A Hybrid Multi-Objective Evolutionary Algorithm Using an Inverse Neural Network for Aircraft Control System Design

Salem F. Adra[1] S.adra@sheffield.ac.uk

Ahmed I. Hamody[1] Cop04aih@sheffield.ac.uk

Ian Griffin[1] I.griffin@sheffield.ac.uk

Peter J. Fleming[1] P.Fleming@sheffield.ac.uk

Abstract – This study introduces a hybrid multi-objective evolutionary algorithm (MOEA) for the optimization of aircraft control system design.
The strategy suggested here is composed mainly of two stages. The first stage consists of training an Artificial Neural Network (ANN) with objective values as inputs and decision variables as outputs to model an approximation of the inverse of the objective function used. The second stage consists of a local improvement phase in objective space preserving objectives relationships, and a mapping process to decision variables using the trained ANN. Both the hybrid MOEA and the original MOEA were applied to an aircraft control system design application for assessment.

1 INTRODUCTION

Evolutionary Algorithms are stochastic approximation techniques based on the concept of "survivals of the fittest". These approximation techniques are especially well tuned for solving multiobjective optimization problems due to their ability to explore vast solution spaces and search from a family of candidate solutions rather than from just a single point. However, these evolutionary techniques are less suited to fine-tuning structures that are already close to optimal solutions. As stated by Davis [5] and re-illustrated by Knowles [13], for improving optimization results achieved by genetic algorithms one should: "Hybridize where possible".

Local search processes conventionaly hybridized with evolutionary algorithms are most widely structured as follows:

1-Local perturbation of the decision variables of a certain individual in a local search range, moving it to another point in an adaptive or fixed size neighbourood in the decision variable space.

2-Evaluating the performance of the new individual by calculating the objective function.

3-Either replacing the old individual by the new one, or rejecting the new individual based on a certain acceptance criteria depending on the local search strategy (e.g. hill climbing, tabu search, simulated annealing).

In this work a new local search procedure is introduced which aims to improve the solutions achieved by a multi- objective genetic algorithm (MOGA) [8] directly in the objective space. A vaguely similar approach was introduced by [10] and applied to the ZDT bi-objective benchmark problems [16]. In [10] the main objective was to accelerate the search of MOEA by approximating the objective function using NN techniques. In Evolutionary Multiobjective Optimization (EMO) the bi-objective case is by far the most heavily studied. EMO applications, by contrast, are frequently more ambitious, with the number of treated objectives [4]. Study reveals [14 p147] that conclusions drawn from bi-objective analysis cannot be generalised to higher numbers of conflicting objectives, hence, the study presented in this work is dedicated to the investigation of "Many objective optimization" problem.

Real-world engineering design problems often involve the satisfaction of multiple performance measures, or objectives, which should be addressed simultaneously. Automotive and aerospace examples provide illustrations of some typical design challenges and demonstrate that these problems often involve a large number of objectives [7]. The local search suggested in this work is designed for addressing many objective (more than 2) optimization problems. The suggested method can be easily coupled with a progressive preference articulation strategy, in order to focus the search into specific regions of interests to the decision maker, by modifying the values of the objectives in an informed way in the right direction towards the goal values. The use of progressive preference articulation is a highly commendable strategy for enhancing the search and coping with many-objectives optimization. Improving solutions in objective space requires an inverse mapping of objective values into decision variable values, known as reverse engineering. This can be done by introducing the inverse of the objective function used in the MOGA. Unfortunately this is not feasible when dealing with complex objective functions, most simulation based objective functions and when the objective function is treated as a black box function. On the other hand, in ideal situations where the objective function is reasonably simple and available, inverting the objective function is a process that increases the computational cost of the algorithm, and therefore is not prefered.

The use of an inverse artificial neural network, trained with the objective values as inputs and the corresponding decision variables as outputs, consequently arose. In the proposed hybrid algorithm, a multi-layer perceptron (MLP) has been adopted for the ANN and trained with the exact objective and decision variable values calculated using the exact objective functions.

[1] Department of Automatic Control and Systems Engineering
The University of Sheffield, Sheffield, UK

2 NEURAL NETWORK

Neural Networks (NN) are sophisticated analytical techniques for modelling functions. It is a powerful approach to modelling stochastic and noisy patterns of data in order to produce predicted values of unknown systems. The NN needs to be trained to achieve desirable predictions and model complex functions as closely as possible. The process of teaching the NN consists of feeding it with samples of data and manipulating weighting variables by adjusting their values and minimizing prediction errors. When training a NN, it is vital to ensure well-spread, meaningful and problem defining data. Abundance of data is an essential point for achieving well-trained NN, although unfortunately data abundance is a major problem in several applications. Multilayer perceptrons (MLPs) are feedforward neural networks trained with the standard backpropagation algorithm [1][18]. They are supervised networks that require training with exact collected data. MLPs are widely used in the field of pattern classification and recognition. One of the drawbacks of ANN is the lack of standardization in choosing the number of hidden layers and hidden neurons per layer, which constitutes the architecture of a NN (see fig 1).

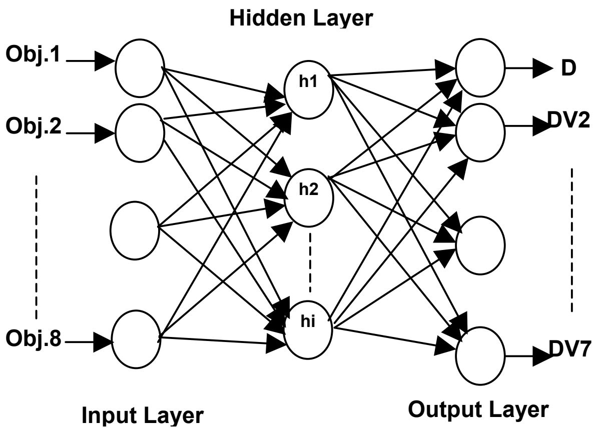

Figure 1: Multi Layer Perceptron

Hybridizing NN with the MOGA is very useful for approximating expensive objective functions. In the context of this work, NN is used in a supplementary local improvement step to map objective values back to decision variable space by approximating the inverse of the objective function. The ability to map objective vectors to decision variables will make it possible to search in objective space for desired combinations of objective values.

3 APPLICATION TO AIRCRAFT CONTROL SYSTEM DESIGN

The classical problem of optimizing an aircraft control system design was addressed to test the performance of the proposed optimization strategy. The importance, standardization, and abundant number of criteria to be optimized included in this classical problem were the main reasons for choosing this real world application as a benchmark for testing the proposed hybrid MOGA. In this section a simplified illustration of an aircraft dynamical model is shown and a common understanding of the multi- objective optimization problem is illustrated.

Figure 2 below illustrates the aircraft body in 3D Cartesian space. The longitudinal, or roll, axis is denoted X, the lateral, or pitch, axis is denoted Y and the vertical, or yaw, axis is denoted Z.

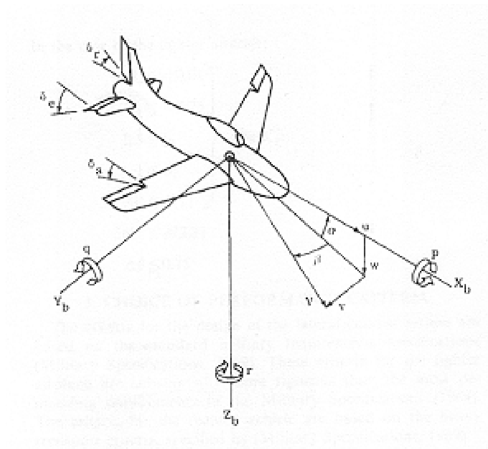

Figure 2: Three main axes of the body of an Aircraft

The motion of an aircraft in the air is described in terms of these 3 axes. During its motion, the aircraft makes a combination of changes in both angles and rates of angular velocities, therefore its dynamical model can be represented by an equation combining the main objectives involved in the motion of the aircraft. This equation is highly non-linear across the operating envelope of the aircraft, but it can be linearized for a small deviation around the equilibrium trajectory.

A simplified dynamical model of an aircraft motion can be represented by a fourth order linear equation [15]. The correpsonding state equation (1) is:

$$\overset{0}{x} = Ax + Bu \qquad \textbf{(1)}$$

where the state vector, x ,is:

$$x = \begin{bmatrix} \beta \\ r \\ p \\ \phi \end{bmatrix} \quad \begin{matrix} \textit{Sideslip} \\ \textit{yaw rate} \\ \textit{roll rate} \\ \textit{bank angle} \end{matrix}$$

Control u vector is:

$$u=\begin{bmatrix}\delta_a \\ \delta_r\end{bmatrix} \quad \begin{matrix}\textit{Aileron control motions} \\ \textit{Rudder control motions}\end{matrix}$$

The control vector, u, is represented by equation (2):

$$u=Cu_p+Kx \qquad \textbf{(2)}$$

where u_p is the pilot's control input vector
and C *and* K are the gain matrices of the form:

$$C=\begin{bmatrix}1 & 0 \\ K_5 & 1\end{bmatrix} \qquad K=\begin{bmatrix}k_6 & k_1 & k_2 & 0 \\ k_7 & k_3 & k_4 & 0\end{bmatrix}$$

By substituting equation (2) into (1) we get:

$$\dot{x} = (A + BK)x + BCu_p \qquad \textbf{(3)}$$

The eigenvalues of the matrix *(A+BK)* define the stability properties of the system modeled in equation (3). In the context of the current work, 8 essential objectives constituted the basis of the optimization process. The actual dynamical model represents additional characteristics, but a simplified model describing the major issues in controlling an aircraft stability is adopted in this work. The first of these 8 objectives to be optimized is the control effort (sum of squares of gain vector).

The other considered criteria consisted of the following:
-The characteristic roots of the matrix *(A+BK)*;

- the spiral root λ_8
- the damping in the roll root λ_R
- the damping ratio of the dutch-roll complex pair ζ_d
- the dutch-roll frequency ω_d.

-The required bank angle for the fighter aircraft according to the millitary specifications 1969 is :

- $\phi(1s) \geq 90^{\circ}$
- $\phi(2.8s) \geq 360^{\circ}$
- Minimise the sideslip angle (β) deviation .

For further details about the aircraft dynamic model and the variables mentioned refer to [2][6].

4 THE PROPOSED HYBRID ALGORITHM

The main goal of the proposed hybrid algorithm (fig 3) is to tackle the problem of optimizing many (more than 2) objectives whose various relationships demonstrate harmony, independency and conflict. The objective is to improve standard results achieved by the multiobjective genetic algorithm in particular, and any global evolutionary approach in general. This is achieved by introducing a local search process inside the evolutionary cycle. The incorporation of local search is designed to fine tune the population of solutions directly in the objective space rather than the decision variable space. When exploring solutions in the objective space, movements should be well structured and delicate to avoid leaping into unfeasible regions, thereby "fine-tuning" the solutions, leaving the process of stochastic exploration of space to the evolutionary global search of the MOGA. A feedforward artificial neural network was incorporated in the evolutionary process of the MOGA. The ANN was trained to represent the inverse relationship; i.e. the exact objective values, resulting from the objective function evalutions in the MOGA, are used as inputs to the ANN coupled with their corresponding decision variables as outputs.

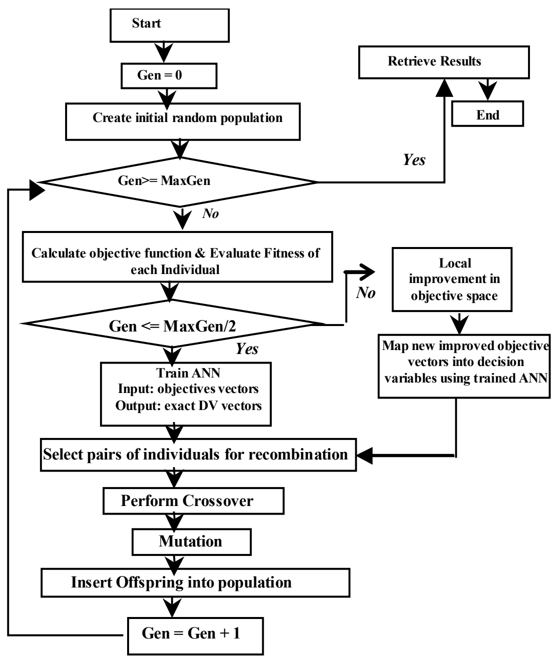

Figure 3: The Hybrid MOGA

The ANN was deployed later on to map new objective vectors highlighted by the local search into corresponding decision variables, with a minimum acceptable error of prediction.

4.1 Inverse ANN Training

Unless using meta-modeling techniques, generally, at every generation of a genetic algorithm, regardless of its sophistication and complexity, the objective function is calculated. This is done to assess the performance of the current population of solutions, and assign fitness scores to each individual based on the objective function results. Consequently the idea of training an ANN with exact data (objective vectors and decision variables) to model the inverse of the objective function has emerged, knowing that adding an ANN training process would not increase the complexity nor the computational costs of the algorithm, as the process would consist of straight forward, reasonably simple, mathematical calculations.

In this work, a multi layer perceptron (MLP) [1] with a single hidden unit, 30 hidden neurons, 8 input units denoting each of the 8 objectives tackled and 7 output units designated to the corresponding decision variables, was deployed (fig 4). The number of hidden neurons was set to 30, and was experientially demonstrated to be the most suitable and avoiding the problem of overfitting models due to high unsuitable ANN complexity for the problem considered in this work. The standard backpropagation learning algorithm was adopted in the learning process and weight adjustment. This training phase takes place at every generation of the MOGA until the process is half through the whole optimization process in terms of the predetermined number of generations. In order to ensure a diverse set of training data, the ANN was trained with the entire objective vectors as inputs and their corresponding decision variables vectors as outputs.

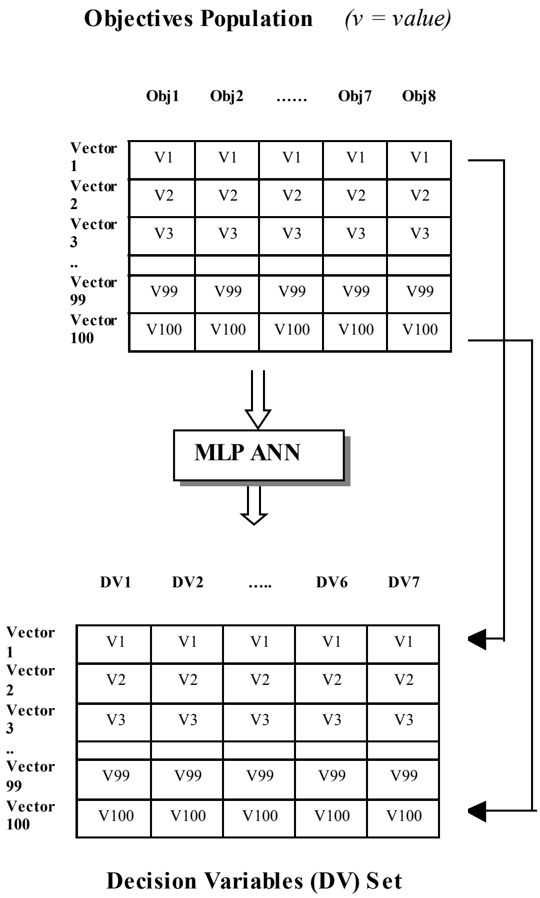

Figure 4: Training the ANN with exact data

It was noticed that training the ANN with 50 generations of data (objectives and DV), when running the MOGA for 100 generations, was enough to reach a reasonably small error in the prediction ability of the ANN. Due to the stochastic nature of the MOGA, we are more confident that the data fed to the ANN for training purposes is well spread and representative assuming no genetic drift.

4.2 Local Improvement

After having trained the ANN for a certain sufficient number of generations, the training process stops and the algorithm switches to the validation mode. At each of the remaining generations, a local improvement operator is applied to the current objective vectors. The introduced local improvement phase is commited to introducing overall local improvements in terms of all objectives as much as possible, i.e. without detriment to some objectives, especially when trying to improve one of two or more competing or harmonious objectives.

This is done by visually (using parallel coordinates [9][11] or scatterplot matrices [3]) or quantitively (e.g. Kendall sample correlation statistic [12]) determining *a priori* the relationships between the objectives and later on performing small logical and feasible improvements while preserving the relationship between the objectives (e.g. see fig.5). In other words, the pair-wise relationships existing between pairs of objectives were identified during the first half of the optimization process where the training process of the neural network takes place. Relationships such as harmony, independency and conflict between pairs of objectives are identified *a priori* the start of the local search process.

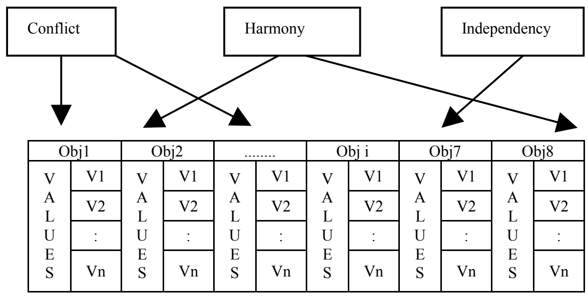

Figure 5: Example of Objectives relationship

Being able to easily integrate progressive preference articulation is a major benefit; the local search process might then be permitted to improve one of two competing objectives while deteriorating the performance of the competing one by an acceptable amount reflecting the decision maker's preferences.

The hybridization interface between the newly implemented local search technique and the MOGA is located prior to the selection for recombination and the recombination steps. This choice of location is designed to make any beneficial effects of the local search operator available to the genetic selection and recombination process. Consequently, the local search fine-tunes the parents of the genetic population instead of the offspring; this ideally should then produce fitter offspring. This concept has its analogy with the heredity mechanism in human biology; healthy parents are more probable to produce healthy offspring while parents suffering from a certain disease are more likely to transmit the disease or points of weakness by heredity to their descendants and thus produce weaker offspring.

The local improvement process is a well-defined and structured mechanism. The functionality of the local improvement process will be illustrated briefly using an 8 objectives optimization problem. The local search starts dealing with independent objectives that are not influenced by the improvement or the deterioration of other objectives. Each independent objective is treated separately, and the local search sorts the objective function vectors into descending order in terms of the designated objective function value, noting that a minimization problem is being considered (see fig 6).

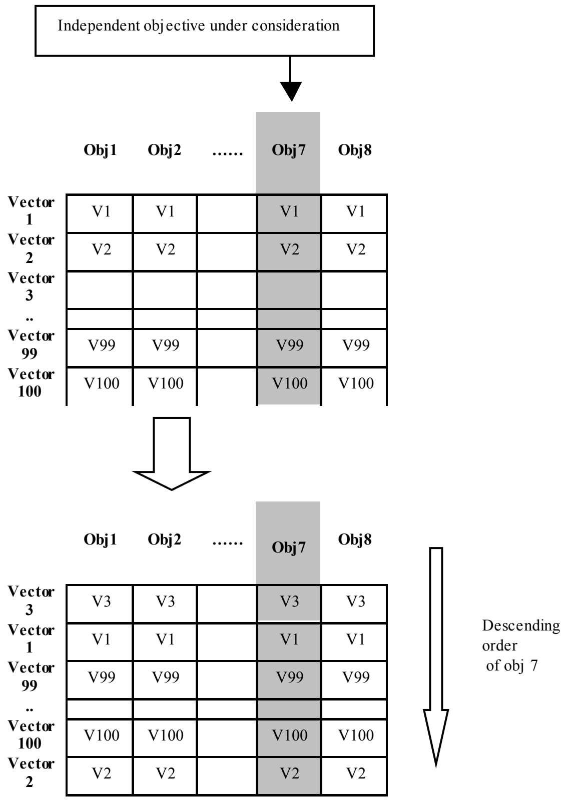

Figure 6: Sorting the objective vector in terms of obj7

After sorting the objective vector in terms of the independent obj 7, the first row in the matrix would consequently consist of the combination of objective values containing the worst (highest) value in terms of obj 7. The improvement process takes place in terms of that objective by shifting each value of obj 7 to the next best value, in other words, the worst value becomes the second worst, 4th worst → 5th worst and so on (for example from fig 6 (in the sorted matrix in term of obj7), v3 under obj 7 becomes v1, v1 → v99 ...etc). In this way, nind – 1 shifts takes place (nind = number of individuals), and end up only improving the best solution (from fig 6, v2 under obj7) by minimizing it furthermore by a small distance concluded from statistical information about the distribution and the clustering of solutions and the average, the minimum and maximum distance between neighbouring solutions in term of a specific objective. When shifting values, repetitive values are taken into consideration, and therefore all equal values are replaced by the next best value.

After having improved independent objectives, cooperating and competing objectives are then improved by applying similar "descending sort and improve" procedure while preserving relationships between the objectives, and taking into considerations the decision maker priorities.

4.3 Objective Vectors to Decision Variables mapping

After the local improvement in objective space takes place, we end up with a new objective vector locally improved compared to the original objective vector that resulted from the MOGA optimization process. The next step is to map the resulting objective vectors to corresponding decision variables vectors to be passed back to the global search process of the MOGA for recombination and mutation. This mapping process will be applied by feeding the new objective vector to the previously trained ANN, which in its turn will predict reasonably accurate corresponding decision variables (fig.7).

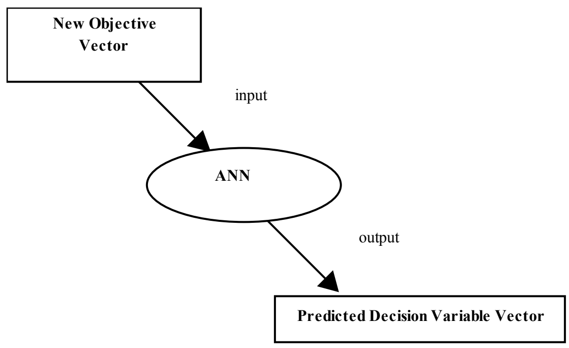

Figure 7: Objective vectors to Decision vectors Mapping

5 Results

The MOGA hybridized with the newly implemented local search technique and the inverse neural network training process was tested against the multiobjective optimization problem of aircraft control system design. The intention was to assess the performance of the hybrid algorithm, also called memetic algorithm, and compare its results with those produced by the MOGA. The MOGA and the hybrid genetic algorithm had the following basic common configuration: The genetic population size was set to 100 individual per generation. Concatenation of real number decision variables was the suitable choice to encode the problem under investigation. In addition a simple non-elitist strategy was used for the selection for survival process (i.e. no generational gap) and a Pareto based ranking along with stochastic universal sampling was used for the selection and recombination process. For the recombination process, a single-point two parents crossover with a probability of 0.8 alongside a Gaussian mutation was implemented. Due to the stochastic nature of the evolutionary strategies, a well-based judgment concerning the performance of a specific algorithm cannot be stated unless the whole optimization process is repeated a number of times. In the case of this work, each algorithm was subjected to 20 iterations, each running for 100 generations.

In the following, statistical results concerning the best non-dominated solutions achieved by the 2 optimization techniques are discussed. The binary ε-indicator [17] was deployed to assess the quality of the non-dominated solutions achieved by the memetic algorithm and contrast it with the quality of the best solutions achieved by the MOGA for each of the 20 runs executed. This performance metric computes the minimum epsilon value required for all solutions in an approximation set A in order to not to be worse than any solution in another approximation set B. Noting that the binary ε-indicator is an asymmetric operator, i.e. “EpsilonMetric (MemeticData, MogaData) ≠ EpsilonMetric (MogaData, MemeticData)“, it was constantly clear that the minimum ε value required for the non dominated solutions achieved by the memetic algorithm to dominate all the best solutions achieved by the MOGA was much more smaller than the minimum ε value required for the non dominated solutions achieved by the MOGA to dominate the memetic solutions.

The out-performance of the memetic algorithm was then articulated in terms of the resulting mean of the ε values achieved for the 20 non-dominated sets of solutions achieved at each run of the 2 algorithms. The significance of the experiential result is then assessed using randomization testing. This is a simple, yet effective, technique that does not rely on any assumptions concerning the attributes of the underlying processes. The central idea of the method is that, if the observed result has arisen by chance, then this value will not appear unusual in a distribution of results obtained through many random relabellings of the samples [14].

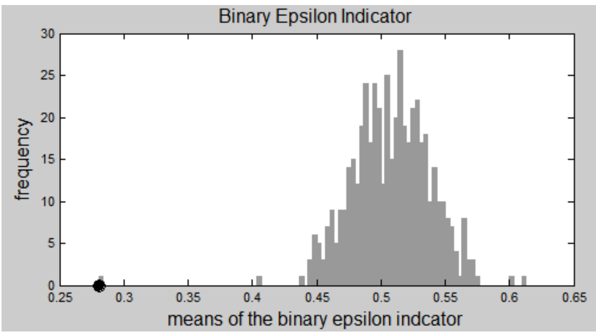

Figure 8: Statistical Significance of the outperformance of the memetic algorithm

In figure 8 the randomization testing of the mean of the binary ε-indicator results are illustrated to depict any statistical significance of the results. The grey histogram represents the randomization results whilst the real observed result is illustrated by the black circle. Generally it is visually remarkable if the achieved results are statistically significant or not, although sometimes closer analysis might be needed and minor improvements can be inferred. The mean (or alternatively median) values of the minimum ε values required for algorithm A to outperform algorithm B according to “ε-Metric(A,B)” (where A denotes the memetic algorithm, B denotes MOGA) was used as a the test statistic. For 1000 iterations, the non dominated solutions resulting from the MOGA and the memetic algorithm were combined, randomly relabeled and assigned to A and B in “ε-Metric(A,B)”. From figure 8, it was very clear that the observed remarks concerning the outperformance of the memetic algorithm over the MOGA were extremely significant. The real observed ε was depicted by the black circle in figure 8, and was obviously remarkable and distinguishable from the randomization results depicted by the grey histogram.

In the remaining part of this section, further results visualizations and testing are illustrated. In fig 9 the average values attained for each of the 8 objectives considered, at every run of the MOGA (dotted line) and the hybrid MOGA (stars line) are illustrated. The average values attained for each objective at the end of each 100 generations constituting a single run of an algorithm reflects the global performance of the underlying optimization technique. Noting that the problem under consideration is a minimization problem, it was very clear that the introduced hybrid algorithm outperformed the MOGA in terms of 7 out of 8 objectives.

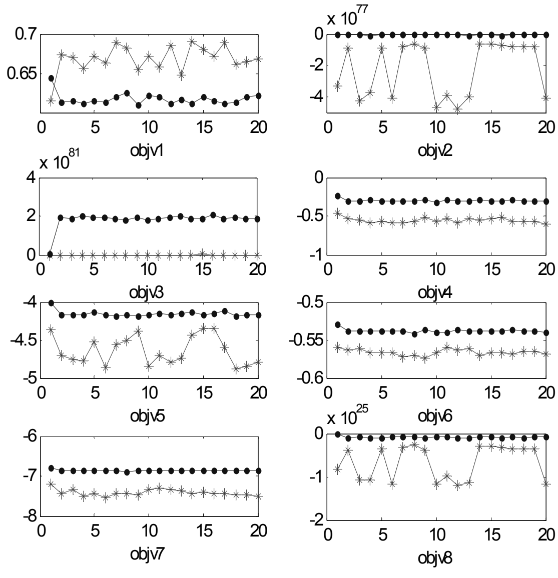

Figure 9: Average values of the objectives

On the other hand, assessing the minimum values achieved for the objectives at each of the 20 runs of the 2 algorithms, it is notable that the minimum values achieved for the 2 conforming objectives (2 and 8) has beaten the values acheived by the MOGA at every run (fig 10). Similar results are seen for objective 4. Results related to objective 7 were favouring the MOGA, while the remaining objectives were globally equivalent in term of minimum values reached in both algorithms.

Assessing the minimum and maximum values achieved for the objectives illustrates the search range of the algorithms and pinpoint the best globally non-dominated values achieved for these objectives .

In figure 11 the maximum values (i.e worst) of the non-dominated solutions attained by both algorithms are shown. It is remarkable that the hybrid MOGA has shiffted downwards the whole search range of objectives 1,2,3,4,5, and in other words has improved the search range by lowering its ceiling values. MOGA resulted lower maximum values in terms of objective 8. Intersecting performances were noted in terms of objective 7, while exactly the same maximum values were attained for objective 6 by both algorithms.

The performance of the introduced hybrid MOGA can be significantly deemed a success and an amelioration to the standard results achieved by the MOGA for the problem of 8 objective optimization of the design of an aircraft lateral control system.

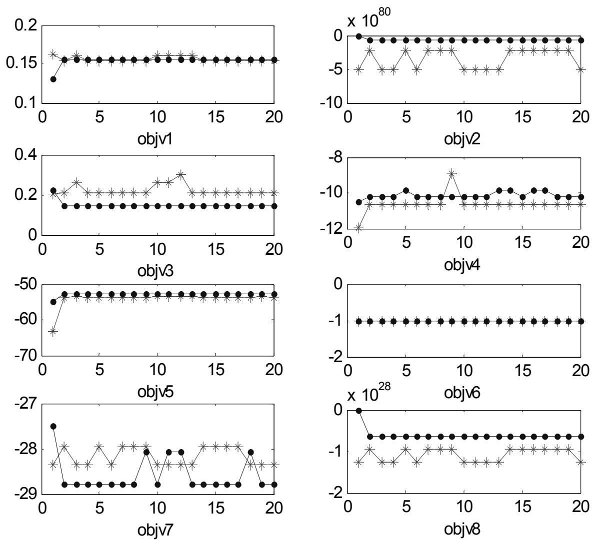

Figure 10: minimum values of the objectives

The search range of the majority of the objectives was improved, and the average values achieved for 7 out of 8 objectives were outperforming the results of the MOGA.

Furthermore, the non-dominated solutions were extracted from the set of non-dominated solutions achieved by the Hybrid MOGA and the MOGA. The statistical results illustrated in figure 12 show that at each run of the algorithms, at least 80% of the non-dominated solutions acheived by the hybrid MOGA were also non-dominated by the best results acheived by the MOGA. Furthermore, mostly less than 40% of the non-dominated solutions acheived by the MOGA were non-dominated compared to the values resulting from the hybrid approach.

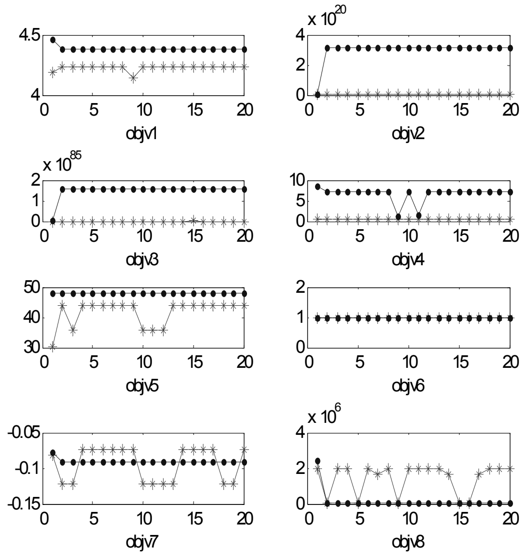

Figure 11: maximum values of the objectives

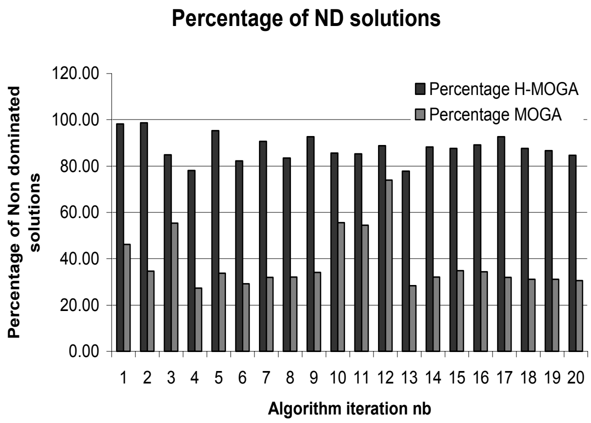

Figure 12: Percentage of nondominated solutions

6 Conclusions

A local improvement process in objective space was hybridized with the multiobjective genetic algorithm. An inverse artificial neural network was trained in the global search process of the MOGA, and was deployed to model the inverse objective function of an aircraft control system design problem to map objective values to decision variable space. The results achieved by the introduced hybrid MOGA are deemed a global improvement in terms of 7 out of 8 objectives, and constantly at least 80 % of the best solutions produced by the hybrid MOGA were non-dominated by the best solutions produced by the MOGA. Most of the research carried in the field of multiobjective optimization focuses on bi-objective problems. In real life applications, the number of objectives to optimize simultaneously can be much higher. Further research investigating the optimization of high numbers of objectives is surely needed and beneficial. On the other hand, artificial neural networks are highly earning interest and are widely used and deemed advantageous in several application domains. Further research concerning the improvement of ANN performances, investigating sophisticated learning algorithms and architectures is certainly profitable.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. Adra and Hamody also would like to thank Ahmad Nassef and Muhammad Al Karouri for their technical advice concerning ANN and Robin Purshouse for his appreciated advice and enlightening discussions.

Bibliography

1. Bishop C.: Neural Network for pattern recognition, Oxford University Press, 1997.
2. Blakelock J. H.: Automatic Control of Aircraft and Missile, Wiley, New York (1995)
3. Cleveland, W. S.: 1993, Visualizing Data, Hobart Press, Summit, New Jersey.
4. Coello, C. A. C., Veldhuizen, D. A. V. and Lamont, G. B.: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York 2002.
5. Davis, L.: Handbook of genetic algorithms, Van Nostrand Reinhold, 1991.
6. Etkin B.: Dynamics of Atmosheric Flight, Wiely, New York (1972).
7. Fleming P.J., Purshouse R.C., Lygoe R.J.: Many – Objectives Optimization: An Engineering Design Perspective, Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005, Guanajuato, Proceedings, Volume 3410 / 2005 ISBN: 3-540-24983-4, Mexico 2005, pp. 14-32.
8. Fonseca, C. M., Fleming, P. J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in S. Forrest (ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kauffman Publishers, San Mateo, California, pp. 416–423 (1993).
9. Fonseca C.M., Fleming P.J.: Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms- part I: unified formulation, IEEE Transactions on systems, Man, and Cybernetics, Part A: system and human 28 (1), 38 – 47 (1998)
10. Gaspar-Cunha A., Vieira A.: A hybrid multi-objective evolutionary algorithm using an inverse Neural Network, pages 25-30, ISBN 3-00-015331-4, (2004)
11. Inselberg, A.: The plane with parallel coordinates, The Visual Computer 1, 69–91 (1985)
12. Kendall, M. G.: 1938, A new measure of rank correlation, Biometrika 30, 81–93.
13. Knowles, J.: Local-search and hybrid evolutionary algorithms for Pareto optimization, PhD thesis, Department of computer science, The University of Reading, UK (2002)
14. Purshouse, R. C.: On the Evolutionary Optimisation of Many Objectives, PhD thesis, Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK 2003
15. Tabak D., Schy A. A., Giesy D.P., Johnson K.G.: Application Of Multiobjective Optimization in Aircraft Control Systems Design, Pergamon Press Ltd. 1979, Automatica Vol. 15, pp. 595~600.
16. Zitzler, E., Deb, K., Thiele, L.: 2000, Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation 8(2), 173–195
17. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M. and Grunert da Fonseca, V.: 2003, Performance assessment of multiobjective optimizers: An analysis and review, IEEE Transactions on Evolutionary Computation 7(2), 117–132.
18. www.ncrg.aston.ac.uk/netlab/