NASA-CR-200244

NAG3-1422

IN-61-CR
8796
p- 9

SOFT COMPUTING METHODS IN DESIGN OF SUPERALLOYS

K. J. Cios, L. Berke *, A. Vary * and S. Sharma
University of Toledo, Toledo, Ohio 43606, U.S.A.
and NASA Lewis Research Center *, Cleveland, Ohio

Abstract

Soft computing techniques of neural networks and genetic algorithms are used in the design of superalloys. The cyclic oxidation attack parameter K_a, generated from tests at NASA Lewis Research Center, is modelled as a function of the superalloy chemistry and test temperature using a neural network. This model is then used in conjunction with a genetic algorithm to obtain an optimized superalloy composition resulting in low K_a values.

I. Introduction

In the paper we show the results of research involving application of soft computing techniques to modelling and optimizing alloys. In design and manufacturing of advanced materials such as superalloys, it is required to come up with a material possessing desired output properties. These properties can be expressed as a function of material composition and parameters of the fabrication process. Optimizing the composition of a material can be broken into two problems: finding the function between inputs, like material composition and process parameters, and outputs like strength and density, and then optimizing that function. Such functions are usually highly non-linear and difficult to find. Moreover, the properties of the superalloys are very sensitive to the process fabrication parameters such as temperature, pressure etc. Because of the above we have used neural networks to learn the mapping function between the inputs and outputs.

Optimization can be defined as a process that seeks to improve performance of a system toward some optimal point or a set of points. Local optimization techniques work well for problems that have a relatively 'nice' search spaces and the user has a good feel for the space. If that is not the case global optimization techniques of genetic algorithms are often used.

Barret in [1] used the data generated from tests at NASA Lewis Research Center to rank the Ni- and Co- based superalloys for their cyclic oxidation resistance. The test results were reduced to a single "attack parameter" K_a, and he used multiple linear regression analysis to derive an estimating equation for this parameter as a function of the alloy chemistry and test temperature. This equation was then used to predict the K_a values for similar alloys and also for a design of an optimal superalloy composition.

Soft computing methods of neural networks, genetic algorithms and fuzzy sets have proven to be useful [2] where the conventional methods have limitations. In this work we use the techniques of neural networks and genetic algorithms for modelling and optimization, respectively. The backpropagation neural network is used for modelling and the GENOCOP genetic algorithm is used for optimization, see Fig 1. It will be shown that the neural network modelling of K_a gives as good, or better, a fit as the linear regression model [1]. Optimization of the function learned by the neural network using the genetic algorithm [3] achieves low values for the K_a parameter.

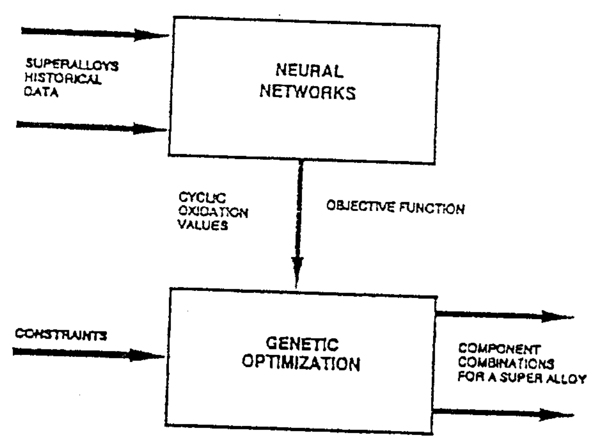

Figure 1. Outline of the neuro-genetic system.

Barret's data [1] was used to train the backpropagation network to model the cyclic oxidation attack parameter K_a as a function of superalloy composition. This trained network was then used as an objective function (K_a) generator for an optimizer using a genetic algorithm, Fig. 1.

In the paper we shall briefly discuss the soft computing methods of neural networks for function approximation in Section II, and genetic algorithms for optimization in Section III. Conclusions are given in Section IV.

II. Function Approximation

Artificial neural networks are composed of many simple non-linear processors called neurons connected in parallel. Each neuron performs computation of the form:

$$o_i = f(s_i) \quad \text{and} \quad s_i = W^T X$$

where $X=(x_1, x_2, \ldots, x_m)$ is the vector input to the neuron and W is the weight matrix with w_{ij} being the weight (connection strength) of the connection between jth element of the input vector and ith neuron. The f() is a non-linear function (usually a sigmoid), o_i is the output of the ith neuron and s_i is the weighted sum of the inputs.

Neural networks can learn from the input/output training data pairs. Once the training is completed the network can be used as a function simulator. The learning capability is a result of the ability of the network to modify the weights through usage of a learning rule. The topology used here is the multi-layer feed-forward network and the learning rule is backpropagation. A neural network with one hidden layer was used to simulate $\log_{10}(K_a)$ as a function of the superalloy chemistry and test temperature. The network had 18 nodes in the input, 36 nodes in the hidden layer and one in the output layer. The superalloys used in the test were Ni- and Co- based and their composition was described by weight percent (wt%) of the components Ni, Co, Cr, Al, Ti, Mo, W, Cb, Ta, C, B, Zr, and Hf. This data is shown in the Appendix.

Barret's [1] fitting of the function using linear regression resulted in the value of R^2 equal 84.43%. We achieved an R^2 value of 86.56% on the same data. Appendix shows the regression and backpropagation results for the comparison of the K_a parameter for the used superalloys. Different results were obtained when multiple tests were conducted for some alloys (experiment repeated) and hence the average values for comparison were used. The trained network was used to predict the K_a value for an alloy, not included in the training data set, being exactly the same as used by Barret. The results shown in Table 1 are better than the ones obtained from regression at both temperatures (1150 & 1200 °C). All values are log to base 10 of the Ka parameter.

Table 1

Temperature	1150 °C	1200 °C
K_a (Observed)	0.7645	1.0865
K_a (Regression)	0.2684	0.7554
K_a (Neural Net)	0.8937	0.9347

III. Optimization

Optimization can be defined as a search towards some optimal point. In most engineering systems attainment of the optimum at any cost is not required, but instead what usually suffices is a "good" solution. Genetic algorithms have proved to be of considerable help towards achieving this goal. The genetic algorithms are global optimizers used to overcome the limitations of many conventional methods like Bayesian/sampling, Monte Carlo, Torn's and simulated annealing [5].

The genetic algorithm (GA) is an evolutionary computation method, useful in performing searches and optimization. A GA involves a set of elements $(x_1, \ldots, x_n)$, called the population $X(t)$ at time t. Each element x_i represents a possible solution and is represented by a string of variables. The standard GA is described as the following sequence of steps [6].

Step 1: Randomly generate an initial population $X(0) = (x_1, x_2, \ldots, x_n)$;
Step 2: Compute the fitness $f(x_i)$ of each individual x_i of the current population;
Step 3: Generate an intermediate population $X_r(t)$ applying the reproduction operator;
Step 4: generate $X(t+1)$ applying other operators to $X_r(t)$;
Step 5: t = t+1; if not (end_test) goto Step 2.

where the most commonly used operators are reproduction, crossover, and mutation.

To improve the objective function value towards an optimum the genetic algorithm only needs the function values at the population points, and not the function itself. In this sense the algorithm is said to be blind. The algorithm [3] uses probabilistic transition rules and random choice as a tool to guide the search towards a region of a search space with likely improvement. The GAs also have the advantage of being able to optimize while avoiding local minima unlike gradient-descent methods. The GA method of optimization is very different from conventional methods and can be characterized by [3, 5] these differences:

- they directly use the code i.e. the parameters
- they search from a population of points instead of a single point
- they are blind to all auxiliary information.
- they use randomized operators.

The algorithm we have used for optimization is the GENOCOP (Genetic Algorithm for Numerical OPtimization) developed at the University of North Carolina, by Zbigniew Michalewicz. The GENOCOP system aims at finding a global optimum (minimum or maximum) of a function subject to linear constraints (equations and inequalities). This algorithm had been demonstrated to successfully optimize both linear and non-linear functions. Even though the algorithm is blind to the function, the functions were needed to generate the function values. We wanted the algorithm to optimize an unknown function, which was simulated on a neural network. The programs were modified so that the function values were generated by another program developed at the University of Toledo, using the backpropagation network.

The problem of designing a superalloy was broken down into two tasks: function approximation and optimization. The backpropagation net was trained using available test data from the tests and thus functioned as a simulator of the K_a parameter. This generated K_a was then used as input to the genetic algorithm, which searched for points with minimum corresponding K_a values. This search led to the results shown in Table 2.

Table 2

GENOCOP solution point at 1100 °C

Element	Weight %
Ni	70.0552444
Co	5.03954935
Cr	9.97962761
Al	3.30380297
Ti	1.36296296
Mo	0.84048849
W	2.05709577
Cb	2.99739814
Ta	3.91278195
C	0.13449860
B	0.00077937
Zr	0.30375364
Hf	0.00200379
V	0.00000000
Re	0.00000000
Cu	0.00000000

The search was restricted to the temperature 1100 °C. The constraints used in finding an alloy composition were obtained from NASA Lewis Research Center and are listed in Table 3.

Table 3

Constraints used in optimization

Lower Limit	Element	Upper Limit
1100	Temp.	1100
50	Ni	100.0
0	Co	10.0
0	Cr	15.0
0	Al	6.0
0	Ti	2.0
0	Mo	2.0
0	W	4.0
0	Cb	3.0
0	Ta	8.0
0	C	0.5
0	B	0.1
0	Zr	1.0
0	Hf	1.0
0	V	0.0
0	Re	0.0
0	Cu	0.0

The obtained results [Table 2] indicate that the desired alloy belongs to group-II alloys [1], ie. chromia/chromite formers. We think that this is a direct result of the given constraints. If a group-I alloy was to be designed, we should have used a much closer range for Aluminum (Al) %weight. We have used the 0 to 6 range (%weight), but it can be noticed from [1] that for group-I alloys the range is 5 to 6. Given the latter, the genetic algorithm optimization might have resulted in a group-I alloy.

The K_a value for these new (designed) alloy composition is 0.90918058, which puts the superalloy in the category of fair according to Barrets' [1] classification in which the K_a values are ranked as,

$$K_a \leq 0.20 \quad \text{excellent}$$
$$0.20 \leq K_a \leq 0.50 \quad \text{good}$$
$$0.50 \leq K_a \leq 1.00 \quad \text{fair}$$

1.00 <=	K_a	<= 5.0	poor
5.00 <=	K_a		catastrophic

The lowest value of K_a obtained in the actual tests at 1100 C, for group-II alloys is 1.708 (U-700) [1]. Thus the soft computing methods have come up with a design that can meet the requirement of low the K_a values.

IV. Conclusions

We have applied the soft computing methods of neural networks and genetic algorithm to the of design of advanced superalloys. The key feature of this approach is the use of the neural network for modelling the material properties as functions of alloy chemistry and process parameters and the use of genetic algorithm for optimizing the function and thus obtaining a superalloy with low Ka values. The genetic algorithm used for optimization needs only the objective function values which are provided as the outputs of the neural network. To summarize the following results were obtained:

1) The trained neural network (R^2=86.56%) gives a better fit than the regression (R^2=84.43%).
2) The predicted value for NASAIR-100 alloy, is much better for the neural net model than the linear regression model (Table 2).
3) A new superalloy, of group-II, was designed using the genetic algorithm, with K_a value of 0.9091 at 1100 °C, which is classified as fair [1]. In test results used for modelling, none of this group superalloys had such a low K_a value.

Given different constraints these results could be most probably further improved.

Acknowledgment

The authors would like to thank Dr. C. A. Barret from NASA lewis Research Center for permission to use the superalloy data.

References

[1] C. A. Barret, "A Statistical Analysis of Elevated Temperature Gravimetric Cyclic Oxidation Data of 36 Ni- and Co- Superalloys Based on an Oxidation Attack Parameter," NASA Technical Memorandum, December 1992.

[2] K. J. Cios, G. Y. Baaklini and A. Vary, "Soft Computing in Design and Manufacturing of Advanced Materials" Journal of Engineering for Gas Turbines and Power, Vol. 117(1), pp. 161-165, 1995

[3] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley Publishing Company, Inc., Reading Ma., 1989.
[4] J. H. Holland, "Genetic Algorithms," Scientific American, pp. 66-72, Jul. 1992.
[5] B. E. Stuckman and E. E. Easom, "A Comparison of Bayesian/Sampling Global Optimization Techniques" IEEE Trans. on Systems, Man & Cybernetics, vol. 22, no. 5, pp 1024-1032, Sep/Oct 1992.
[6] V. Maniezzo, "Genetic Evolution of the Topology and Weight Distribution of Neural networks" IEEE Trans. on Neural Networks, vol. 5, no. 1, pp 39-53, Jan 1994.

Appendix

Neural network training results.

	Alloy	Temp	Ka Observed	Ka NN	Ka Regression
(1)	Alloy-625	1100	28.71441	33.12075	11.27800
(2)	Alloy-625	1150	36.42085	65.32808	17.99260
(3)	Alloy-718	1100	28.56603	30.54570	36.16710
(4)	Alloy-718	1150	43.39103	60.06204	69.82400
(5)	Astroloy	1100	3.23743	10.80936	9.13700
(6)	Astroloy	1150	61.72343	21.69202	21.93610
(7)	B-1900	1000	0.05310	0.05354	0.01870
(8)	B-1900	1100	0.19269	0.44463	0.31000
(9)	B-1900	1150	1.68384	1.66802	1.08980
(10)	B-1900-+-Hf	1100	0.72219	0.38940	0.32770
(11)	B-1900-+-Hf	1150	1.10053	1.87759	1.15220
(12)	IN-100	1093	28.49377	34.80566	1.86570
(13)	IN-100	1100	46.06277	39.31424	24.30670
(14)	IN-100	1150	97.48773	84.51817	76.63070
(15)	IN-713-LC	1100	0.71499	20.17901	0.94390
(16)	IN-713-LC	1150	1.67359	71.84557	2.66850
(17)	IN-738	1000	1.69805	5.20595	3.12460
(18)	IN-738	1100	29.32580	30.11619	19.59870
(19)	IN-738	1150	37.93149	59.08810	44.55700
(20)	IN-792	1100	22.54759	28.20979	19.20340
(21)	IN-792	1150	50.10717	66.91138	52.25930
(22)	IN-939	1100	32.58367	40.81313	30.14130
(23)	IN-939	1150	55.37961	64.90826	49.41480
(24)	MAR-M-200	1100	8.21296	20.95077	14.35090
(25)	MAR-M-200	1150	74.25060	53.29665	47.77800
(26)	MAR-M-200-+-Hf	1100	17.31210	26.54911	16.17680
(27)	MAR-M-200-+-Hf	1150	64.41692	74.50749	53.85680
(28)	MAR-M-211	1100	73.45983	17.29419	11.60070
(29)	MAR-M-211	1150	57.18736	44.17740	38.62180
(30)	MAR-M-246	1100	1.55292	3.21440	0.83760

:ion,
›any,
pp.
of
ans.
032,
ight
ıral
n
-

(31)	MAR-M-246	1150	18.07799	11.27847	2.50060
(32)	MAR-M-247	1000	0.05250	0.06792	2.50060
(33)	MAR-M-247	1100	0.50699	0.91254	0.04770
(34)	MAR-M-247	1150	4.98482	4.22766	0.77430
(35)	MAR-M-421	1100	9.53126	16.23865	2.69280
(36)	MAR-M-421	1150	34.93413	34.81770	8.63530
(37)	NASA-TRW-VIA	1100	0.32934	0.43451	19.84710
(38)	NASA-TRW-VIA	1150	1.59019	1.86423	0.35330
(39)	Nimonic-115	1000	0.40851	0.80131	1.37760
(40)	Nimonic-115	1150	1.64002	15.90560	0.40710
(41)	NX-188	1100	3.44588	3.38532	7.43090
(42)	NX-188	1150	8.21391	14.63356	2.28170
(43)	Rene-41	1150	33.14362	49.25496	12.40500
(44)	Rene-80	1100	37.40245	33.01795	38.79820
(45)	Rene-80	1150	60.76452	67.99076	20.00150
(46)	Rene-120	1100	6.85409	12.29986	50.70860
(47)	Rene-120	1150	14.91077	30.55272	8.85880
(48)	Rene-125	1100	3.02273	2.86913	24.49300
(49)	Rene-125	1150	9.78363	12.35521	2.06020
(50)	R-150-SX	1000	6.00136	5.29724	6.85800
(51)	R-150-SX	1100	45.00908	66.84979	2.84800
(52)	R-150-SX	1150	314.84732	151.56540	68.24000
(53)	TAZ-8A	1000	0.09700	0.07279	282.51901
(54)	TAZ-8A	1100	0.56735	0.70713	0.02520
(55)	TAZ-8A	1150	4.64408	2.87144	0.52440
(56)	TRW-R	1000	0.05600	0.03252	2.05340
(57)	TRW-R	1100	0.10650	0.26918	0.03230
(58)	TRW-R	1150	0.91201	1.19591	0.53650
(59)	TRW-1800	1100	0.73097	1.24753	1.88630
(60)	TRW-1800	1150	3.69020	3.55140	0.87460
(61)	U-520	1100	31.64828	16.21437	2.34160
(62)	U-520	1150	55.97576	32.47507	17.25930
(63)	U-700	1000	1.30707	0.97578	33.32080
(64)	U-700	1100	6.96226	6.64431	0.76570
(65)	U-700	1150	29.63467	15.18273	5.42470
(66)	U-710	1100	33.75592	26.89057	13.02350
(67)	U-710	1150	48.91026	48.23917	20.20680
(68)	U-720	1000	6.38851	5.16179	41.19590
(69)	U-720	1100	32.33329	23.74652	3.92420
(70)	U-720	1150	41.57671	43.54115	19.29180
(71)	Waspaloy	1000	4.99862	3.30446	39.33060
(72)	Waspaloy	1100	9.62941	18.38443	3.70670
(73)	Waspaloy	1150	28.89349	36.43763	15.17910
(74)	WAZ-20	1100	21.14707	32.38919	28.51700
(75)	WAZ-20	1150	89.21751	91.72762	15.08830
(76)	MAR-M-509	1100	25.42729	37.99707	82.03130
(77)	MAR-M-509	1150	49.77372	62.82031	25.66680
(78)	W-152	1093	47.03811	16.46076	38.77640
(79)	W-152	1100	45.28975	20.17901	16.11080
(80)	W-152	1150	120.57302	71.84557	54.95520
(81)	X-40	1100	35.57131	25.85235	95.14120
					24.45800