5 and 6 show how well the evolutionarily optimised curves compare with the experimental data, where solid lines are computed creep curves using the corresponding constants listed in Table 3. The data are true creep data for the aluminium alloy given by Kowalewski et al. [1]. The best fitting results are given by Fig. 5(c). However, the quality of the fittings are all reasonably
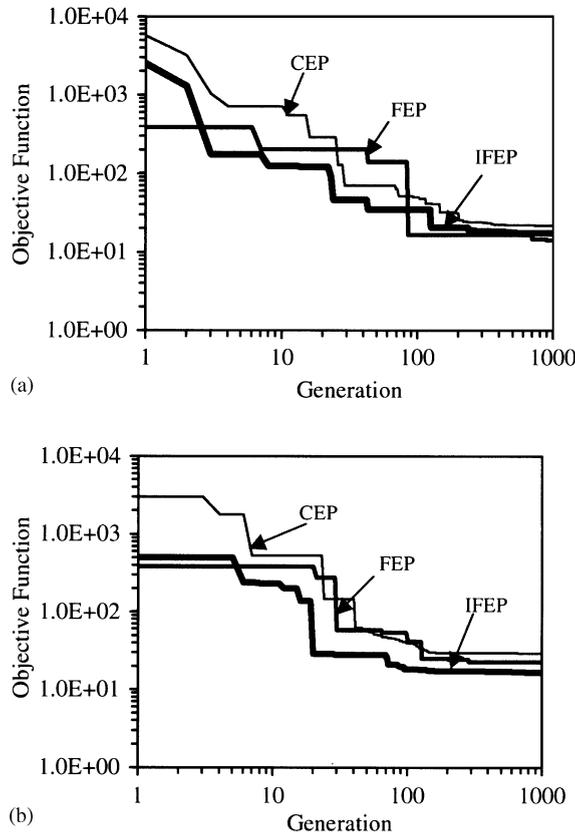
Fig. 4. Comparison of convergences of the three optimization methods: CEP, FEP and IFEP for: (a) OFI; and (b) OFII.

Table 3
List of the determined material constants and residuals for the two objective functions and the three optimisation methods

Objective function	Method	A	B	h	H^a	K_c	D	Residual	Residual of the other function[a]
	CEP	1.95e-15,	0.1035,	1.44e2,	0.4632,	3.27e-4,	2.75	13.3217	17.36284
OFI	FEP	2.07e-15,	0.1147,	3.59e4,	0.1072,	2.55e-4,	2.75	12.9634	22.04500
	IEEP	1.08e-14,	0.1132,	6.02e5,	0.1471,	1.10e-4,	2.75	9.45553	14.16029
	CEP	3.31e-17,	0.1200,	4.15e2,	0.0808,	6.09e-4,	2.75	29.2140	22.07700
OFII	FEP	3.52e-16,	0.1103,	2.16e2,	0.2248,	4.57e-4,	2.75	22.5608	16.73375
	IFEP	1.06e-13,	0.0922,	9.81e5,	0.0501,	7.48e-5,	2.75	16.3721	11.61950

[a] The column gives the values of OFII at the minima of OFI in the row OFI, and the values of OFI at the minima of OFII in the row OFII.

good from the practical point of view. This indicates that all three evolutionary optimisation methods are suitable for determining the material constants for the unified creep damage constitutive equations from experimental results, although the IFEP method provides slightly better optimisation results.
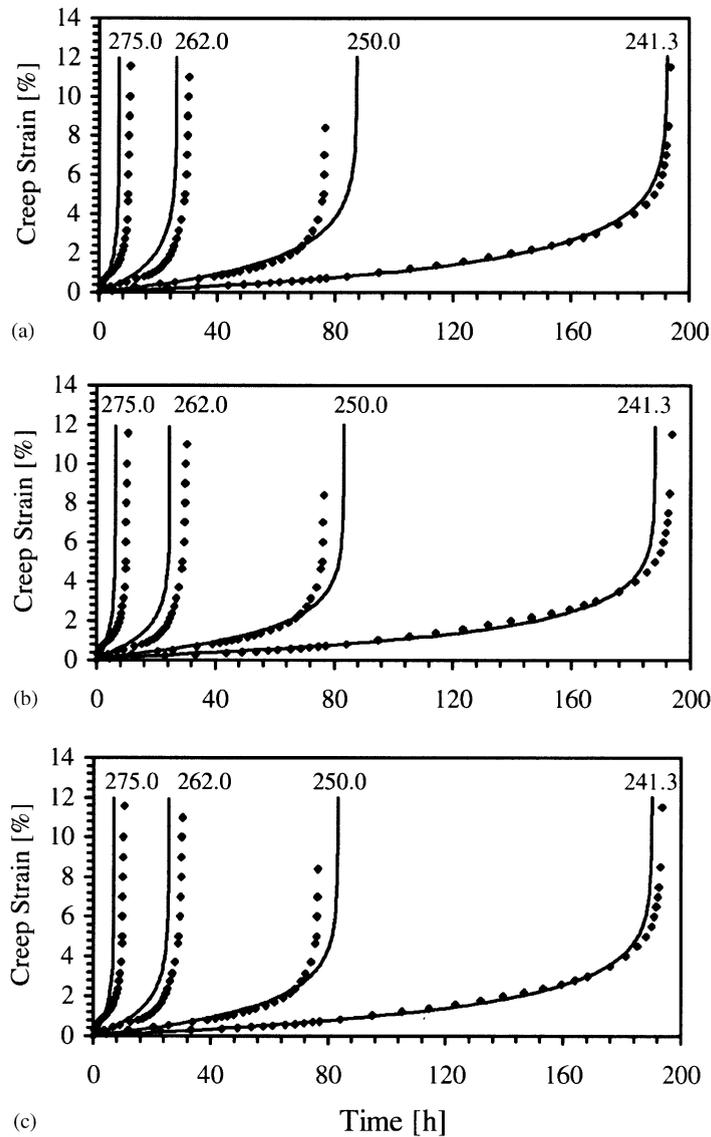
Fig. 5. Comparison of experimental (legends) and computed (solid lines) creep curves for aluminium alloy at $150°C$. The computations were carried out using the optimised sets of material constants obtained from the three methods: (a) CEP; (b) FEP; and (c) IFEP, for the OFI. The numbers given for each curve refer to stress levels in MPa.

6. Conclusions

This paper has analysed the difficulties in determining the material constants within unified creep damage constitutive equations from experimental data. It is found that the objective functions to be optimised contain narrow ridges and large plateaus that make the optimisation particularly difficult
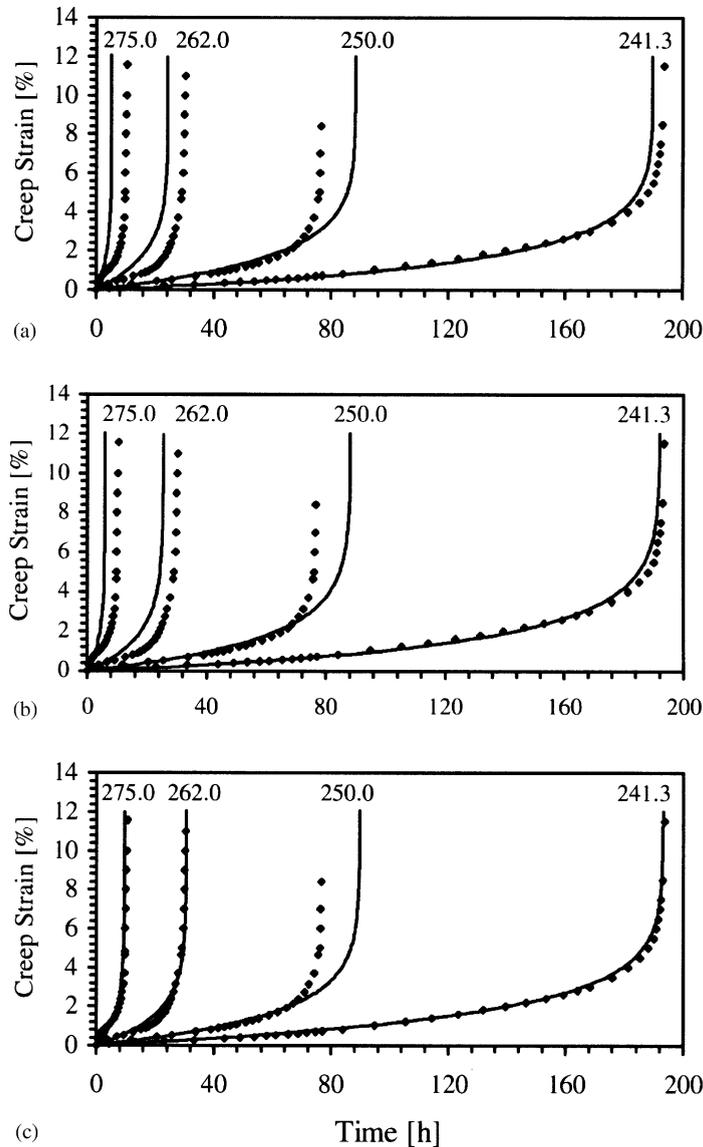
Fig. 6. Comparison of experimental (legends) and computed (solid lines) creep curves for aluminium alloy at 150°C. The computations were carried out using the optimised sets of material constants obtained from the three methods: (a) CEP; (b) FEP; and (c) IFEP, for the OFII. The numbers given for each curve refer to stress levels in MPa.

for classical optimisation methods. Out of the two objective functions formulated, OFI provides slightly better results in terms of better fitting between the computed and experimental data.

A PC software package has been developed for determining the material constants in unified creep damage constitutive equations. The package includes three evolutionary programming algorithms which are particularly suitable for solving the optimisation problems. The algorithms can be started

from any random initial values without tedious manual selection of starting points. They can also work well with different objective functions.

Evolutionary algorithms are robust optimisation techniques that can deal with complex, non-differentiable and non-convex functions. This is particularly useful to determine unified creep damage constitutive equations. Evolutionary programming algorithms use real-value vectors and self-adaptation in their chromosome representation and thus perform better than binary genetic algorithms on complex continuous functions, such as those described in this paper.

The creep damage constitutive Eqs. (1)–(4) are regarded just as an example of application to demonstrate the effectiveness of the optimisation algorithms. The effect of primary creep is marginal, especially for the low stress test results. This leads to scattered values of h and H^* determined from different optimisation algorithms, as shown in Table 3. Future research needs to be carried out to identify/develop a set of damage constitutive equations to accurately model the creep damage behaviour of the material. The traditional creep damage constitutive equations, such as single damage variable equations developed by Kachanov [16], Hayhurst [17], etc., will be considered.

Acknowledgements

References

[1] Kowalewski ZL, Lin J, Hayhurst DR. Experimental and theoretical evaluation of a high-accuracy uniaxial creep testpiece with slit extensometer ridges. International Journal of Mechanical Sciences 1994;36(8):751–69.

[2] Lin J, Hayhurst DR, Dyson BF. A new design of uni-axial creep testpiece with slit extensometer ridges for improved accuracy of strain measurement. International Journal of Mechanical Sciences 1993;35(1):63–78.

[3] Kowalewski ZL, Hayhurst DR, Dyson BF. Mechanisms-based creep constitutive equations for an aluminium alloy. Journal of Strain Analysis 1994;29:309–16.

[4] Lin J, Yang JB. GA-based multiple objective optimisation for determining viscoplastic constitutive equations for superplastic alloys. International Journal of Plasticity 1999;15:1181–96.

[5] Perrin IJ, Hayhurst DR. A method for transformation of creep constitutive equations. International Journal of Pressure Vessels 1996;68:299–309.

[6] Lin J, Hayhurst DR. Constitutive equations for multi-axial straining of leather under uni-axial stress. European Journal of Mechanics A/Solids 1993;12(4):471–92.

[7] Zhou M, Dunne FP. Mechanism-based constitutive equations for the superplastic behaviour of a titanium alloy. Journal of Strain Analysis 1996;31(3):187–96.

[8] Fogel DB. System identification through simulated evolution: a machine learning approach to modelling. Needham Heights, MA, USA: Ginn Press, 1991.

[9] Fogel DB, Atmar JW. Comparing genetic operators with Gaussian mutations in simulated evolutionary process using linear systems. Biological Cybernetics 1990;63(2):111–4.

[10] Wilde DJ. Monotonicity and dominance in optimal hydraulic cylinder design. ASME Journal of Engineering for Industry 1975;97(4):1390–4.

[11] Papalambros PY, Li LH. A production system for use of global optimisation knowledge. ASME Journal of Mechanisms, Transmissions and Automation in Design 1985;107:277–84.

[12] Li BL, Chen Y. Optimum design technique using local and global monotonicity analysis. Journal of Southwest Jiaotong University 1988;1:19–31.

[13] Yao X, Liu Y. Fast evolutionary programming. Proceedings of the fifth annual conference on evolutionary programming (EP'96), San Diego, USA. Cambridge, MA: MIT Press, 1996. p. 451–60.

[14] Yao X, Liu Y, Lin G. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 1999;3(2):82–102.

[15] Yao X, Lin G, Liu Y. An analysis of evolutionary algorithms based on neighbourhood and step sizes. In: Angeline PJ, Reynolds RG, McDonnell JR, Eberhart R, editors. Evolutionary Programming VI: Proceedings of the Sixth Annual Conference on Evolutionary Programming (EP'97), Lecture Notes in Computer Science, vol. 1213. Berlin: Springer, 1997. p. 297–307.

[16] Kachanov LM. In: Kennedy AJ, editor. The theory of creep (English translation). Wetherby, England: British Library, 1960.

[17] Hayhurst DR. Creep rupture under multi-axial states of stress. Journal of Mechanics and Physics of Solids 1972;20:381.