Preliminary Airframe Design Using Co-Evolutionary Multiobjective Genetic Algorithms

Ian C. Parmee
Plymouth Engineering Design Centre,
University of Plymouth,
Drakes Circus,
Plymouth. PL4 8AA. UK.
Phone: +44 (0) 1752 233509
Fax: +44 (0) 1752 233529
e-mail: iparmee@plymouth.ac.uk

Andrew H. Watson
Plymouth Engineering Design Centre,
University of Plymouth,
Drakes Circus,
Plymouth. PL4 8AA. UK.
Phone: +44 (0) 1752 233508
Fax: +44 (0) 1752 233529
e-mail: awatson@soc.plym.ac.uk

Abstract

A novel multiobjective optimisation approach utilising a genetic algorithm (GA) for the preliminary design of airframes is introduced. Concurrent GA processes each optimise one objective related to the problem. The fitness measure for individuals within each GA is adjusted by comparing the values of the variable parameters of identified solutions relating to a single objective with those of the solutions of the other GA's. A penalty relating to the degree of diversity of their variable values as compared to those of the other GA processes is then imposed taking into consideration a generational parameter constraint map. Initial convergence upon individual objectives leads to overall convergence of all processes upon a single feasible design region. A sensitivity analysis to ensure that relative importance of a parameter is taken into account is also introduced. Design paths from the run are stored and can be used by the designer to explore not only the optimum solution provided by the method but also solutions which are biased towards each of the design objectives without further function calls to the design model.

1 INTRODUCTION

This paper presents research at the Plymouth Engineering Design Centre (PEDC) relating to the integration of evolutionary and adaptive computing with the design process [1]. The goal of preliminary design in this case is to identify optimal design regions relating to several objectives within the whole design space, utilising preliminary design models.

Most real world problems involve more than one objective function and it is generally the case that multiple objective functions are conflicting to some extent. Various methods have been employed for multiobjective optimisation including aggregating functions and Pareto approaches [2], that utilise GA search capabilities in addition to a small number of techniques that are entirely GA based. Some of these techniques provide single objective optimal solutions whilst others define an objective trade-off front comprising of many non-dominated solutions.

Aggregating functions include weighted sum methods where the user assigns each objective and the total fitness is the sum of all the weighted fitness values [3]. These methods will not produce a trade-off front unless many differing weight combinations are processed. Another aggregating function technique is to reduce the problem to that of minimising a single objective and consider all other objectives as constraints bound by some allowable levels ε. This technique is known as the ε-constraint or the trade-off method [4] and will produce an optimal solution but not a trade-off front. Other aggregating function methods include goal-attainment based on global criterion and penalty functions based on the ε-constraint method and weighting objectives method. As with the other aggregating function techniques they produce single optimal solutions.

Alternative approaches include the Vector Evaluated Genetic Algorithm (VEGA) [5] which uses sub-populations generated by performing proportional selection according to each objective in turn. A new generation is obtained by allowing crossover between these sub-populations. A Pareto front can be generated but because this approach selects individuals based on a single objective good trade-offs can be eliminated. Other non-Pareto approaches include lexicographic ordering [6], evolutionary strategies [7] and weighted sum methods with sharing [8].

The main Pareto-based approaches include Pareto-based fitness assignment [2] using non-dominated ranking and selection to move a population towards the Pareto front in a multiobjective problem. A set of non-dominated solutions are identified which are then assigned the highest rank and eliminated from further consideration. Another set of Pareto non-dominated strings are determined from the remaining population and are assigned the next highest rank. This process continues until the population is suitably ranked. The Multiple Objective GA (MOGA) [9] proposes a scheme in which the rank of a certain individual corresponds to the number of chromosomes in the current population by which it is dominated. The Non-dominated Sorting Genetic Algorithm (NSGA) [10] is based on several layers of classification of the individuals. A tournament selection scheme based on Pareto dominance leads to the Niched Pareto GA [11], where good performance depends upon a sharing factor and tournament selection size. No single solution is given when using Pareto methods and it is left to the designer to choose an appropriate design point or region within the identified Pareto front. Algorithms are therefore available which identify a single design solution that satisfies a number of objectives or others which produce a Pareto front. It is suggested that an ideal for multiobjective optimisation within preliminary design would be an algorithm that produces single objective high performance solutions, the Pareto front and, through designer interaction, an optimal solution to the problem at hand. The development of the method outlined here is progress towards the achievement of these goals.

2 THE AIRFRAME PRELIMINARY DESIGN MODEL

A computer model relating to the preliminary design of military aircraft has been developed in collaboration with British Aerospace plc (BAe) [13]. At present the miniCAPS model utilises 9 variable parameters and produces a total of 12 outputs relating to various objectives. The model includes a variety of disciplines including preliminary geometric definition, aerodynamic analysis, mass estimation and performance analysis. Input and output variables are listed in tables 1 and 2.

Input Parameter
0 Climb Mach Number
1 Cruise Height
2 Cruise Mach Number
3 Gross Wing Plan Area
4 Aspect Ratio
5 Wing Taper Ratio
6 Wing LE Sweep
7 Wing Tip/Chord Ratio

Table 1: Model Variables

Output Parameter
0 Take off Distance
1 Landing Speed
2 Specific Excess Power 1
3 Specific Excess Power 2
4 Sustained Turn R 1
5 Sustained Turn R 2
6 Attained Turn R 1
7 Attained Turn R2
8 Ferry Range
9 Mass Take-off
10 Wing Span
11 Chord/Fuselage length

Table 2: Model Outputs

The proposed distributed method utilises individual GA's for the optimisation of each objective. The problem is therefore reduced to a number of concurrent co-evolutionary tasks specific to the overall design domain. PVM software [14] controls the distributed architecture ensuring minimal clock time for these multiobjective problems.

3 FITNESS CALCULATION

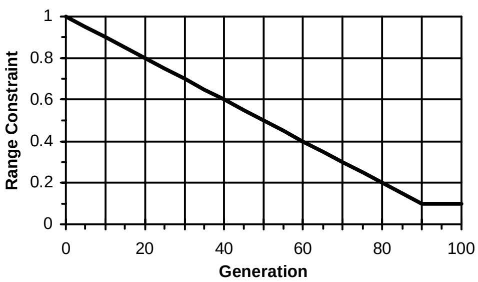
The fitness for each objective is normalised relative to the maximum and minimum values found during each GA run with constant adjustment as new upper and lower limits are identified. For each generation, solutions relating to each objective are compared with the best individual from the other GA populations. If a variable is outside a range defined by a range constraint map it is adjusted by a penalty function. Suppose we are optimising two objectives, the subsonic specific excess power (SEP1) and the ferry range (FR). Two GA's (S0 and S1) are initialised, S0 optimising SEP1 and S1 optimising FR. The process of calculating the fitness of population S0 is described in the following steps: -

1. Rank the fitness of population S0 using SEP1.
2. Rank the fitness of population S1 using the ferry range.
3. Starting with individual number 1 (the fittest), variable 1, compare the value with the equivalent variable of the best individual in S1. Return the difference between the two values divided by the total range defined for the variable being examined.
4. Compare the returned value against the value given by the range constraint map for the generation number.
5. If the returned value is greater than the constraint map value, apply a fitness penalty to individual 1.
6. Repeat steps 3-5 for all variables in individual 1.
7. Repeat steps 3-6 for all individuals in S0.

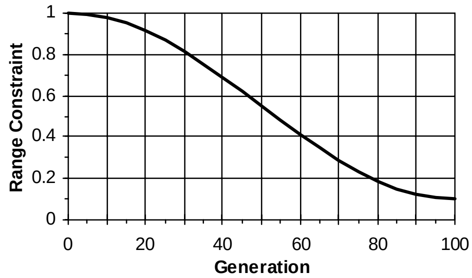
Note that the process is repeated for all individuals in population S1, which are compared with the best individual in S0.

4 THE RANGE CONSTRAINT MAP

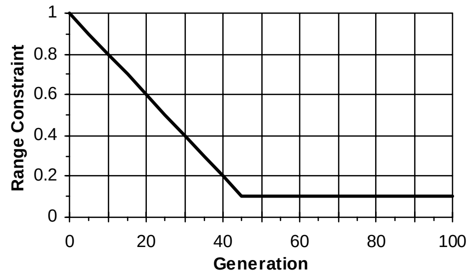
The range constraint map has to fulfil three objectives. Initially the map must allow each GA to produce an optimal solution based on its own specified objective. This is achieved by setting the value of the map to 1.0, allowing each GA to use the whole range for each variable. As the run progresses the map, through inflicted penalties, increasingly reduces variable diversity to draw all concurrent GA searches from their separate objectives towards a single optimal design region where all objectives are best satisfied. The constraint maps include a linear decrease in range constraint and a range constraint reduction based on a sine curve. The map must also allow some difference in variable values for each GA towards the end of a run to provide space within which the method can search for an overall optimal solution. This is achieved by setting a minimum value for the range constraint. The number of generations allocated to this final phase of exploration is tested using 2 values i.e. 10% and 50% of the maximum generations. This produces the four maps presented in figure 1. Note that the minimum value for the maps is set to 0.1 (10% of the variable range).



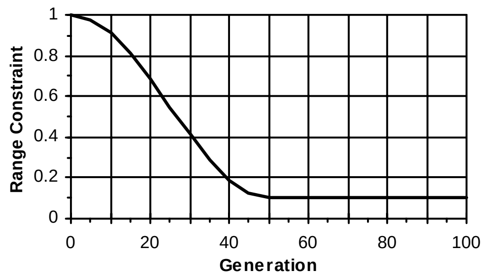
(a)



(b)



(c)



(d)

Figure 1: Various constraint maps used for the initial two objective experiments, (a) linear ramp map, (b) sine map, (c) half linear ramp map and (d) half sine map.

5 SENSITIVITY ANALYSIS

All variable parameters are assigned equal importance when assessing constraint map penalties. However, in most real design situations variables will have differing degrees of influence upon any given objective. Analysis is required therefore to determine which variables have the greatest bearing on each objective. An on-line sensitivity analysis which ranks the variables according to their influence upon each objective is introduced. This design sensitivity ranking is then used to adjust the fitness of each solution to ensure that the values of the most influential variables are within the range defined by the constraint map. Solutions are assigned the highest fitness penalty where their most influential variables lie outside of the current constraint map range. This ensures that subsequent populations contain high levels of feasible solutions in terms of the most influential variables and relatively redundant variables have little or no effect on overall solution fitness.

A sensitivity analysis method is required that has minimal computational overheads and provides an independent measure for each input parameter. Various methods available include:

- One-Factor Experiment - The one-factor experiment evaluates the effect of one variable parameter on performance while holding all others constant. If there is an interaction of the factor studied with some other factor then this interaction cannot be observed.
- Several Factors, One At A Time - The main limitation of several factors, one at a time, is that no interaction among the factors studied can be observed.
- Several Factors, All At the Same Time - This situation makes separation of any of the main factor effects impossible, as well as no observation of interactions.
- Full-Factorial Experiment - A full-factorial experiment is orthogonal, orthogonality means that factors can be evaluated independently of one another; the effect of one factor does not influence the effect of another. If a full-factorial experiment is used, there is a minimum of 2^f possible combinations that must be tested (f is the number of factors with 2 values for each factor).
- Fractional-Factorial Experiments (FFE's) - Use only a portion of the total possible combinations e.g. 1/2 FFE, 1/4 FFE, and a 1/8 FFE. Certain treatment conditions are use to maintain orthogonality among the various factors.
- Taguchi methods [12] use a family of orthogonal FFE matrices (orthagonal arrays, OA). The method incorporates a process for generating data that utilises a mathematically derived matrix to methodically gather and evaluate the effect of numerous parameters on a response variable.

The Taguchi method has been selected to determine the sensitivity of each input as interaction can be taken into account to some extent whilst incurring minimal computational cost. A detailed description of the Taguchi method is given in [12].

In most practical cases, once the number of design parameters and the number of settings per design parameter are determined, the task of finding a suitable orthogonal array is easily reduced to selecting an already-constructed table [12]. As the model has 9 variables 3 levels are chosen for each input so the L27 OA is used for the sensitivity experiment, this requires 27 evaluations for each sensitivity experiment. The fitness penalty is scaled between 0.5 for the most sensitive variable and 0.0 for the least sensitive.

6 RESULTS

In order to test the method initially two objectives are chosen which are known to be highly conflicting, i.e. subsonic Specific Excess Power (SEP1) and Ferry Range (FR). Each GA process (labelled S0 and S1) initially has a population size of 100 using a 16 bit binary encoding for each variable. The crossover rate is set to 0.6 and the mutation rate is 0.01 (1/population size). The reproduction method used is roulette wheel selection with one elite individual and a total of 100 generations being processed, the fitness penalty is set to 0.5. The Taguchi sensitivity analysis is not included in this initial experiment. Figures 2 and 3 show the average results obtained over 25 runs.

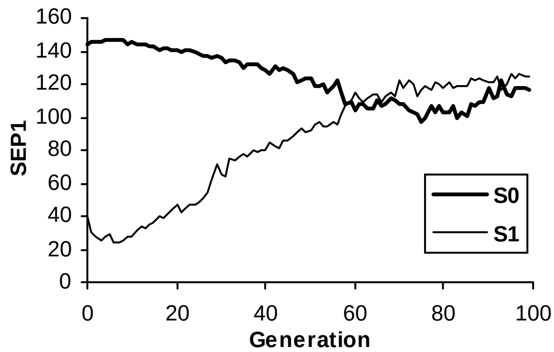


Figure 2: SEP1 vs. Generations

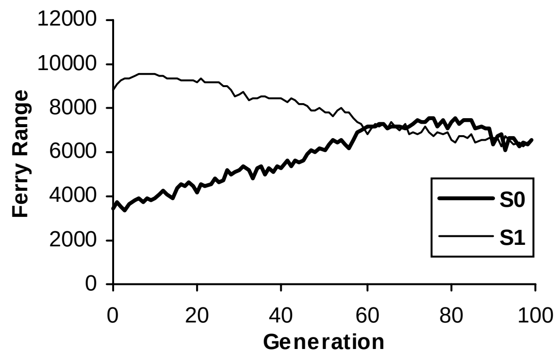


Figure 3: Ferry Range vs. Generations

Figure 2 shows that S0 (the GA optimising SEP1) produces near optimal solutions at the start of the run but as the run progresses this decreases while the SEP1 value of S1 (optimising Ferry Range) increases. This effect is also shown in figure 3 which shows the ferry range reducing for S1 and increasing for S0 to a common design region. This illustrates how both S0 and S1 converge on a feasible region of the design space where high performance solutions best satisfying both objectives are prevalent. In order to assess the robustness of the technique the standard deviations of the fittest individuals of S0 and S1 have been calculated over 25 runs and are shown in figures 4 & 5 for SEP1and ferry range respectively.

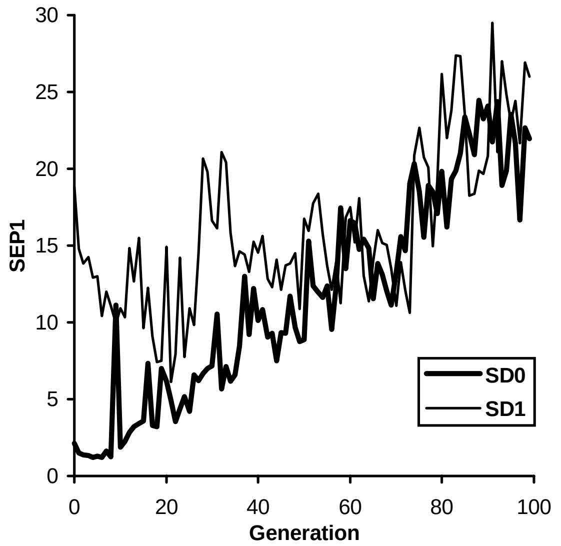


Figure 4: Standard Deviation For SEP1

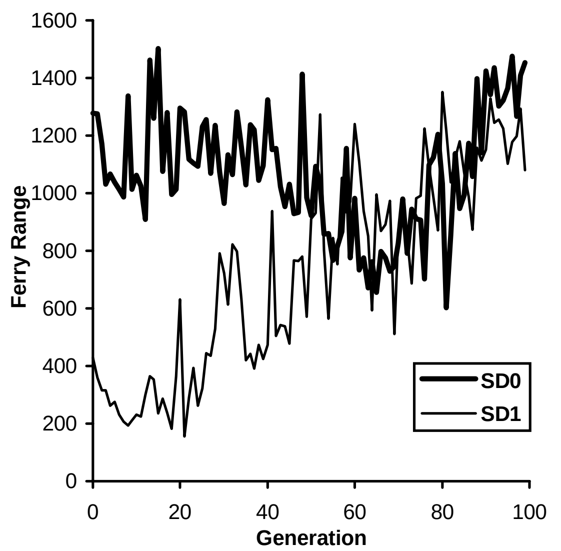


Figure 5: Standard Deviation for Ferry Range

The best individuals from each generation are saved and the averaged results are shown in figure 6. The known Pareto front is also shown in figure 6 for comparison.

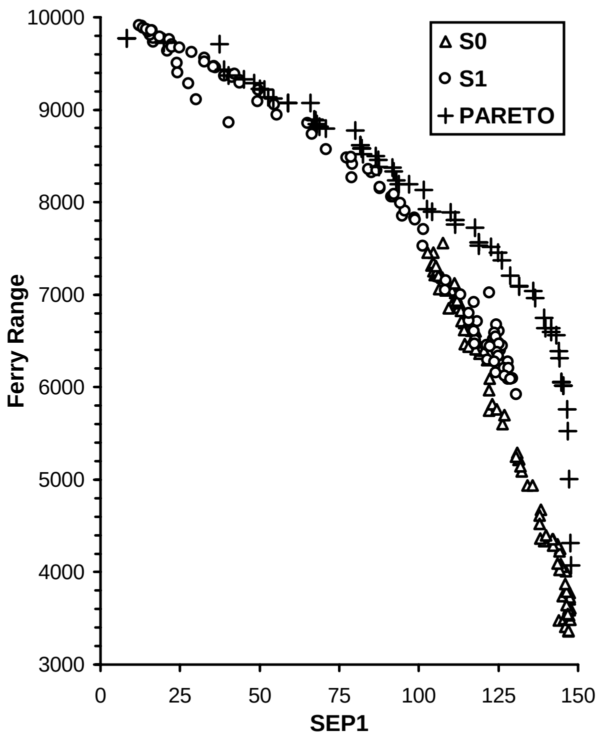


Figure 6: The Pareto Front and GA Results

The initial generations seek optimal solutions for their particular objective and as the constraint map starts to restrict the variables of each GA the populations attempt to traverse the Pareto front before converging upon a common region. Further testing for other constraint maps, various population sizes and inclusion of the Taguchi analysis produced the end of run results presented in table 3.

Initial runs using a population size of 100 are inconclusive in determining the best type of constraint map to use. The algorithm is to be used for preliminary design where the identification of a feasible design region is the primary objective. Analysis of the paths described by each map showed that the linear map produced results closest to the known Pareto front. The results do show that although the method is effective with smaller population sizes the design paths tend to move further away from the Pareto front. The Taguchi analysis was tested with the linear constraint map but for two objectives the results show no significant advantage in using the additional fitness ranking based on sensitivity of the objectives to individual variables.

Constraint map	Taguchi	Pop. Size	S0 (optimising SEP1) SEP1	SD SEP1	FR	SD FR	S1 (optimising Ferry Range) SEP1	SD SEP1	FR	SD FR
Linear to 10%	no	100	116.8	21.9	6587.9	1453.8	125.2	26.0	6542.1	1079.0
Linear to 10%	yes	100	120.2	20.0	6344.6	1359.9	134.3	15.5	5647.2	926.7
Linear to 10%	no	50	108.4	22.5	6875.1	1307.4	119.4	23.0	6414.4	1191.6
Linear to 10%	yes	50	113.2	30.5	6210.7	1971.9	114.3	30.3	6649.5	1420.6
Linear to 10%	no	25	113.8	26.8	6442.1	1265.2	98.3	42.8	7065.1	1574.1
Linear to 10%	yes	25	104.8	23.6	6833.1	1153.8	110.9	27.0	6573.7	1442.5
Linear to 10%	no	10	106.1	27.1	6432.0	1583.9	91.4	27.0	6993.9	1379.2
Linear to 10%	yes	10	113.0	29.8	5890.4	1550.4	99.5	32.5	6609.4	1305.0
Linear half to 10%	no	100	131.3	11.6	5946.2	1294.5	131.1	33.1	6062.3	1272.7
Linear half to 10%	no	50	122.5	18.6	6272.4	1300.5	126.5	28.5	6155.7	1288.5
Linear half to 10%	no	25	102.0	30.4	6202.5	2006.8	19.6	18.6	9489.6	389.0
Linear half to 10%	no	10	107.6	23.7	6240.5	1412.0	98.6	28.6	6969.4	1091.3
Sine to 10%	no	100	117.8	22.2	6409.6	1496.9	128.9	24.9	6083.3	1242.9
Sine to 10%	no	50	109.9	29.2	6655.7	1633.3	117.7	35.4	6551.8	1389.7
Sine to 10%	no	25	120.7	21.9	6127.1	1373.1	99.5	41.5	6942.4	1481.6
Sine to 10%	no	10	93.9	30.7	6392.3	1721.1	25.4	19.9	9394.9	480.5

Table 3: Two Objective Optimisation Results

6.1 OPTIMISING THREE OBJECTIVES

The complexity of the design problem is now increased to three design objectives, these are :- SEP1, FR and subsonic Attained Turn Rate (ATR1) with associated GA's S0, S1 and S2. Figures 7 to 9 show the average results from 25 runs. Each GA has a population size of 100 with a crossover rate of 0.6 and a mutation rate of 0.01 (1/population size). The reproduction method used is roulette wheel selection and a total of 100 generations are evaluated. The constraint map used is the linear map to 10% and initially the Taguchi analysis is not included. The results are averaged over 25 runs.

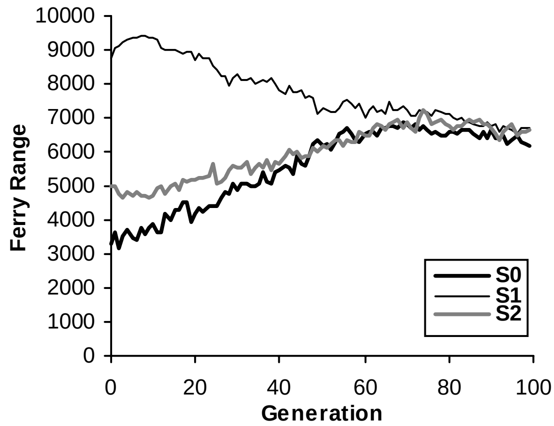


Figure 8: Ferry Range vs. Generations

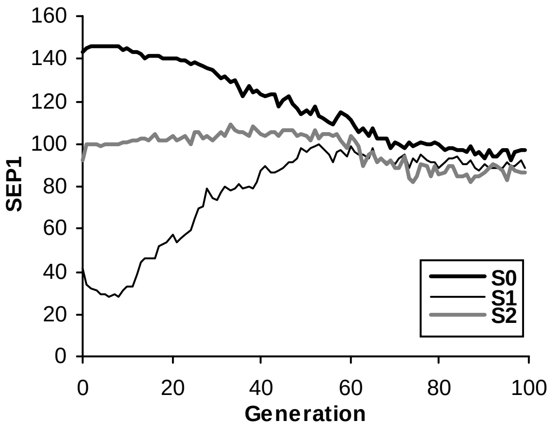


Figure 7: SEP1 vs. Generations

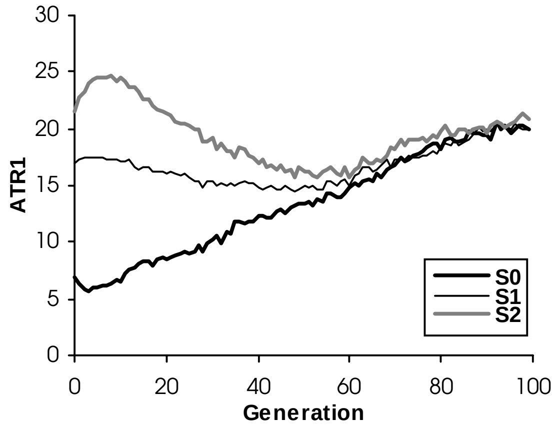


Figure 9: ATR1 vs. Generations

Figures 7,8 and 9 show each GA converging to a single region. The best individual from each population can be plotted to show the evolution of the design from initial single objective high performance regions to a single design region satisfying all objectives. The results from the three objective problem are plotted in figure 10.

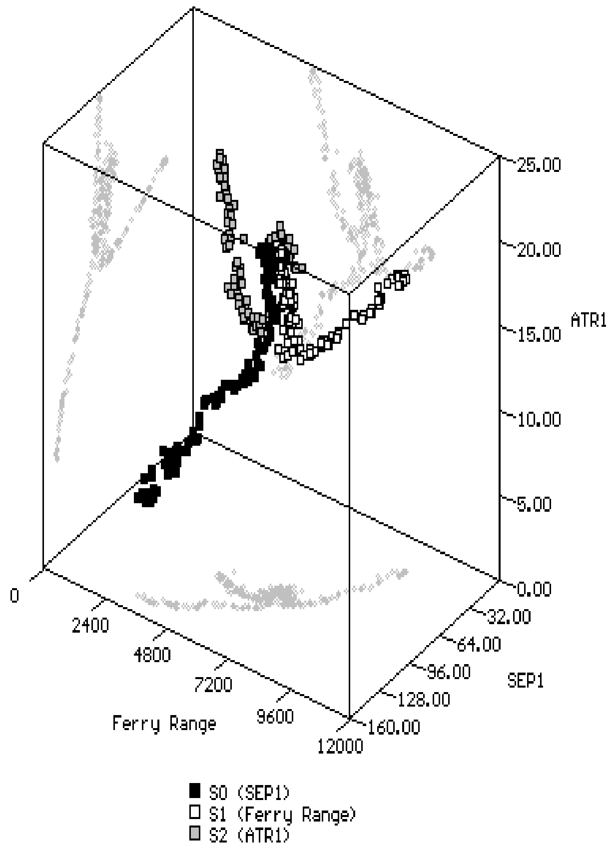


Figure 10: Three Objective Problem Design Paths

Figure 10 also shows projected shadows in the three objective planes. Each GA initially optimises its own objective at the start of the run (shown by the three end points) and then, as the run progresses, converges to a single design region.

The run is then repeated with the Taguchi analysis included. The standard deviations, of the 25 runs, for the ferry range with and without the Taguchi analysis are shown for comparison in figures 11 and 12 respectively.

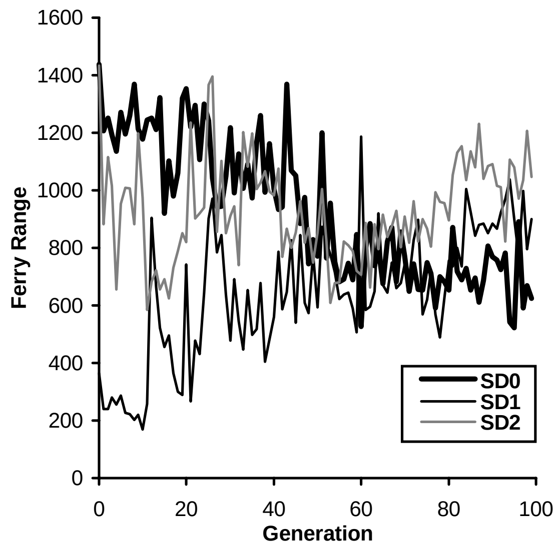


Figure 11: Standard Deviation For Ferry Range Without Taguchi Analysis

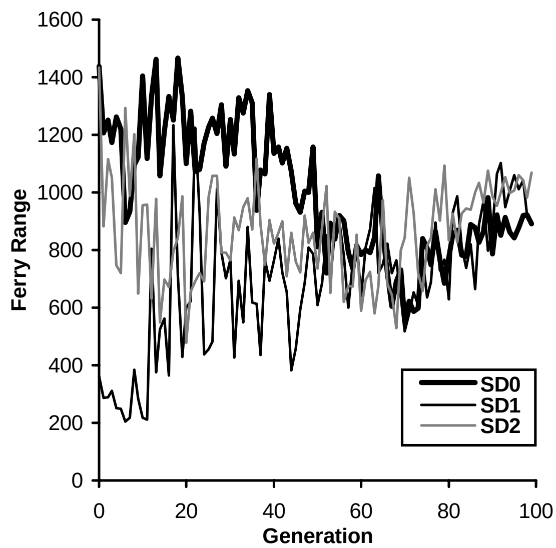


Figure 12: Standard Deviation For Ferry Range With Taguchi Analysis Included

Figure 12 shows that the Taguchi analysis reduces the variation of results from the runs to a common level towards the end of the run. The deviation at the end of the run being around 1000km (the maximum ferry range is ≈ 10,000km) which is consistent with the minimum 10% limit produced by the constraint map. The results are also consistent for the SEP1 and ATR1 objectives.

7 DISCUSSION AND CONCLUSIONS

The results presented show that the methods outlined can provide the design engineer with valuable information during a preliminary design study. The main advantages of the method are: -

- Local objective optimal solutions can be identified after the first few generations.
- Design paths are produced which trace the trade-off surface to some extent.
- A feasible design region for the problem is identified.
- Information about the important and redundant input parameters during the run is available.
- All information is produced with one run of the algorithm.

Local objective optimal solutions provide initial solutions to the problem, giving the engineer an idea of the maximum achievable results for these parameters when optimised alone. Runs which optimise two objectives can be shown to approximately traverse the Pareto front of the feasible design space from opposite ends of the Pareto front. The results using three objectives show the ability of the method to converge on an optimal solution by approximating a Pareto surface from three different starting points. The identification of sensitive parameters aids the search process by ensuring that the most important parameters have the greatest influence in the direction of the searches as it moves through the design space. The Taguchi analysis shows little effect with the two objective runs but does improve the results from the three objective experiments. This suggests that the on-line sensitivity analysis has a role to play as the number of objectives increases, and further work is investigating this utility.

The use of parallel GA's produces a linear decrease in running time for the method bringing the whole process within an acceptable time frame, and the results suggest that quicker less detailed runs can easily be achieved using smaller population sizes.

All runs were performed on a 6-processor Sun-Ultra Enterprise 4000 using Gnu C++, and Parallel Virtual Machine (PVM) Software [14], using 2nd author code.

It is unclear how this method will cope with discontinuous Pareto fronts and is currently under investigation.

Acknowledgements

The research has been carried out within the Plymouth Engineering Design Centre, at the University of Plymouth, with financial support from both the Engineering & Physical Science Research Council and British Aerospace Plc. under grant number **GR/L/31654**. We wish to thank both parties for their continuing support.

References

[1] Parmee, I.C. (1998). Exploring The Design Potential Of Evolutionary/Adaptive Search And Other Computational Intelligence Technologies. Proceedings of ACDM'98. Springer-Verlag.

[2] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimisation, and Machine Learning. Addison-Wesley, Reading, MA.

[3] Jakob, W. et al (1992). Application of genetic algorithms to task planning and learning. PPSN 2nd Workshop, Lecture Notes in Computer Science, Amsterdam, pp. 291-300, North-Holland Publishing Company.

[4] Rizel, B.J. et al (1994). Using Genetic Algorithms To Solve A Multiple Objective Groundwater Pollution Containment Problem. Water Resources Research, vol. 30, pp. 1589-1603.

[5] Schaffer, J.D. (1984). Some Experiments In Machine Learning Using Vector Evaluated Genetic Algorithms. (Doctoral Dissertation). Nashville, TN: Vanderbilt University.

[6] Fonseca, C.M. & Flemming, P.J. (1995). An Overview of Evolutionary Algorithms In Multiobjective Optimisation. Evolutionary Computing, vol 3, pp. 1-16.

[7] Kursawe, F. (1991). A Varient of Evolution Strategies For Vector Optimisation. PPSN 1st Workshop, Lecture Notes in Computer Science, Amsterdam, pp. 193-197, Springer-Verlag.

[8] Hajela, P. & Lin, C.Y. (1992). Genetic Search Strategies in Multicriterion Optimal Design. Structural Optimisation, vol. 4, pp. 99-107.

[9] Fonseca, C.M. & Flemming, P.J. (1993). Genetic Algorithms For Multiobjective Optimization: Formulation, Discussion and Generalization. Proceedings of the 5th International Conference On

Genetic Algorithms. San Mateo, California, pp. 416-423, University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[10] Srinivas, N. & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, vol. 2, pp. 221-248.

[11] Horn, J. & Nafpliotis. (1993) Multiobjective Optimization Using The Niched Pareto Genetic Algorithm. Tech. Rep. IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.

[12] Peace, G.S. (1992). Taguchi Methods. Addison-Wesley, Reading, MA.

[13] Webb, E. (1997). MINICAPS – a simplified version of CAPS for use as a research tool. Unclassified Report Bae-WOA-RP-GEN-11313, British Aerospace Plc.

[14] Breshears, C. (1995). A Beginner's Guide to PVM – Parallel Virtual Machine. Joint Institute For Computational Science, University of Tennessee, Knoxville, TN.