INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 99 MUNICH, AUGUST 24–26, 1999

GENETIC ALGORITHMS BASED SYSTEMS FOR CONCEPTUAL ENGINEERING DESIGN

Dragan Cvetković and Ian C. Parmee

Keywords: Multi-objective optimisation, conceptual design, genetic algorithms, preferences, scenarios

1 Introduction

In this paper we try to integrate methods of preferences and scenarios with Genetic Algorithms used to perform multi–objective optimisation. The goal is to make a system that will be able to work together with the designer during the conceptual design phase, where interaction and designer knowledge are sometimes more important than accuracy.

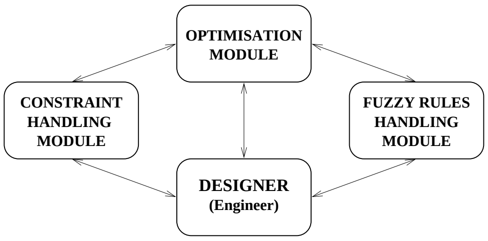

Figure 1: Schema of a preliminary design system.

The core of our computer / human system is presented in Figure 1 and it consist of 4 parts:

1. Genetic Algorithm based module for multi-objective optimisation;
2. Dynamical constraints handling module (scenarios etc.);
3. Fuzzy preference handling module (e.g.. for objective importance specification);
4. The designer.

The first 3 parts (modules) are described in the following sections, the fourth one cannot be formalised. We apply it to a British Aerospace (BAe) conceptual airframe design problem [4] which has at the moment 9 variables and 13 outputs. Optimisation represent a rather small part of the BAe problem. The fuzzy nature of initial design concepts, efficient exploration across the many different variants that the designer wishes to assess is of more interest. Evolutionary Computing should be able to support design exploration whilst also suggesting best direction [4]. Therefore, interaction with the designer (team) is very important. Our goal is to assist the designer in this preliminary phase of the design process more in the sense of MCDA (multiple criteria decision aid), then MCDM (multiple criteria decision making) [1]. Although being designed for the BAe problem, the techniques used are generic and could be easily integrated with other conceptual design problems.

2 Genetic algorithms and the optimisation module

The genetic algorithm used is based on the Breeder Genetic Algorithm [7]. It also utilises genetic operators suitable for real valued chromosome: arithmetic crossover, exponential mutation etc. As genetic algorithms in general are scalar function optimisers, we have adapted it to use techniques for multi–objective optimisation [4]. Techniques that we have used include

- Weighted sums based methods (WEI);
- Lexicographic sorting based methods (LEX);
- Pareto front based methods (with and without ranking) (PAR);
- Distributed VEGA [8] based method and its refinements (VMIX, VCOPY, ...).

Some of the results (on optimising y_4 and y_9, the two most conflicting objectives) are given in Table 1. Results are averaged over 50 runs and average value and standard deviation (σ) are presented. See [4] for more results and discussion.

Table 1: Results on optimising y_4 and y_9.

Method	y_4	$\sigma(y_4)$	y_9	$\sigma(y_9)$	Method	y_4	$\sigma(y_4)$	y_9	$\sigma(y_9)$
PAR	0.29	1.19	7811.8	106.24	LEX	0.17	0.80	7846.1	84.12
WEI	76.16	0.002	6198.9	0.29	VEGA	71.80	13.0	5886.4	230.30
VMIX	20.70	13.6	7355.5	159.80	VCOPY	20.30	16.0	7116.1	260.00

However, we shall always bear in mind that in this preliminary design phase accuracy of results (say, to the 4th or 5th digit after decimal point) is not as important as the capability to search different regions simultaneously.

3 Preferences and fuzzy rules

We will now address the problem of qualitative versus quantitative characterisation of the relative importance of objectives in a multi–objective optimisation framework. It is usually much easier for the designer to give qualitative marks to the objectives (i.e. "Objective A is much more important than objective B") than to set the weight w_A of objective A to 0.1 or to 0.09.

We use the method of fuzzy preferences [5] and induced preference order for our transformation. We introduce the following predicates [2]:

relation	intended meaning	relation	intended meaning
$\approx$	is equally important	$\prec$	is less important
$\ll$	is much less important	#	don't know or don't care
$\neg$	is not important	!	is important

which, together with the complementary relations $\succ$ and $\gg$, can help us build the relationship matrix R necessary for "words to numbers" transformation. For this transformation we employ, among other things, concepts of "leaving score" [5] and the induced order for the relation R.

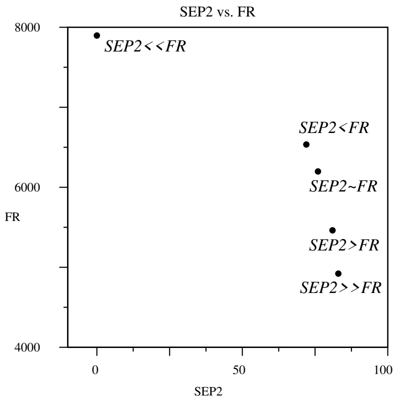

Figure 2: The influence of preference settings on BAe function.

We have integrated preferences in two genetic algorithm methods: one that uses weighted sums based multi–objective optimisation, and one that uses modified Pareto front method which can compute the objective weights. See [3] for more details. Figure 2 shows results that have been obtained optimising the BAe function using different preferences for y_4 (SEP2) and y_9 (FR).

4 Dynamical constraints and scenarios

The third module of our system from Figure 1 handles dynamical constraints and *scenarios*. Each scenario is a set of additional constraints / objectives that the designer can change, add and/or delete interactively. More formally, a scenario is represented as conjunctions of relations (constraints) in a fairly rich mathematical language (similar to the concept of clause in Prolog or in resolution based automated theorem proving [6]). Each scenario is a function of variables, objectives and (eventually) some additional parameters and returns a value between 0 and 1 as a percentage of the relations satisfied for given values of (its) inputs.

The concept behind the scenarios is that the designer can specify some conditions that are not part of the mathematical model (such as 'I would like to have $y_5 \in [0,4]$ or, if not possible (or feasible), then I would like to have $y_1 + y_3 > 100$ etc."). This also helps the design system to concentrate more on certain regions of design space. The additional advantage of scenarios is that they are dynamical and are being interpreted on–the–fly, without any need for recompiling the system or changing the program or mathematical model. So they can be added, modified or deleted on–line.

We have integrated scenarios with our design system. We can assign a different levels of importance to each scenario and we can calculate the value of a set of scenarios in three different ways:

1. Using weights or preferences for specifying scenario importance and use $\sum_i w_i \cdot v(S_i)$;
2. Calculating value as $\sum_i \mu^i \cdot v(S_i)$, where $\sum_i \mu^i = 1$ and $0 < \mu \leq 1$;
3. Using distributed island based GA where each island considers only one scenario.

The third approach is, in our opinion, the best and the most promising one as it enable us to employ software agents (or some other methods) that monitor levels of fulfilment of scenarios and which can signal back possibility or impossibility of satisfying of scenarios. In this case we don't have to worry about meeting as many as possible of potentially contradictory scenarios and we also don't have the problem of specifying (quantitatively or qualitatively) the importance (as in the first method) or the order (necessary for the second method) of scenarios.

5 Conclusion

In this paper we have presented a computer based system for preliminary engineering design and discussed three main parts of it. The optimisation part is genetic algorithms based. The fuzzy and preferences part translates designers linguistic statements into numbers and which deals with importance of objectives. The constraint handling module is used to handle dynamically additional constraints which the designer can add, modify and delete interactively and that could be used to refine the model or to direct search into certain regions. We have applied those methods to the BAe problem, and some results have been presented. Future work on this system will include a stronger interactive communication with the designer as well as some refinement on preference modelling.

References

[1] Carlsson, C., "Active Decision Support Systems", in *4th European Congress on Intelligent Techniques and Soft Computing EUFIT '96*, ELITE–Foundation, Aachen, September 1996, pp. 1279–1288.

[2] Cvetković, D. and Parmee, I. C., "Evolutionary Design and Multi–objective Optimisation", in *6th European Congress on Intelligent Techniques and Soft Computing EUFIT'98*, Aachen, Germany, September 1998, pp. 397–401.

[3] Cvetković, D. and Parmee, I. C., "Use of Preferences for GA–based Multi–objective Optimisation", in W. Banzhaf and J. D. et al., editors, *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, Orlando, Florida, USA, 1999.

[4] Cvetković, D., Parmee, I. C. and Webb, E., "Multi–objective Optimisation and Preliminary Airframe Design", in I. C. Parmee, editor, *Adaptive Computing in Design and Manufacture: The Integration of Evolutionary and Adaptive Computing Technologies with Product/System Design and Realisation*, Springer–Verlag, April 1998, pp. 255–267.

[5] Fodor, J. and Roubens, M., "Fuzzy Preference Modelling and Multicriteria Decision Support", Kluwer Academic Publishers, 1994.

[6] Loveland, D. W., "Automated Theorem Proving: A Logical Basis", North-Holland, Amsterdam, 1978.

[7] Mühlenbein, H. and Schlierkamp-Voosen, D., "Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization", *Evolutionary Computations*, **1**(1)1993, pp. 25–49.

[8] Schaffer, J. D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", in J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms*, Lawrance Erlbaum Associates, 1985, pp. 93–100.

Dragan Cvetković, PEDC, University of Plymouth, Drake Circus, Plymouth, PL4 8AA, UK, Phone +44 (0)1752 233 508, Fax +44 (0)1752 233 529, Email dcvetkovic@plymouth.ac.uk