Appendix A

This is the documentation for the genetic algorithm program, as described in chapter 9.

Program MAP_GENETIC_ALGORITHM

Provenance of Source Code

The program was produced by

Iqbal Shah, Thomas Sourmail and H.K.D.H. Bhadeshia,
Phase Transformations Group,
Department of Materials Science and Metallurgy,
University of Cambridge,
Cambridge, U.K.

Application added: September 2002

Purpose

An application of the genetic algorithm (GA) for reaching a solution given a fitting function. This can in theory be applied to any problem, where a database of inputs and outputs has trained a neural network.

Specification

Language:	C
Product form:	Source code and executable files for UNIX/Linux machines.

Complete program.

Description

Neural networks can be used to perform complex regression with a large number of inputs [1,2]. The resulting non-linear models can capture their interactions, but we may wish to use the neural network backwards and identify sets of input variables resulting in a desired output value. However, the large numbers of variables and the non-linear character of the model render this difficult.

The problem is that of looking for the input set (x_i), which will give a desired output y, with:

$$y = f(x_1, x_2, x_3 .....x_n)$$

with the function f being non-linear. Genetic algorithms (GAs) is one possible method of solving such a problem. Goldberg described these algorithms as “search procedures based on the mechanics of natural selection and natural genetics” [50].

Once a target output value t is selected, the GA randomly generates sets of inputs called “chromosomes”, which exist in populations:

$$\mathrm{X}_i = \left[x_{i1}, x_{i2}, x_{i3}, ...x_{iz}\right]$$

Each chromosome X_i is composed of a set of variables $\{x_{i1...}\ x_{iz}\}$ called "genes". These sets are then ranked according to a fitness factor, which describes how well they perform. In the above example, a fitness factor F could be:

$$F = \frac{1}{(y_i - f_i)^2}$$

where y_i is the desired output for an i^{th} set of inputs. Those that are close to the desired output are allowed to "breed" in order to generate another z sets of inputs, whereas the others are eliminated. In this way, it should be possible to evolve towards the correct set of inputs, hence demonstrating that the GA process is based on the concept of “survival of the fittest”.

It is recommended that Goldberg [3] is read before proceeding with this program, in order to understand the terminology, as well as the effects and reasons for using different operators.

MAP provides a set of data files, which can be used with the GA program for finding different sets of inputs. The *ga_for_all.zip* file, which can be downloaded from here, contains the following files. A working version of a GA is also included to optimise the yield strength of austenitic stainless steels as a function of composition (wt%), temperature (K) and heat treatment temperature (K) (MAP_NEURAL_AUSTENITIC_YIELD):

iqbalcode.c - A C program used to run the genetic algorithm (GA).

treatout.c - A C program for un-normalising the output data files, but is kept within the /unnormalise subdirectory.

gareadme.txt/ gareadme.doc - A manual giving further details about running and using the GA program for altering different settings to optimise the process.

nn-input - Normalised inputs file (target and accuracy values for the GA).

norm_test.in - This is a text file, which contains the normalised input variables initialised by the GA to be fed into the neural network.

generate44/ generate55 - This is the executable file for the neural network program. generate44 runs on UNIX selntems and generate55 on the PC. It reads the normalised input data file, norm_test.in, and uses the weight files in subdirectory c. The results are written to the temporary output file _out.

nn-output_b - A file which contains the normalised inputs and outputs of the GA after the program has concluded, but is kept within the /unnormalise subdirectory.

score-output - A file that contains the scores of each chromosome after the program has concluded.

result A file, which contains the unnormalised values for inputs, outputs and error of each chromosome after unnormalising data in the normalised nn-output_b file.

These files must be placed in the same directory as the GA executable program and should not be changed (unless otherwise stated). As previously mentioned, a working neural network has been included. The following are concerned with the neural network files that work with the GA, but can be changed according to user requirements:

*_w*f* - The weights files for the different models.

**.lu* - Files containing information for calculating the size of the error bars for the different models.

spec1.t1 - An altered version of *spec.t1*, which is a dynamic file, created by *spec.ex/spec.exe*, which contains information about the module and the number of data items being supplied. It is read by the program *generate44/generate55.exe*.

outran.x - A normalised output file that was created when developing the model. It is accessed by *generate44* via *spec1.t1*.

MINMAX - Minimum and maximum values for the wt% of each element, the temperature (K) and heat treatment temperature (K) used in the original database. (i.e. the min and max values of each column in the original database before normalisation).

The file *gareadme.txt* contains a manual, which provides a detailed set of instructions for downloading and running the GA program. A summary of the information in this manual is given here.

References

1. D.J.C. MacKay, 1997, *Mathematical Modelling of Weld Phenomena 3*, eds. H. Cerjak & H.K.D.H. Bhadeshia, Inst. of Materials, London, pp 359.
2. D.J.C MacKay's website at http://wol.ra.phy.cam.ac.uk/mackay/README.html#Source_code
3. D.E. Goldberg: Genetic Algorithms in Search, Optimisation and Machine Learning. Addison Wesley (1989).

Parameters

Input parameters

To specify the target value and accuracy desired, *nn-input* must be amended.

Row 1	Normalised target value
Row 2	Accuracy (in decimal e.g. 0.1 for 10%)

The normalised target value relates to a real value, according to the MINMAX file used.

To initiate the GA search, the inputs are randomly generated and placed in *norm_test.in*. It should be noted that each chromosome generally relates to a different steel composition, but this could change over the course of optimisation. For the current yield strength model, the composition, temperature and heat treatment temperature, totalling 19 data items, are specified:

Gene number	**Variable**
1	Normalised Cr wt.%

2	Normalised Ni wt.%
3	Normalised Mo wt.%
4	Normalised Mn wt.%
5	Normalised Si wt.%
6	Normalised Nb wt.%
7	Normalised Ti wt.%
8	Normalised V wt.%
9	Normalised Cu wt.%
10	Normalised N wt.%
11	Normalised C wt.%
12	Normalised B wt.%
13	Normalised P wt.%
14	Normalised S wt.%
15	Normalised Co wt.%
16	Normalised Al wt.%
17	Normalised stoichometric addition ratio of Nb and Ti
18	Normalised heat treatment temperature / K
19	Normalised test temperature / K

Each input is normalised using the equation:

normalised value = (value - min)/(max - min) - 0.5

where the values for min and max are defined as follows:

Gene number	**Variable**	**Min**	**Max**
1	Cr wt.%	15.90	21.06
2	Ni wt.%	8.40	34.45
3	Mo wt.%	0.00	2.91
4	Mn wt.%	0.61	1.82
5	Si wt.%	0.00	1.15

6	Nb wt.%	0.00	0.95
7	Ti wt.%	0.00	0.56
8	V wt.%	0.00	0.06
9	Cu wt.%	0.00	0.35
10	N wt.%	0.00	0.08
11	C wt.%	0.01	0.12
12	B wt.%	0.00	0.02
13	P wt.%	0.00	0.04
14	S wt.%	0.00	0.05
15	Co wt.%	0.00	0.54
16	Al wt.%	0.00	0.52
17	Stoichometric addition ratio of Nb and Ti	0.00	3.06
18	Heat treatment temperature / K	1279	1473
19	Test temperature / K	293	1273

Output parameters

Two output files are produced by the GA program:

nn-output_b	-	A file which contains the normalised inputs and outputs of the GA after the program has concluded, but is kept within the /unnormalise subdirectory.
score-output	-	A file that contains the scores of each chromosome after the program has concluded.

score-output simply prints out the scores for each chromosome.

nn-output_b contains the inputs (composition, temperature, heat treatment temperature for yield strength), prediction and (prediction + error). This is done for the best chromosomes within all populations:

Column 1-19	-	The normalised predicted inputs.
Column 20	-	The normalised predicted output
Column 21	-	A value for the error, which includes the experimental noise of the database, an estimation of the uncertainty in the prediction and the test error. However, this is added to the prediction value so that the MINMAX file can easily handle the

		value.

The normalised values for the temperatures in all columns must be un-normalised using the equation:

actual value = (normalised value + 0.5)*(max - min) + min

The C program, *treatout.c*, is used to translate the output files to produce the actual values of inputs and outputs which are written to *result*.

Error Indicators

None.

Accuracy

See:

Input parameters, output parameters.

Further Comments

None.

Example

1. Program text

```
Complete program
```

2. Program data

The input file is:

```
0.4
0.1
```

3. Program results

The output file *nn-output_b*, which contains the normalised values for yield strength model including composition wt %, heat treatment temperature (HTT in K) and test temperature (TST in K):

Cr	**Ni**	**Mo**	**Mn**	**Si**	**Nb**	**Ti**	**V**
-0.50098	1.02233	0.847732	0.956254	1.071616	-0.45782	0.511222	-0.11883
-0.16101	0.256176	-0.03355	0.183018	0.667754	0.58004	-0.18862	-0.38744
-0.96214	0.485774	0.956256	-0.55776	0.768124	-0.01484	0.192882	0.44606
-0.54415	0.593968	0.41279	0.437972	0.003138	0.114784	-0.09118	0.200396
0.509944	0.885392	0.756296	0.02447	1.043348	-0.22878	0.298126	0.316028
-0.92847	0.001604	0.835018	-0.91864	0.415142	0.670886	-0.1629	0.095364

Cu	**N**	**C**	**B**	**P**	**S**	**Co**	**Al**
-0.06951	-0.14572	-0.1708	0.410288	-0.58563	0.81566	0.770394	0.469506
0.770666	-0.18601	-0.52197	0.280574	-0.20051	0.57496	0.061582	0.535408
0.590836	1.031872	-0.39492	0.117738	-0.4555	-0.39937	-0.01293	0.157058
-0.04015	0.34606	0.453118	0.162696	0.045304	0.042122	0.436606	-0.43415
-0.22158	0.446884	0.769066	0.596744	-0.61458	0.5797	0.105568	0.751496
1.077814	0.245742	0.827036	-0.28689	0.102666	-0.22555	-0.25317	0.025078

Ratio	**HTT**	**TST**	**Pred**	**Pred + Err**
-0.35223	0.88121	-0.08682	0.38812	1.503626
0.261894	0.853972	1.071118	-0.64601	0.623897
-0.10509	-0.33942	0.971052	-0.61658	1.0188
0.207308	-0.95481	0.586838	-0.77732	0.956581
0.20268	-0.16369	0.380716	0.047217	1.08183
0.573096	-0.18247	-0.57423	0.742344	1.859643

These are then run with *treatout.c*, where the normalised values are converted into the actual values for composition wt %, heat treatment temperature (K) and test temperature (K) are output to *result*:

Cr	**Ni**	**Mo**	**Mn**	**Si**	**Nb**	**Ti**	**V**
14.80976	10.47962	2.553839	1.308281	1.821103	0.01	0.04	0
14.80976	10.47962	4.365	1.308281	1.821103	0.01	0.04	0
17.52873	5.464738	1.162568	2.425	0.322959	0.01	0.04	0
18.48	21.425	4.365	1.215	0.575	0.01	0.04	0
13.58437	4.390227	4.365	2.425	1.470473	0.01	0.04	0
13.58437	4.390227	1.086076	2.425	1.725	0.01	0.04	0
18.8676	9.704428	3.859469	2.264457	0.857313	0.01	0.04	0
18.8676	9.704428	4.365	2.264457	0.857313	0.01	0.04	0

Cu	**N**	**C**	**B**	**P**	**S**	**Co**	**Al**
0.17	0.005871	0.016949	0.00459	0.025	0.013	0	0
0.17	0.005871	0.016949	0.00459	0.025	0.013	0	0
0.17	0.003882	0.026032	0.007638	0.025	0.013	0	0
0.17	0.0405	0.066	0.0075	0.025	0.013	0	0
0.17	0.035633	0.02123	0.007806	0.025	0.013	0	0
0.17	0.035633	0.02123	0.0075	0.025	0.013	0	0
0.17	0.057779	0.031085	0.004425	0.025	0.013	0	0
0.17	0.057779	0.031085	0.004425	0.025	0.013	0	0

Ratio	**HTT**	**TST**	**Pred**	**Pred + Err**
0.880904	1400	298	214.0423	100.2677
0.880904	1400	298	214.9005	103.1479
0.699653	1400	298	289.0625	158.9392
1.53125	1400	298	392.0956	248.0149
0.138988	1400	298	271.873	126.3189
1.53125	1400	298	252.3648	204.8545
0.98207	1400	298	378.1229	205.0144
0.98207	1400	298	379.159	205.091

nb. The results in the above table are for the best performing chromosomes for each population.

Auxiliary Routines

None.

Keywords

austenitic stainless steel, genetic algorithm, yield strength, neural network.