Genetic programming approach for modelling of tensile strength of cold drawn material

L. GUSEL[*], R. RUDOLF, N. ROMCEVIC[a], B. BUCHMEISTER
University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia
[a]*Institute of Physics, University of Belgrade, Pregrevica 118, 11080 Beograd, Serbia*

Genetic programming is one of the most general evolutionary computation methods. It is an automated method for creating a working computer program from a problem's high-level statement. Genetic programming does this by genetically breeding a population of computer programs using the principles of Darwin's natural selection and biologically inspired operations. In our research, material was formed by drawing using different process parameters and then determining the tensile strengths (dependent variable) of the specimens. On the basis of experimental data, various different prediction models for the tensile stress were developed by the genetic programming method. To make a comparison, the models for tensile stress were also developed by the standard regression method. The accuracies of the best genetic programme models were proved by a testing data set and comparison between the genetic and regression models.

(Received February 15, 2012; accepted April 11, 2012)

Keywords: Cold drawing, Genetic programming, Modelling, Tensile stress, Effective strain

1. Introduction

Many modelling methods for predicting dependent output variables have been developed to reduce the costs of the experiments and computer computations. Traditional methods often employed to solve real complex problems tend to inhibit elaborate explorations of the search space. They can be expensive and often result in sub-optimal solutions. Because of the pre-specified size and shape of the model, the latter is often incapable of capturing complex relationships between influencing parameters. Evolutionary Computation (*EC*) is generating considerable interest for solving real engineering problems. It is proving robust in delivering global optimal solutions and helping to resolve those limitations encountered in traditional methods. *EC* harnesses the power of natural selection to turn computers into optimization tools. This is very applicable to different problems in the manufacturing industry [1]. One of most important *EC* methods is genetic programming (*GP*) which is an evolutionary computation method for imitating the biological evolution of living organisms.

Different researches for modelling a tensile strength and other mechanical characteristics have been carried out using a neural network or genetic algorithms for modelling [2-8], thus forming process parameters but only a few dealing with much more general genetic programming method [10-13,15]. The recent use of genetic algorithms in various domains connected to materials science, solid state physics and chemistry, crystallography and engineering is reviewed in [2]. All these methods can be efficiently used also for obtaining other characteristics [6, 8], especially in simple system modelling. In some researches integration of artificial neural network with genetic algorithm in order to accelerate the search of optima models were applied [3, 5]. When compared to some other released papers on this subject, where mostly genetic algorithms [2-5], neural networks [4, 5-8] and regression analysis [9] are used to obtain suitable models, in this paper we describe a new approach for modelling of mechanical characteristics of formed material - an evolutionary computation method. The *GP* method is often used for complex system modelling, but it can also be effectively used for the modelling of a relatively simple system, such as the systems described in our paper. Experimental data obtained during the cold drawing processes under different conditions serves as an environment to which, during simulated evolution, models for the tensile strength have to be adapted. Different values for effective strains and coefficients of friction were used as independent input variables (parameters), while tensile strength was a dependent output variable. The prediction accuracy of different models was proved with the testing data set.

2. Method used

Two reliable methods of modelling are used in this paper: genetic programming and regression analysis, which are widely known and used. Genetic programming is probably the most general approach from among evolutionary computation methods in which the structures subject to adaptation are those hierarchically organized computer programs whose size and form dynamically change during simulated evolution. The space for solutions in the *GP* method is the huge space of all possible computer programs consisting of components describing the problem area studied. The aim of *GP* is to find out the computer programme that best solves the problem. Possible solutions in *GP* are all those possible computer programmes that can be composed in a recursive manner from a set of function genes *F* and a set of terminal genes *T*. Function genes *F* are arithmetical functions, Boolean functions, relation functions, etc., while terminal genes are numerical constants, logical constants, variables, etc. [1]. The initial population is obtained by the creation of random computer programmes consisting of available function genes from set *F* and available terminal genes from set *T*. Each program represents a random point in the searching space. The creation of an initial population is a blind random search for solutions within the huge space of possible solutions. The next step is the calculation of an individual's adaptation to the environment (i.e., calculation of fitness for each computer programme). Fitness is a guideline for modifying those structures undergoing adaptation. Computer programs change in *GP*, in particular during genetic operations regarding reproduction and crossover. The reproduction operation gives a higher probability of selection to more successful organisms. They are copied unchanged into the next generation. The crossover operation ensures the exchange of genetic material between computer programmes. The mutation operation increases the genetic diversity of a population. After finishing the first cycle, which includes creation of the initial population, calculation of fitness for each individual of the population, and genetic modification of the contents of the computer programmes and formation of a new population, an iterative repetition of fitness calculation and genetic modification follows. After a certain number of generations the computer programs are usually much better adapted to the environment. The definition of the environment depends on the problem dealt with. The evolution is terminated when the termination criterion is fulfilled. This can be a prescribed number of generations or sufficient quality of the solution. In order to obtain a successful solution, the problem must be processed over several independent runs. The number of runs required for the satisfactory solution depends on the difficulty of the problem.

3. Experimental work

The aim of the experimental work was to determine the influence of the effective strain ε_e and coefficient of friction μ during cold drawing on the change of tensile strength of cold drawn copper alloy. Copper alloy rods were deformed by cold drawing under different conditions. The drawing speed was 20 m/min and the angle of drawing die was $\delta = 28°$. Rods were drawn from an initial diameter of D=20 mm to six different diameters (i.e. six different effective strains). Three different lubricants with different coefficients of friction (μ=0.07, μ=0.11 and μ=0.16) were used for the drawing process. In order to evaluate the tensile strength, standard specimens for tensile tests were prepared from locations in the middles of the drawn rods. In this way we obtained 18 different experimental specimens. The tensile strengths of all specimens were determined by providing three tensile tests for each specimen in order to provide reliable results. The results (average values) for tensile strength are presented in Table 1**.** Experimental data serve as an environment which, during simulated evolution, models for tensile strength have to adapt.

Table 1. Experimental results.

Nr.	Effective strain ε_e	Coef. of friction μ	Tensile strength R_m *[N/mm^2]*
initial spec.	/	/	481
1	0.10	0.07	498
2	0.21	0.07	513
3	0.32	0.07	524
4	0.44	0.07	530
5	0.57	0.07	532
6	0.71	0.07	535
7	0.10	0.11	502
8	0.32	0.11	531
9	0.71	0.11	540
10	0.10	0.16	506
11	0.44	0.16	536
12	0.71	0.16	544
13	0.21	0.11	517
14	0.44	0.11	530
15	0.57	0.11	530
16	0.21	0.16	529
17	0.32	0.16	535
18	0.57	0.16	547

4. Modelling of tensile strength by genetic programming

In the *GP* method the initial random population *P (t)* consists of randomly generated organisms which are, in fact, mathematical models. The variable *t* represents the generation time. Each organism in the initial population consists of the available function genes *F* and terminal genes *T*. Terminal genes are in fact independent variables: strain and coefficient of friction. Random floating-point numbers within the range [-10, 10] are added to the set of terminals to increase the genetic diversities of the organisms. Function genes *F* are basic arithmetical, exponential and cosine functions.

4.1 Evolutionary parameters

The absolute deviation *R* (*i*, *t*) of individual model *i* (organism) in generation time *t* for the *GP* approach, was introduced as the standard raw fitness measurement [1]:

$$R(i,t) = \sum_{j=1}^{n} \left| E(j) - P(i,j) \right|, \tag{1}$$

where *E* (*j*) is the experimental value for measurement *j*, *P* (*i*, *j*) is the predicted value returned by the individual model *i* for measurement *j*, and *n* is the maximum number of measurements. The aim of the optimisation task is to find those models that equation (1) would give as having as low an absolute deviation as possible. However, because it is unnecessary that the smallest values of the above equation also means the smallest percentage deviation of this model, the average absolute percentage deviation of all measurements for individual model *i* was defined as [1]:

$$\Delta(i) = \frac{R(i,t)}{|E(j)|\,n} \cdot 100\% \qquad (2)$$

Equation (2) was not used as the fitness measurement for evaluating population, but only to find the best organism in population after completing the run.

In the *GP* method, reproduction, crossover, and mutation operations were used for altering the population *P (t)*. Evaluation and altering of the population *P (t)* were repeated until termination condition had been fulfilled. The termination condition was the prescribed maximum number of generation to be run. Reproduction, crossover, and mutation were used as genetic operations. The evolutionary processes were controlled by the following evolutionary parameters: population size 1,000, maximum number of generations to be run 50, probability of reproduction 0.15, probability of crossover 0.7, probability of mutation 0.15, maximum depth for initial random organisms 6, maximum depth of mutation 6, and maximum permissible depth of organisms after crossover 12.

4.2. Realisation of the evolutionary process

The modelling of tensile strength was carried out by the special *GP* system (computer program). Each individual *GP* run started with the training phase by the training data set shown in Table 1 (Nr.1 to Nr.12). The testing data set (Table 1: Nr.13 to Nr. 18) was not included within the training range. Each run lasted up to generation 30 when it was temporarily interrupted. If an average percentage deviation Δ (i) of at least one prediction model (organism) in the population was smaller than 5%, the evolution of the population continued up to generation 50, otherwise it was terminated. After each training phase, the accuracy of predicting the best models was tested using the testing data set. As much as 400 independent runs were executed for modelling the tensile strength. The *GP* models in our research were developed originally as a prefix LISP expression, and then converted into an infix notation.

5. Genetic modelling results

GP modelling was executed by two different genes function sets *F* = (+,-,*, /) and *F* = (+,-,*, /, ZEXP). The best (the most accurate) model obtained with genes function set *F* = (+,-,*, /) is quite complicated and is written is LISP as:

```
(- (+ (% (+ -6.72502 μ) (- (+ (- (* ε -5.63273) (- ε ε)) (*
(+ (+ -6.72502 μ) μ) (* ε (- (* -0.168495 ε) 4.86573))))
(+ ε (% (% -5.63273 ε) (+ -6.72502 ε))))) (* (* -
7.44391 (+ 7.15074 μ)) (+ μ -9.92379))) (+ (- (% (% -
5.63273 ε) (+ -3.00006 ε)) (* (+ (+ -6.72502 (+ (- μ μ)
(+ -5.92924 μ))) μ) (% (* ε -6.44657) (+ (% 1.97763
μ) -7.44391)))) (- (% (% -9.3319 ε) (* 7.25773 (+
7.15074 (% (% μ μ) (+ (- μ μ) (+ -5.92924 μ)))))) (* (%
7.8429 ε) (+ -0.289511 ε))))))
```

The same model can be written as the mathematical expression:

$$7.842 - \frac{2.27}{\varepsilon} + \frac{5.632}{\varepsilon(\varepsilon - 3)} - (-73.871 + 7.443\mu)(7.15 + \mu) - \frac{12.893\mu - 81.574}{1.977 - 7.443\mu} + \frac{7.623 - 1.285\mu}{41.398\varepsilon - 7.15\varepsilon\mu} + \frac{\mu - 6.725}{\frac{5.632}{\varepsilon^2 - 6.725\varepsilon} - 6.632\varepsilon + (-4.865\varepsilon - 0.168\varepsilon^2)(2\mu - 6.725)} \qquad (3)$$

The model (3) was generated in generation No. 49 and has the average percentage deviation of the training data Δ (i) = 0,12 % and that of the testing data Δ(i) = 0,13%.

Percentage deviation is in fact the percentage error between a single experimental value and the value predicted by the genetic model.

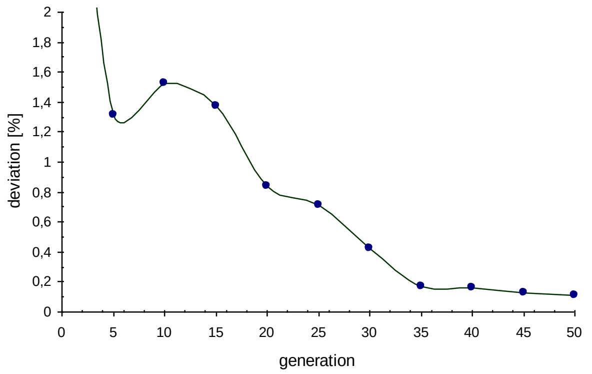

*Fig. 1. Percentage deviation curve between the best models regarding individual generation and experimental results (F = +, -, *, /).*

Fig. 1 shows the percentage deviation curve (Δ_i) between the best model regarding individual generation and experimental results when using the set of function genes *F* = {+, -, *, /}. It is obvious that in early generations the best models are not as precise as the models generated in late generations. The relatively slow improvement of the best models in later generations (after generation 35) is due to the unification trends of the population leading to the shortage of new genetic ideas.

Slightly better accuracy (Δ (i) = 0, 11 %, and that of the testing data Δ(i) = 0,13%) of the *GP* model was obtained when the genes function set which includes the exponent function was used: *F* = (+,-,*, /, ZEXP):

```
(+ (% (% (+ (- (* -9.08852 9.68961) (% (ZEXP
3.27675) 7.46124)) (- (% μ μ) (% μ μ))) (* (- (% ε μ) (*
(- (% ε μ) (* 3.27675 ε)) ε)) (- (ZEXP 4.38517) (-
9.68448 -5.12255)))) (- (+ (- (* ε 1.3429) (ZEXP (%
(% (+ μ ε) ε) -7.12236))) (* (- (- 9.68448 -5.12255) (%
μ μ)) (* ε μ))) (* (ZEXP (% (% (+ μ ε) ε) -7.12236)) (-
(% ε μ) (* 3.27675 ε))))) (- (+ (- (+ (ZEXP ε) (ZEXP
6.23761)) (- (% μ ε) (* 6.11433 (+ μ 6.68026)))) (* (+
(- (* -9.08852 9.68961) (- (% ε μ) (% μ μ))) (- μ (+ -
7.61921 -8.03228))) (ZEXP (* -5.76456 ε)))) (* (ZEXP
(* (+ -9.80642 (- μ μ)) (* (% (- μ μ) (+ ε ε)) (- (ZEXP
4.38517) (- μ μ))))) (+ (* (ZEXP (% (% (+ μ ε) ε) -
7.12236)) (- (% ε μ) (* 3.27675 ε))) (- (- 9.68448 -
5.12255) (* (- (- 9.68448 -5.12255) (- μ μ)) (* ε μ))))))
```

This model can be written as the mathematical expression:

$$537.673+e^{\varepsilon}+6.114\mu-\frac{\mu}{\varepsilon}+e^{-5.764\varepsilon}\left(\mu-\varepsilon/\mu-71.412\right)+\frac{e^{\frac{-0.1404\mu}{\varepsilon}}\varepsilon(0.869-2.847\mu)}{\mu}-$$
$$14.807\varepsilon\mu-\frac{0.03e^{\frac{0.1404\mu}{\varepsilon}}\mu^{2}}{(0.305\varepsilon-0.305\varepsilon^{2}+\varepsilon^{2}\mu)\left(-0.062\mu+\varepsilon\left(-0.062+\left(0.206+0.097e^{\frac{0.1404\mu}{\varepsilon}}\right)\mu+\mu^{2}e^{\frac{0.1404}{\varepsilon}}\right)\right)} \tag{4}$$

The most accurate simple model with model has average deviation Δ (i) = 1,24 % (testing data Δ(i) = 1,72%) is:

```
(+ (* (* ε 8.72112) 8.72112) (ZEXP 6.19403))
```

Written as the mathematical expression:

$$487.84 + 76.05\,\varepsilon \tag{5}$$

Fig. 2 shows the number of genes of the best genetic model (generated with function genes *F* = {+, -, *, /}) in each generation. In generation 1 created randomly, the best model contains large number of genes (83) and then, in the second generation, there is large decrease to only 19 genes in the best model. From generations 3 the number of genes in the best model increases constantly and reaches the value of 95 in generation 50. The higher number of genes usually means higher complexity of the genetic model.

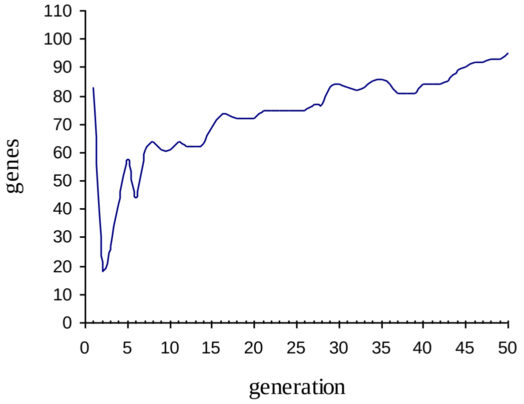

*Fig. 2. The number of genes in the best model in each generation obtained by function genes F = {+, -, *, /}.*

6. Regression analysis result

A mathematical model for regression method was chosen according to [14]:

$$y(x)=b_0+\sum_{i=1}^{n}b_ix_i+\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}b_{ij}x_{ij}+\sum_{i=1}^{n}\sum_{j=i}^{n}b_{ij}x_{ij}^{2} \tag{6}$$

In the equation (6) y is dependent variable, x_i, x_{ij} are independent variables, while b_0, b_i, b_{ij} are coefficients to be determined by using regression analysis. In our case, the dependent variable was tensile strength (R_m), while effective strain ε_e and coefficient of friction μ were independent variables. Coefficients b_0, b_i and b_{ij} were determined by using the regression analysis computer program. By inserting the computed values of coefficients into the equation (6.1), the regression model for impact toughness can be presented as:

$$R_m = 480{,}781 + 150{,}864\,\varepsilon + 66{,}115\,\mu - 133{,}762\,\varepsilon^2 - 238{,}080\,\mu^2 + 162{,}736\,\varepsilon\mu \tag{7}$$

Equation (7) represents a mathematical model of effective strain's influence and the coefficient of friction on tensile strength for chosen material within the experimental area. It has the average percentage deviation of the training data set Δ(i) = 0,21 % and that of the testing data set Δ(i) = 0,24 %.

When regression models are compared to genetic programming models, the first important difference is the complexities of the genetic models. Due to its evolutionary concept, genetic programming models are complex, with lots of genes, and the forms of these models can be confusing. But the form of a *GP* model (5) is very simple. Of course, when it comes to the accuracies of different models, *GP* models show much greater accuracy then regression models.

7. Conclusions

The models presented in the paper are a result of the self-organization and stochastic processes taking place during simulated evolution, and not of human intelligence. The accuracies of the models developed during the training phase were also confirmed using testing data not included within the training range. Only two genetically developed models out of many successful solutions are presented here.

The main advantage of presented *GP* method, when compared with other modelling methods, is much better accuracy of *GP* obtained models and wider range of suitable models. Because of the high precision regarding the models developed by the genetic programming approach, with the proposed concept, the excessive number of experiments/simulations can be avoided, which leads to the reduction of the product development costs. The research showed that simple, and in the same time, very precise models are often hard to reach. This is due to the fact that evolution is a stochastic process, and, therefore, rationality in the development of the models is rare. However, in many metal-forming processes the accuracy of prediction is of vital importance, not the model complexity.

References

[1] M. Korns, Springer, New York, (2010).
[2] W. Paszkowicz, Materials and Manufacturing Processes, **24**(2), 174 (2009).
[3] D. Farrusseng, F. Clerc, C. Mirodatos, R. Rakotomalala, Computational Materials Science, **45**(1), 52 (2009).
[4] R. C. Dimitriu, H. K. D. H. Bhadeshiaa, C. Fillonb, C. Polonib, Materials and manufacturing Processes, **24**, 10 (2009).
[5] F. Pettersson, A. Biswas, P. K. Sen, H. Saxén, N. Chakraborti, Materials and Manufacturing Processes, **24**(3), 320 (2009).
[6] P. Udhayakumar, S. Kumanan, International Journal of Simulation Modelling, **9**(1), 28 (2010).
[7] S. Ganguly, S. Datta, N. Chakraborti, Computational Materials Science, **45**, 158 (2009).
[8] S. Yigit, R. Eryigit, F. V. Çelebi, Optoelectron. Adv. Mater. – Rapid Commun. **5**(9), 1026 (2011).
[9] M. Naderi, M. Ketabchi, M. Abbasi, W. Bleck, Journal of Materials Processing Technology, **211**(6), 1117 (2011).
[10] L. Gusel, M. Brezocnik, Computational Material Science, **37**, 476 (2006).
[11] W. Afzal, R. Torkar, Expert Systems with Applications, **38**(9), 11984 (2011).
[12] N. Chakraborti, R. Sreevathsan, R. Jayakanth, B. Bhattacharya, Computational Materials Science **45**(1), 1 (2009).
[13] M. Brezocnik, B. Buchmeister, L. Gusel, Materials and Manufacturing Process., **26**(3), 501 (2011).
[14] R. Nisbet, J. Elder, G. Miner, Elsevier Academic Press, London, (2009).
[15] L. Gusel, M. Brezocnik, R. Rudolf, I. Anzel, Z. Lazarevic, N. Romcevic, Optoelectron. Adv. Mater. – Rapid Commun. **4**(3), 395 (2010).

*Corresponding author: leog@uni-mb.si