Directed Multiple Objective Search of Design Spaces Using Genetic Algorithms and Neural Networks

David S. Todd
d.s.todd@ncl.ac.uk
EPSRC Engineering Design Centre

Pratyush Sen
pratyush.sen@ncl.ac.uk
Department of Marine Technology

Armstrong Building, The University,
Newcastle-upon-Tyne, NE1 7RU.
United Kingdom.

Abstract

When performing search using Multiple Objective Genetic Algorithms (MOGAs) the aim is to maintain a diverse range of solutions whilst trying to converge these solutions onto the trade-off surface. In this paper a method to focus a MOGA search onto specific areas of this trade-off surface is investigated. Using interaction with a designer or decision maker his general preferences can be captured during search and modelled using an Artificial Neural Network (ANN). This allows the designer to direct the search of the design space into the regions of most benefit.

1 BACKGROUND

Design is a process of divergence and convergence. During the initial design stages diversity is most important to explore as many solutions as possible. Convergence is then necessary in order to settle on a preferred design. Both of these terms are widely used in the GA community and the link between the action of the GA and the design process means that the GA is an extremely powerful design tool. When performing multiple objective optimisation during design the aims are similar. The designer would initially like to look at a variety of different designs and later would aim to find a solution with the best balance of objectives for his needs. This means that the design process, after initial diversity, must be directed towards a preferred solution.

The work carried out for this paper follows on from work previously published at ICGA 97 (Todd 1997). This involved the application of a multiobjective GA to a ship loading problem. The GA used in that paper utilised a second special population called the 'Pareto Population' which stores all non-dominated or Pareto solutions as they evolve over the generations. When this system was formulated and tested it became apparent that this feature could be utilised further to enhance the search process. Several static methods were developed but the method explained here provides an interactive method of directing multiple criteria GA search though the use of Artificial Neural Networks in order to concentrate effort on specific or preferred regions of the objective space.

2 DESIGNER PREFERENCE

An essential element in any decision making process is obviously the decision maker. In most GA applications the decision maker/designer defines only the problem model. The GA then attempts to synthesise a range of solutions for that model and then the decision maker selects one. The decision maker may also use some form of tool (Sen 1998), after execution has finished, to select one or more alternatives from the solutions presented. Increasingly multiple criteria tools are attempting to obtain preferences and information from the designer. The designer wants to use these new methods to investigate the available solution space and his priority ordering over it more efficiently and possibly, in the case of GAs, adapt the parameters and/or constraints of the problem (Parmee 1998). By using this type of tool the designer can shape the design process in the light of his preferences and discard any design directions which are inappropriate or impractical.

The aim here is to investigate a way of allowing the decision maker to interactively adapt the multiobjective GA search. This will lead to the multiobjective GA concentrating its effort on those areas which are preferred by the decision maker. This is an important consideration in the case where evaluation times are high so that unnecessary computations can be avoided.

As mentioned earlier this method utilises a second 'Pareto Population' and encourages the growth of specific designs in the normal population by placing preference selected individuals back into the population. In order to select these individuals a preference model has to be generated, modified and maintained. The approach explained in this paper uses an Artificial Neural Network to model the preference surface and to select preferred pareto individuals for placement back into the normal population. Thus it differs from previous GA/ANN work (e.g. Grierson 1996, Bull 1997) which has focused on training neural networks to generate a single objective fitness function, based on data from an information model or an expensive evaluation routine, and then using a normal single objective GA on this model to find the maxima or minima on this surface.

3 NEURAL NETWORKS

Neural networks are a computational technique which mimic the computational abilities of biological systems. A neural network has three key characteristics:

i) It consists of a number of processing elements (neurons).
ii) Each neuron is connected to other elements through weighted links.
iii) The functionality is determined by modifying these weights during a learning phase.

There are many types of neural network architecture in the literature, however the back propagation neural network (Werbos 1974, Rummelhart 1986) is the most widely known and this is applied here to capture preferences. The back-propagation neural network is a multilayer Perceptron network with a non-linear transfer function within the neurons. The network is classified as supervised in that it is used in a two stage process. The first stage of this process is learning and the second stage is prediction. During the learning process the neural network is "taught" to recognise a given set of input and output conditions. The learning rule used is called back-propagation. Additionally the neurons use an enhanced transfer function, often a sigmoid, and they are usually arranged into three or more layers (Figure 1). Only feed forward connections are allowed and connections must be between adjacent layers.

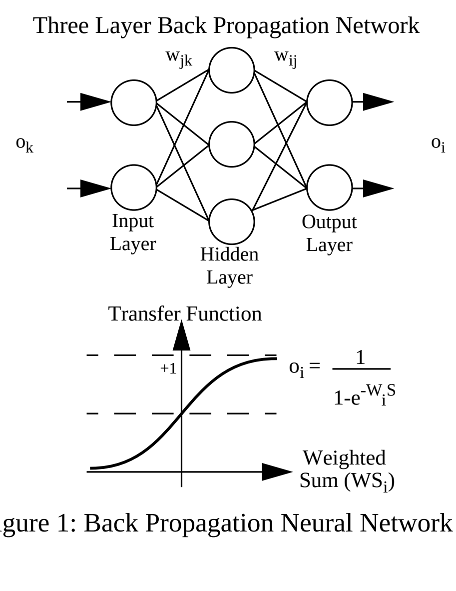

Figure 1: Back Propagation Neural Network

The back propagation neural network is taught to create a mapping between input and output patterns. During the training process the input-output pairs are known. However, instead of using a delta rule (difference between input and output) a squared error rule is used. The weights are then adjusted by small amounts to reduce the error across single neurons moving backwards from the output. The adjustment of weights via this 'back propagation' is continued until the squared error term is reduced below a certain threshold over the complete training data set. When this has been achieved the trained network can be used as a predictor giving an output pattern based on any given inputs.

4 THE MULTIPLE CRITERIA GENETIC ALGORITHM (MCGA)

The MCGA, the search algorithm, uses a standard Multiple Objective GA(MOGA) structure with several modifications. The normal processes of the MCGA are shown in Figures 2 and 3 and are explained more fully, along with a review of related MOGA work, elsewhere (Todd 1997).

The main difference between the MCGA and normal MOGAs is the introduction of a pareto population. This allows the MCGA to maintain a full set of currently non-dominated solutions including ones which have been lost from the population. The pareto population is updated every generation with the new non-dominated solutions of the current population. The population is then ranked and any duplicates or dominated solutions, which have been surpassed by newly evolved solutions, are removed. This population is utilised within the fitness sharing, selection and crossover procedures. However, in this paper the selection method is the most significant search driver. In the MCGA selection is a three step strategy:

Step 1 - The pareto individuals within the population are passed directly through to the mating pool.
Step 2 - In order to promote the generation of new pareto individuals and maintain diversity a random selection of strings from the pareto population are inserted into the mating pool.
Step 3 - The remainder of the mating pool is filled from the current population using roulette wheel selection based on the string fitness after sharing.

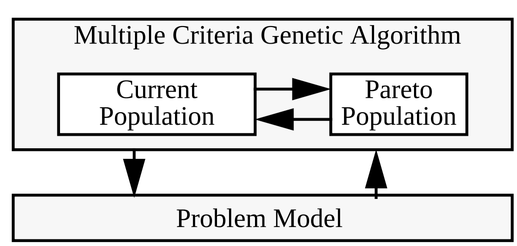

Figure 2 : System Structure

For the method reported in this paper the selection method has been modified with the emphasis being placed on Step 2 in which the random method above is replaced by a neural network selection procedure driven by user preference. The method of integration of GA and ANN is described in the next section along with the method of preference capture.

1. Create population.
2. Evaluate population on all criteria.
3. Rank population using dominance.
4. Update the Pareto Population.
5. Perform Fitness Conversion.
6. Perform Fitness Sharing.
7. Selection :
Step 1: Elitist strategy.
Step 2: Pareto Strategy.
Step 3: Roulette Wheel Strategy.
8. Perform Restricted Crossover(p=0.7).
9. Perform Mutation(p=0.01).
10. Evaluate population on all criteria.
11. Return to 3 unless end reached.
12. Output Designs

Figure 3 : The MCGA Algorithm

5 THE INTEGRATION OF MCGA AND NEURAL NETWORK

This technique uses a neural network to generate a preference surface based on preference data collected from the user. This surface is then used to select the preferred members of the Pareto Population for re-introduction into the normal population. This will promote further investigation into the preferred areas of the search space this in turn generating more individuals. The process is defined in Figure 3 in the form of a flow chart.

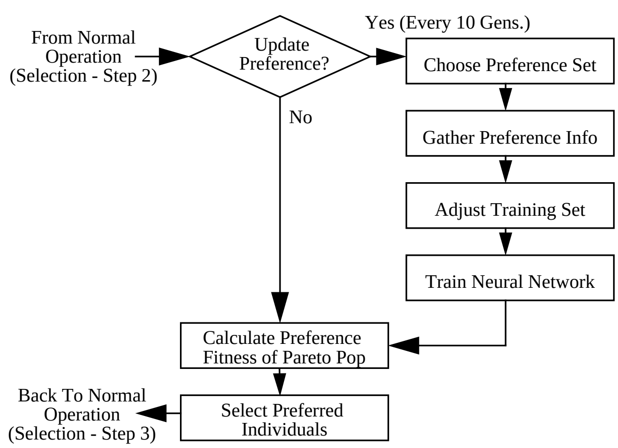

Figure 4 : MCGA / ANN Interaction

The preferencing process takes place at regular intervals through the MCGA process with the first preference occurring after ten generations. Prior to this the standard elitist/roulette procedure is used. When preferencing is to take place the preference set is automatically chosen. This preference set is a selection of ten individuals from the normal population. In the first set chosen this is done randomly. After this the system tries to select a broad cross section of individuals in terms of their neural preference score. The aim is to generate a more general picture of the surface. Two control individuals are also inserted into this set. These have the largest and smallest preference values for the current training set. The new set is displayed to the designer/decision maker in a random order. The system then gathers preference information by asking for a score between 0 and 1 for each member of the preference set. When complete the new scores for the control individuals are used to adjust the scores from the previous preference sets. Ten previous sets may be held giving a total training set size of 100 points when full. The adjusted training set is then used to train the neural network using the back propagation method. The newly formed preference surface can then be used to score the Pareto individuals between 0 and 1, one being the most preferred. This score is then used to select a set of individuals from the Pareto Population which are re-inserted into the population in order to promote search in the preferred regions of the search space.

The neural network was implemented and integrated into the MCGA. The method of picking individuals employs two additional parameters called *proportion* and *closeness*. Proportion is set between 0 and 1 and defines the percentage of individuals that are inserted into the population using the method. The greater this value the greater the effect of the selection. High values of proportion have the effect of greatly reducing the population diversity. The second variable, closeness, is also defined between 0 and 1 and relates to the closeness to the ideal preferred design. It is used to provide a threshold at which individuals will be selected to be re-inserted into the population. This threshold is set at 1-*closeness* based on the score given for a design by the preference surface.

6 SYSTEM TESTING

A population of 250 was run over 50 generations on the two criteria problem shown in Figure 5. The problem is defined over 2 variables x and y with values between 0-10, each with 10 bit encoding. The trade off surface is defined by two hills with offset centres. Each hill represents a criterion and is defined by the following exponentially decaying cos function, in this case centred at 5,5:

$$F(x,y)=\cos(((x-5)^2+(y-5)^2)/10)*1/\exp(((x-5)^2+(y-5)^2)/10)*8+1$$

If drawn on the same variable axes the pareto points fall along the line drawn between the centres of the two hills. The aim of the search is to find points along or very near this line. If a third offset hill was added the pareto points would lie in a area bounded by lines drawn between the peaks of the three hills. The tests were run initially with the standard MCGA and then with the MCGA/ANN with proportion=0.5 (50% of population chosen by designer preference) and closeness=0.2 (Individuals with preference scores greater than 1 - 0.2 = 0.8 will be selected).

The neural network layout used was a 3 layer system with 2 input, 5 hidden and 1 output neurons. The inputs take in the two criteria values. They are processed through the hidden layer via w_{jk} and through w_{ij} to the output layer. The number in the hidden layer was chosen as 5 arbitrarily and could be increased or decreased if required.

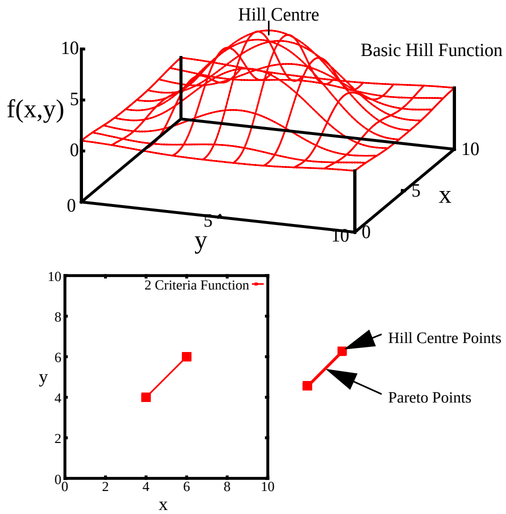

Figure 5 : Two Criteria Problem

After 10 generations of search the decision maker is presented with a table of randomly chosen designs and their associated criteria scores. An example of this is shown in Figure 6. The table shows numbered designs with their associated criteria values.

The decision maker is asked to score the designs relative to each other and not to any previous sets. He can also score two designs the same if he so wishes. In this test it was decided that ideally, designs with a ratio criterion 0:criterion 1 of around 9:5, and both values as large as possible, were preferred. In practice the judgement criteria would be much more involved and complex and the decision maker would have to base his judgements on heuristics and design experience. Here the design set was scored as shown in Figure 8. Accuracy is not particularly important; for example the user doesn't have to specify more that a single decimal place. This is due to the fact that the ANN is fitting a surface and will smooth out any minor flaws. However, consistency does play a big part in judgement and the decision maker should be careful not to score two similar designs with vastly different scores as this will cause problems during curve fitting.

From the preference set in Figure 7, design 8 is clearly seen as the best, followed by design 1 and then design 3. Most of the other designs were poor and given low scores. When the scores have been obtained, the code enters the neural network training phase. The generated preference surface can then be used to score the Pareto Population. This score is the preferred fitness score for each string. This fitness is then used to re-insert the most preferred individuals into the population with the aim of promoting more individuals in this area. The population is then filled up using roulette wheel selection on the current members of the population. The search then continues for another 10 generations as normal.

```
*** Interactive Preference ***

Design 0 :        2.52965      7.46026
Design 1 :        7.07147      5.87281
Design 2 :       0.478745      5.34876
Design 3 :        7.48098       7.0025
Design 4 :        3.81956      7.80841
Design 5 :         4.0185     0.469328
Design 6 :          0.619      6.84764
Design 7 :        3.10982      8.14307
Design 8 :        8.42858      5.96905
Design 9 :       0.465795      5.86715

Please Score the above Designs on scale of 0-1 :
(Base your judgements on this set of designs only, not any
previous ones)
```

Figure 6 : Design Judgment List

```
Score for Design 0 :   0.0
Score for Design 1 :   0.8
Score for Design 2 :   0.0
Score for Design 3 :   0.7
Score for Design 4 :   0.1
Score for Design 5 :   0.1
Score for Design 6 :   0.0
Score for Design 7 :   0.1
Score for Design 8 :   1.0
Score for Design 9 :   0.0
Thank you - returning to processing
```

Figure 7: Preference Data Collection

Following this another preference set is chosen, this time with the aim of presenting a broader set of individuals based on their preferred values. This set is shown in Figure 8. It can be seen that two designs, design 6 (8.42858, 5.96905) and design 3 (0.478745, 5.34876) appeared in the previous preference set (Figure 6) as designs 8 and 2 respectively. These are the two control individuals used to scale the previous sets. The decision maker again gives his preference scores between 0 and 1. This decision maker's scoring process was repeated 5 times and the Pareto Solutions are displayed to the user with their preference scores, generated from the final round of preference scoring.

```
*** Interactive Preference ***

Design 0 :        8.34814      6.14678
Design 1 :       0.564551      1.40331
Design 2 :        4.39327      8.86197
Design 3 :       0.478745      5.34876
Design 4 :        7.71528      7.08282
Design 5 :        8.68926      5.28533
Design 6 :        8.42858      5.96905
Design 7 :        8.39937      2.91864
Design 8 :         8.3139      2.16067
Design 9 :         8.8383      4.80271

Please Score the above Designs on scale of 0-1 :
(Base your judgements on this set of designs only not any
previous ones)
```

Figure 8: 2nd Design Judgment List

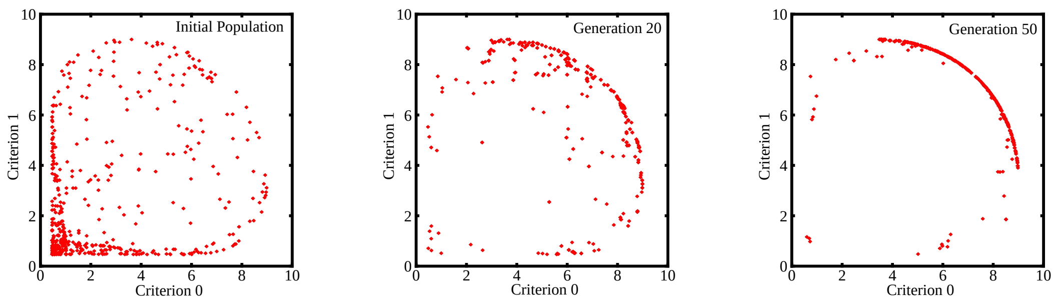

Figure 9 : Normal Population Distribution from Standard MCGA

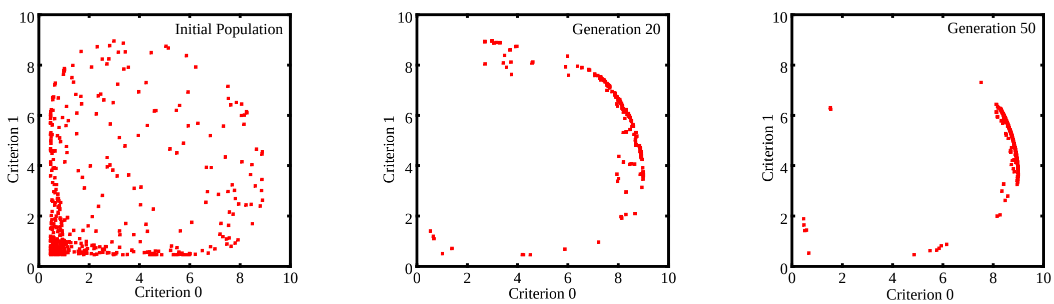

Figure 10 : Normal Population Distribution from MCGA with ANN Preference

Various graphs were plotted from the results. Figures 9 and 10 show the distribution of the normal population after initial, 20 and 50 generations for both the normal MCGA run and the MCGA/ANN run. As the preferences of the designer become apparent the search shifts most of its effort into generating new solutions in the preferred region. The initial population in Figure 10 shows a broad spread as expected. At generation 20, ten generations after the first preference session the population has redirected its effort towards the preferred region. After several more preference rounds the population is dictated by expressed preferences and the population clusters close to the 9,5 point. By changing the value of closeness the efforts of the MCGA can be focused even more precisely. It is clear that the bulk of the discovered pareto solutions in the MCGA/ANN run are towards the preferred values of criteria 0 and 1 (Figure 11). There are still some other solutions present in the pareto population due to the fact that the MCGA does not discard any Pareto solutions it finds. These solutions are likely to have been generated early in the search process.

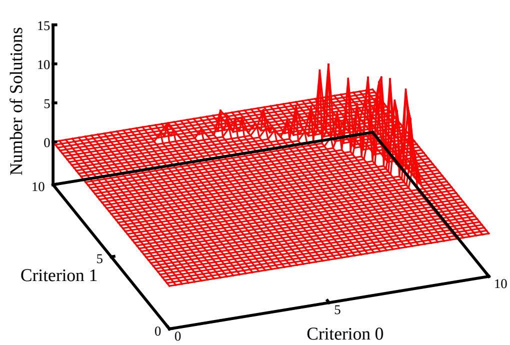

Figure 11 : Final Distribution of Pareto Population

clusters close to the 9,5 point. By changing the value of closeness the efforts of the MCGA can be focused even more precisely. It is clear that the bulk of the discovered pareto solutions in the MCGA/ANN run are towards the preferred values of criteria 0 and 1 (Figure 11). There are still some other solutions present in the pareto population due to the fact that the MCGA does not discard any Pareto solutions it finds. These solutions are likely to have been generated early in the search process.

An interesting result to look at is the development of the preference surface. Three of the preference surfaces for this run are shown in Figure 12. The initial surface is quite smooth as only a few points are specified. As more points are obtained the preference surface becomes more defined and extra features begin to emerge. In the final plot the surface is strongly structured with the preference being greatest in the correct area for this example. Away from this area the surface quickly falls away thus reducing the likelihood of selection.

Although the technique performs well when handled correctly there are several problems which were highlighted during the tests. Firstly, the neural network, because it is started from random points sometimes gets stuck in local minima during the back propagation procedure. This leads to preference surfaces which do not truly reflect the preferences of the decision maker. This was overcome by allowing the neural network to be re-run if the mean squared error was too great. This can be done several times until a low enough value is found.

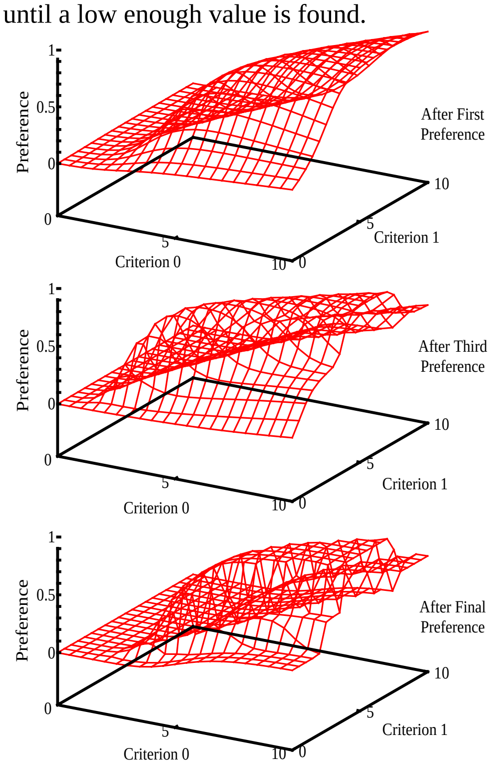

Figure 12 : Evolution of the Preference Surface

The biggest problem is checking for consistency. There is no consistency checking within the neural network. This means that if the same or similar points are given greatly different values of preference this can cause the MCGA to struggle when trying to fit surfaces. In such cases effects like the one mentioned above can occur and poor surfaces are generated. Some form of pre-processing is therefore required before back-propagation to increase the robustness of the surface fitting. If necessary the decision maker could be questioned about inconsistencies and asked to reconcile them. The designer may also want to relate the criteria scores back to specific design parameters. Presently the system only takes into account judgements on design performance not design characteristics. This is another area for further investigation.

7 CONCLUSIONS

This paper introduced a novel interactive method for controlling the evolution of multiple objective GA search. These methods allow the decision maker to tailor the direction of search performed to suit his own requirements.

The method described is used to concentrate search effort on the regions of the Pareto surface of greatest interest to the decision maker. The decision maker, with some knowledge of the form of the Pareto surface, is allowed to specify preferences that indicate the types of solutions desired. The MCGA will then re-direct its effort into finding solutions with the specified properties.

The interactive process described uses a back-propagation algorithm and adjusts the weights of an Artificial Neural Network to model preference information from the user. The system asks the decision maker to score the quality of solutions during search. These comments are then formed into a preference surface which can be used to direct search into regions of the search space which are more suited to the users' requirements.

Further work is still required to increase the stability of the method particularly when dealing with inconsistent judgements. More tests on more complex and higher order problems are also being carried out to improve the robustness and flexibility of the system.

Acknowledgements

This research was funded by the Engineering and Physical Science Research Council (EPSRC), UK.

Bibliography

L.Bull (1997), Model-based Evolutionary Computing: A Neural Network And Genetic Algorithm Architecture, *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp611-616.

D.E.Grierson (1996), Conceptual Design Using Emergent Computing Techniques, *Emergent Computing Methods in Design*: NATO ASI Series F: Computer Systems and Sciences, Vol.149, Springer-Verlag, pp150-161.

I.Parmee, D.Cvetkovic and E.Webb (1998), Multi-objective Optimisation and Preliminary Airframe Design, *Adaptive Computing In Design and Manufacture*, Springer-Verlag London, ISBN 3-540-76254-X.

D.Rumelhart G.E.Hinton and R.J.Williams (1986), Learning Representations by Back-Propagation Errors, *Nature*, Vol.232, No.9, pp533-566.

P.Sen and J.B.Yang (1998), *Multiple Criteria Decision Support in Engineering Design*, Springer-Verlag, London, ISBN 3-540-19932-2.

D.S.Todd and P.Sen (1997), A Multiple Criteria Genetic Algorithm for Containership Loading, *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan Kaufmann, ISBN 1-55860-487-1, pp674-681.

P.J.Werbos (1974), *Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences*, PhD Thesis, Havard University, Cambridge, MA, USA.