An Informed Convergence Accelerator for Evolutionary Multiobjective Optimiser

Salem F. Adra[1]
+44 (0) 1142225236
The University of Sheffield, UK
s.adra@sheffield.ac.uk

Ian Griffin[1]
+44 (0) 1142225682
The University of Sheffield, UK
i.griffin@sheffield.ac.uk

Peter J. Fleming[1]
+44 (0) 1142225663
The University of Sheffield, UK
p.fleming@sheffield.ac.uk

ABSTRACT
A novel optimisation accelerator deploying neural network predictions and objective space direct manipulation strategies is presented. The concept of directing the search through the use of 'mirage' solutions is introduced and investigated. The accelerator is meant to be a portable component that can be plugged into any stochastic optimisation algorithm, such as genetic algorithms. The purpose of the new component termed as the Informed Convergence Accelerator (ICA) is to enhance the search capability, convergence extent and most especially the speed of convergence of the hosting stochastic global optimisation technique. ICA was hybridized with the Non-Dominated Sorting Genetic Algorithm (NSGA-II). Enhanced results were achieved demonstrating the utility of the introduced component.

Categories and Subject Descriptors
I.2.8 **Computing Methodologies**: Artificial Intelligence -*Problem Solving, Control Methods and Search.*

General Terms
Algorithms, Performance, Experimentation,

Keywords
Evolutionary Multiobjective Optimisation, Convergence Acceleration

1. INTRODUCTION
Differently from single objective optimisation which aims to maximize, minimize or achieve a certain goal value for a single objective, multiobjective optimisation consists of multiple criteria, more often competing, that need to be optimised simultaneously. Automotive and aerospace applications provide illustrations of some typical design challenges and demonstrate that these problems often involve a large number of objectives. Solving a Multiobjective Optimisation Problem (MOP) consist of finding a well distributed set of optimal solutions or tradeoffs which cannot be improved furthermore in terms of any single objective without introducing a consequential deterioration in terms of one or more other competing objectives. This set of optimal solutions is called the Pareto optimal set. Without any loss of generality, an optimisation problem can be formulated as a minimization of a certain function **Z(X),** where **Z(X) = {Z1(X)...Zn(X)}** is a vector of objective functions, **n** is the number of objectives to be optimised and **X** is a vector of decision variables. In Figure 1 an optimisation problem where 3 decision variables are optimised with respect to two competing objectives is illustrated. A multiobjective optimiser tackling a MOP should ideally provide the decision maker (DM) with a diverse set of tradeoff solutions close to or preferably lying on the true Pareto front. In optimal scenarios, the set of solutions for a MOP, also called *approximation set* [1], is achieved within an acceptable amount of time and an affordable budget of computational effort.

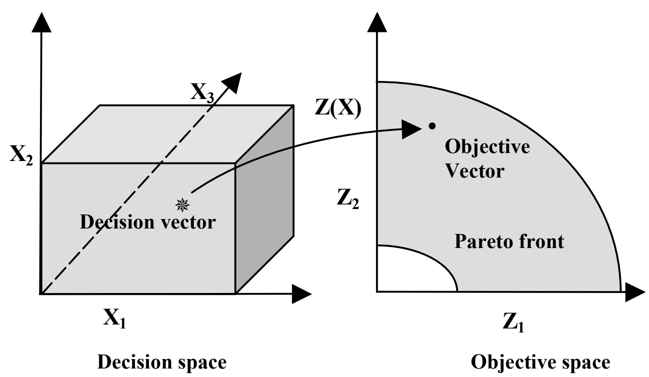


Figure 1. The Multiobjective Problem Domain.

Evolutionary Algorithms (EAs) are well tuned for solving MOPs due to their ability to explore vast solution spaces and search from a family of candidate solutions. Traditional evolutionary computation (EC) techniques usually consist of an explorative set of procedures in the decision variable space represented by recombination and mutation. Despite their utility for solving MOPs, EAs repeated search in the decision variable space imposes an extensive number of objective function calculations and therefore makes these approximation techniques computationally expensive especially when dealing with objective functions which are expensive to evaluate in themselves.

The use of surrogate models using Neural Networks (NN), or other metamodelling techniques such as Kriging-based

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

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

approximations or response surface models [2], [3] is a well established strategy that replaces the computation of actual objective functions and reduces the computational burdens of EAs caused by the calculation of expensive objective functions.

In this work a new add-on operator that accelerates the cycle of multiobjective optimisers is introduced. The operator termed as the *Informed Convergence Accelerator (ICA)* is based on an underlying machine learning strategy that deploys neural networks to learn and capture the function that maps from the objective space to the decision variable space of a MOP. Although it makes use of metamodelling techniques in an unconventional way, ICA is not designed as a metamodel which substitutes the use of the real objective functions. The ICA is a portable component that can be plugged into any stochastic optimisation algorithm, such as EAs. Its main purpose is to enhance the search capability of the hosting EA by increasing the speed of convergence of the handled solutions towards the Pareto front. The convergence acceleration resulting from the ICA does not interfere with the active diversification mechanisms of the hosting search strategies and requires reduced budgets of objective function evaluations.

EAs have traditionally emphasized decision space exploitation and decision space to objective space mapping and have failed to make direct exploitation of the objective space. Differently from traditional EAs, ICA makes use of the predictive capabilities of neural networks and an innovative direct search in the objective space. Performing local search in the objective space was previously introduced in [4], and briefly suggested in [5] and [6] and was demonstrated to be beneficial. The ICA suggests deterministically improved solutions in the objective space together with their decision variables which are predicted by the trained NN. This composes the machine learning component of ICA. The suggested solutions expose enhanced regions in the objective and the decision space and guide the optimiser's search towards good regions of solutions at a much faster rate. These introduced solutions might be either unfeasible or lacking accuracy in the decision variable space, but uncover very promising and desirable regions in the objective and the decision space. Indeed, studies [7] [8] showed that keeping dominated and infeasible solutions in the population helps enhancing the explorative capacities of the optimiser.

In the context of this work ICA is hybridized with the Non-Dominated Sorting Genetic Algorithm (NSGA-II) [9] (Figure 5), a well-established multiobjective optimiser in the Evolutionary Multiobjective Optimisation (EMO) community.

In the following sections of the paper, a brief introduction of neural networks and the needed concepts for this paper will be presented. The suggested ICA will then be introduced and described in details. Last but not least, the experimental results will be shown and future work directions will be stated.

2. NEURAL NETWORK

Neural Networks are a powerful approach for modeling stochastic and noisy patterns of data in order to produce predicted values for unknown systems. The NN needs to be trained in order to achieve desirable predictions and model complex functions accurately. The process of training a 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 and problem defining data. Multilayer perceptrons (MLPs) [10] (Figure 2) are feedforward neural networks usually trained with the standard backpropagation algorithm [10, 11]. They are supervised networks that require training with exact collected data. MLPs are widely used in the field of pattern classification and recognition. Hybridizing NN with an EA is very useful for approximating expensive objective functions.

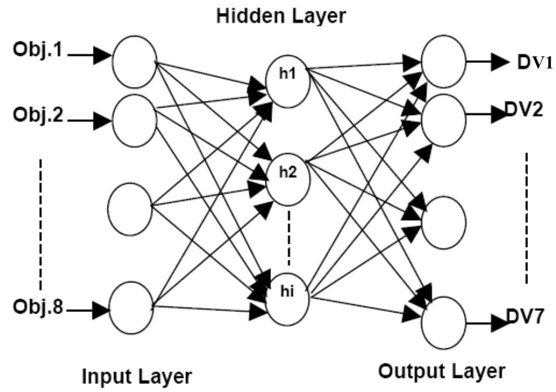


Figure 2. Multi Layer Perceptron

3. THE INFORMED CONVERGENCE ACCELERATOR

3.1 Motivation

In many application domains, it might be required to rerun a MOEA, as such stochastic optimisation techniques has no guarantee for finding optimal solutions within a single run. The need for re-executing a MOEA is especially required when facing a multiobjective optimisation scenario, where decision maker's preferences might progressively change over time. Initializing a MOEA with a population of previously stored good solutions does not necessarily satisfy the DM's interests and meet the desired scenarios. Indeed, it might be required to converge to a new region of interest (ROI) ignoring or constraining certain objectives or dimensions. Unless certain progressive preference articulation techniques are incorporated in the search process and are manipulating the fitness assignment procedure, restarting a new run of a MOEA from a certain front in the objective space might not converge to the new articulated ROI. This scenario especially occurs if the new preference articulations aim at stressing regions of the space previously considered as sub optimal during the previous runs.

3.2 Training The Neural Network

In this work, a NN was deployed to capture the function which maps a certain objective vector back to its corresponding vector of decision variables. This is achieved by training a NN with the objective vectors as inputs and their corresponding decision variables as outputs. The training data is the exact data resulting from the entitled executions of the objective function within the cycle of a single run of a MOEA. Moreover, the training of the neural network takes place offline during an entire previous execution of a certain MOEA. The offline training of the NN allows it to establish a 'good knowledge' about the global landscape of the objective and decision space of a certain optimisation problem. The NN is then used -when and if required-at a following execution of the same MOEA or an alternative MOEA attempting to solve the same optimisation problem. More specifically, a multilayer perceptron was deployed for training a NN with the exact data resulting from 20000 executions of the objective functions used in this work within 200 generations of NSGA-II. The architecture of the NN was designed based on a trial-and-error set of experimentations. The standard backpropagation algorithm was used for training the NN.

3.3 Mining the Objective Space

The ICA consists of a local improvement procedure in the objective space followed by a mapping process to the decision space. The local improvement will be responsible for enhancing the quality of the current front handled by a MOEA, especially when the search gets stagnated at certain areas of the space after certain number of generations.

Because of the stochastic nature of MOEAs' search, it is noteworthy that in many scenarios the manipulated solutions only become interesting after a considerable number of generations, especially when the MOEA is initialised with a random population of solutions. The introduced ICA speeds up the optimiser and guides the search towards ROIs in the objective space upon the DM/Operator's command. Executing ICA early in the optimisation process is particularly efficient as this accelerates the process of exploring and reaching superior areas of the hyperspace which otherwise would be reached at much later phases of the process. In this way the optimiser efficiently spends the significant amount of generations exploiting ROIs and near optimal regions of the hyperspace. Figure 3 illustrates the actions of the hybridised MOEA which includes the ICA. Trajectories 2 and 3 describe the specific actions of the ICA.

Trajectory 1: the mapping between a decision variables vector realised by a MOEA and its corresponding computed objective values vector.

Trajectory 2: the resulting objective vector – a member of the approximation set at generation 'n' - is improved in the objective space.

Trajectory 3: a prediction of the decision variables vector corresponding to the improved objective vector is made using the neural network previously trained with the exact data resulting from an earlier execution of an MOEA.

Once executed, the ICA starts by creating a 'mirage' new set of solutions in the objective space. The new solutions are a

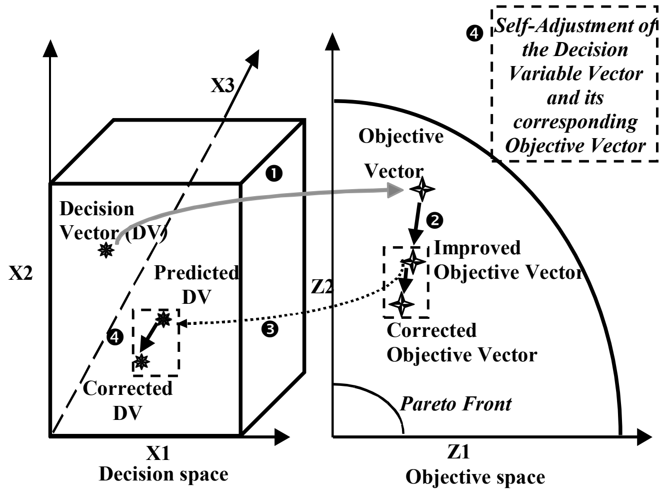


Figure 3. Illustration of the ICA on a single solution

deterministically improved version –in terms of Pareto optimality or DM preferences- of the current front of solutions (in the objective space) handled by the optimiser. The introduced solutions -objective vectors and their estimated decision vectors- predicted by the previously trained NN, can be thought of as baits thrown at desired and specific locations in the objective space. These solutions are termed as 'mirage' because they can map to inaccurate decision variables in the vicinities of the real decision variables. Nevertheless, these solutions attract the MOEA and force it to reconsider the content of the active archive. Basically, the devised solutions signal to the optimiser that a more attractive area of the space is exposed. The optimiser abandons the sub optimal solutions it is dealing with, filters out the inferior points, and converges towards the highlighted regions of the objective and the decision variable space.

When close enough to the Pareto front, careful and delicate step sizes should be taken when introducing the new solutions in the objective space to avoid leaping into the infeasible regions of the objective and/or the decision space. In the context of this work, the objective space improvement process was based on simple transitions and linear interpolation similar to the local search applied in [4] and [6] (Figure 4).

The step size of the objective space improvement is an application dependent parameter, and should be influenced by the landscape of the objective space, which is proper to a certain problem. In the context of this set of experimentations, and after investigating different step sizes (in the range 1% to 20%), the step size (***X*** in

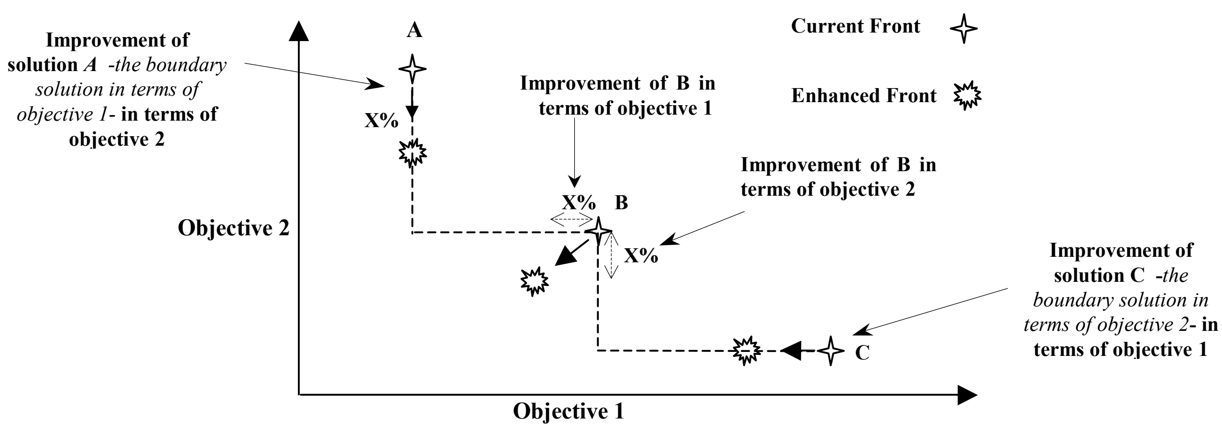


Figure 4. Deterministic improvement of the tradeoff surface in objective space

Figure 4) of the objective space improvement was set to an average of 10% progression of the objective values towards the next best values achieved at the time of the local improvement. Future work will be dedicated for experimenting the use of progressively articulated objective space improvement step sizes towards goal values and ROI.

```
-Initialize Population
-Generate random population P0– size Nind
-Evaluate objective values
For i=1 to Gen
    -Assign rank to Pi-1
    -Calculate Crowding measure of solutions in Pi-1
    -Generate offspring population Q – size Nind
      -Binary tournament selection
      -Recombination and Mutation
      -Evaluate objective values for the offspring population Q
     -Combine parent population Pi-1 and offspring Population
      Q – size: 2*Nind
    -Assign rank to the combined Population
    -Determine crowding distance for the combined Population
    -Select Nind solutions to form Pi and propagate to the next
    generation
      If ICA executed by the DM
          Apply ICA on Pi
        (New Objective vectors are mapped to their decision
              vectors using a previously trained NN)
      End
End loop
```

Figure 5. NSGA-II Hybridized with ICA

After improving the objective vector in objective space, a more or less accurate prediction of the corresponding decision variable vectors are predicted using the previously trained NN. The ICA does not look for adjusting any decision variables inaccuracy, and therefore does not require any additional objective function evaluations. However, this is achieved on the expense of a constrained usage mode. Using the ICA, any accompanied inaccuracy introduced by the NN will be observed for a certain number of generations. It is therefore suggested that the ICA should be executed in an interruptible way and upon the DM request. The MOEA should also be executed on its own for a few generations, after stopping the ICA process, in order to allow for self-readjustment, or 'cooling down', of any decision variables' imprecision introduced by the NN predictions (step 4 in Figure 3). The self-readjustment of any results' imprecision will be achieved by the exploitation/exploration processes (which include the evaluation of the exact objective function) of the MOEA in the highlighted areas of the decision variable space.

4. TEST FUNCTIONS

In this work the test functions used to test the effect of the introduced ICA are the convex bi-objective test function ZDT1 and the discontinuous test function ZDT3. These test functions belongs to a set of test functions introduced in [12] and are widely used in the EMO community to test multiobjective optimisers due to their well defined true Pareto fronts and the different challenges they introduce. NSGA-II together with its ICA-hybridized version were attempting to solve (minimize) the above mentioned test functions and had the following configuration:

Table 1. Optimisers Configuration

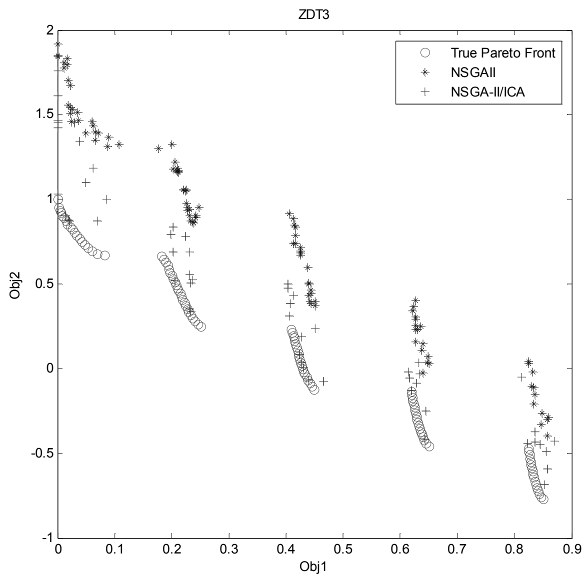
Size of Population	100
Crossover operator	Simulated Binary Crossover (SBX) [13] Probability: 0.8
Mutation Operator	Naïve Gaussian Mutation with Probability: 1/(number of Decision Variables)
Number of generations	50
Number of Runs	10
Starting Population	Same Random Population (different at each run)

In order to analyze the performance of the multiobjective optimiser and its ICA-hybridized version, visualizing the Pareto fronts achieved for the bi-objective test functions was noted sufficient to judge the efficiency of the ICA, especially because the ZDT test functions possessed well defined true Pareto fronts.

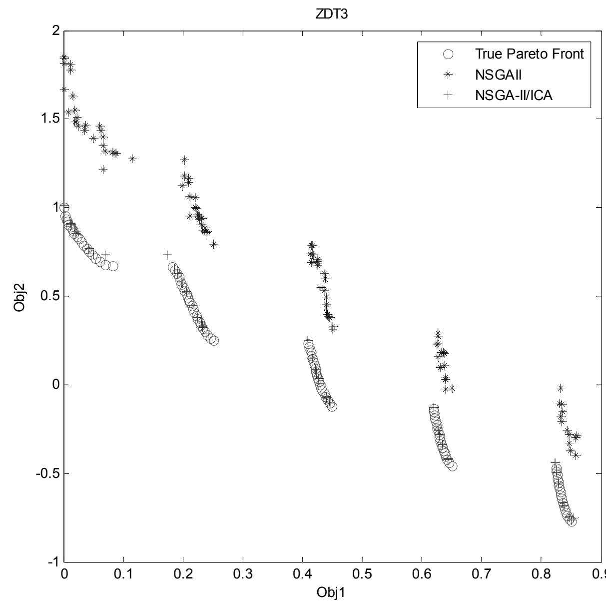
5. RESULTS

At 3 different stages of the optimisation process, 3 different snapshots illustrating the fronts achieved in the objective space by NSGA-II and its ICA hybridized version -NSGA-II/ICA- are presented in Figure 6 (a, b, c). The fronts handled by NSGA-II and NSGA-II/ICA are represented by the signs: (*) and (+) respectively. In Figure 6 (a, b, c), NSGA-II was optimising the discontinuous test function, ZDT3. NSGA-II is a well-established optimiser which usually gets to the true Pareto front of that test function within an average of 200 generations. In the carried experimentations NSGA-II was only executed for 50 generations. The ICA with its previously trained NN was plugged into NSGA-II and launched at the 30th generation for 6 successive generations. The '+' in Figure 6 (a, b, c) present the new solutions introduced by the ICA in the objective space. The true Pareto front of the ZDT3 test function is illustrated by the small circles in Figure 6 (a, b, c). Figure 6 (d) shows the accuracy (decision variables) of the results achieved by NSGA-II/ICA at the end of the optimisation process. The dots **'.'** in Figure 6 (d) correspond to the real values of the 2 objectives (*column 1→objective 1, column 2→objective 2*), while the stars **'*'** correspond to the values of the 2 objectives achieved by the NSGA-II/ICA. The real values ('.') are the objectives values calculated using the exact objective function for the decision variables produced at the end of NSGA-II/ICA process. The magnitude of the bars joining '.' and '*' in the NN accuracy plots illustrate the results accuracy and the difference between the expected objective vector and the actual results. Executing the ICA at the 30th generation of NSGA-II has introduced a new front (+) in the objective space.

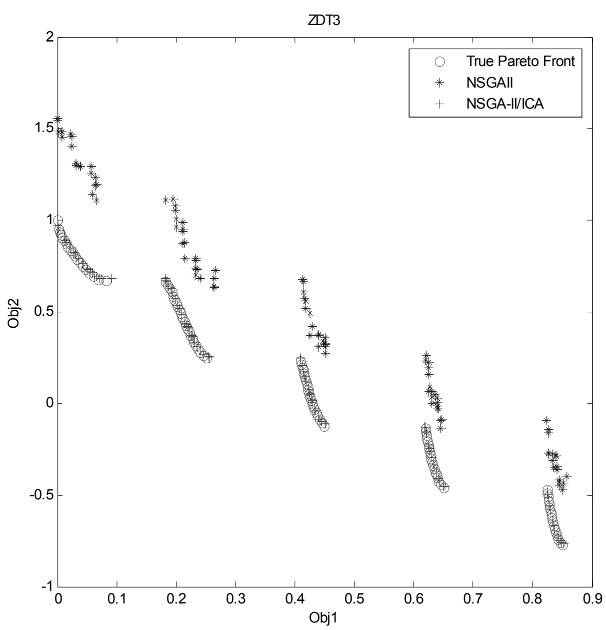
The corresponding decision variables for the introduced front were predicted by the previously trained NN and lacked accuracy in the decision space.



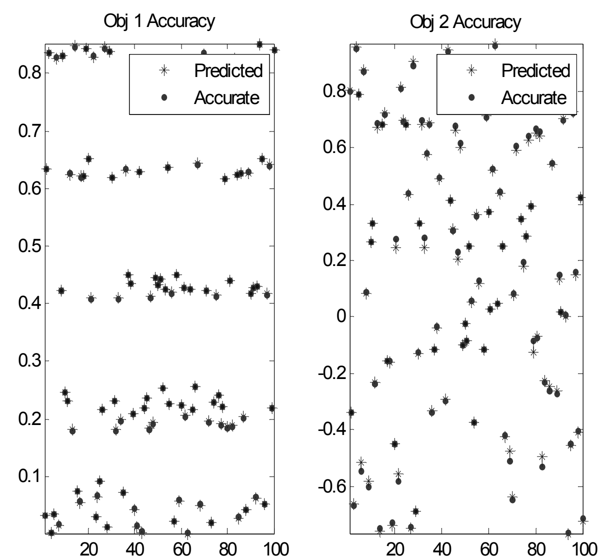
(a) ZDT3: 30th generation



(b) ZDT3: 35th generation



(c) ZDT3: last generation (50th)

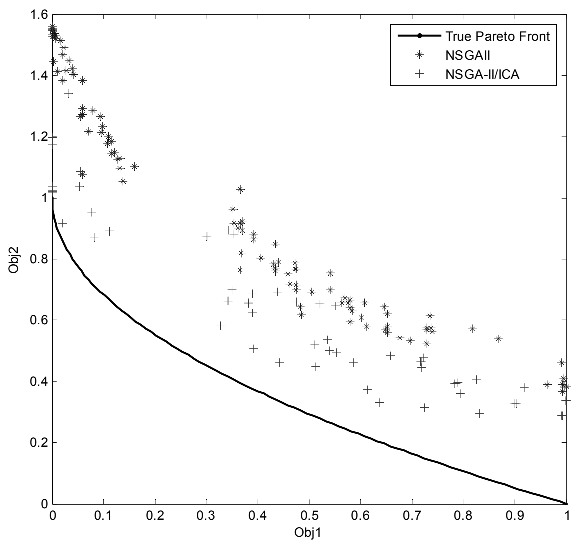


(d) Results Accuracy at the 50th generation

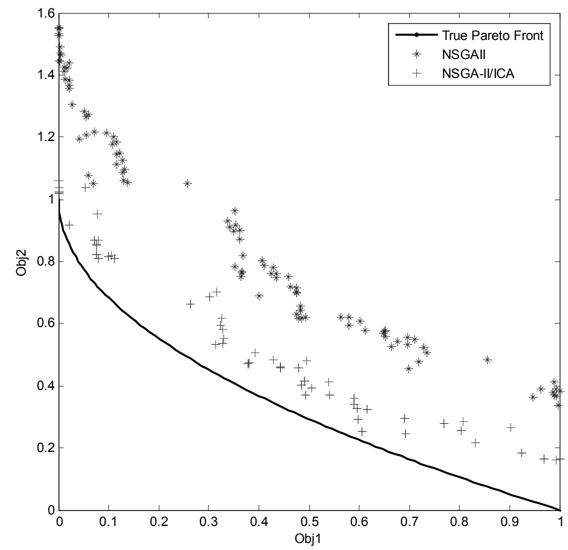
Figure 6. NSGA-II and NSGA-II/ICA Optimising ZDT3

In order to investigate the impact of the ICA on the optimisation process, executing the ICA at the 30th generation has resulted the introduction of the NSGA-II/ICA optimiser which will control and manipulate independently the new set of solutions (+) until the end of the optimisation process. On the other hand, NSGA-II will carry on its operations on the original front of solutions (*). Splitting the optimisation process at the 30th generation into 2 optimisers running concurrently and operating on two separate populations of solutions was aimed at contrasting the final results that would be achieved by NSGA-II boosted by the ICA and the results achieved by the standalone NSGA-II within 50 generations.

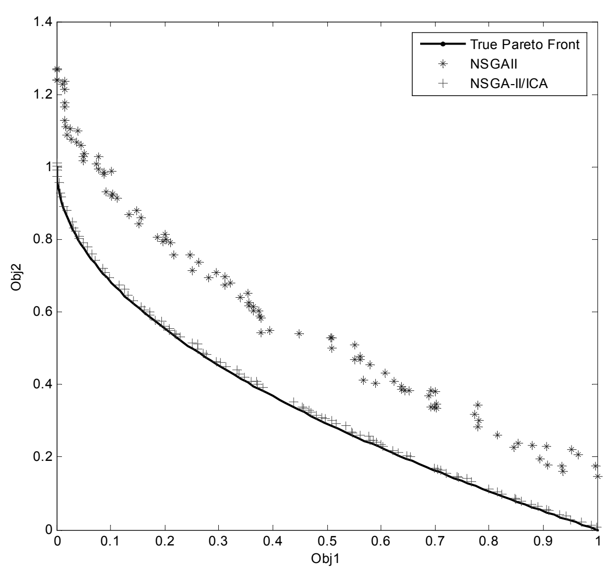
Despite introducing some inaccuracy in the decision variable space, the ICA made the optimiser leap into more interesting areas of the objective space and much promising regions in the decision variable space. Reducing the transitory inaccuracy in the decision variable space can be dealt with by ameliorating the predictive capacities of the NN and deploying sophisticated training algorithms and will constitute a future work. Having most of the handled solutions already very close to the true Pareto front ((+) in Figure 6b), the ICA was halted at the 35th generation in the NSGA-II/ICA framework. This is done in order to allow the adjustment of any decision variable inaccuracy (introduced by the ICA). Within few more generations it was observed that the optimiser starts exploiting the highlighted and promising decision variable regions which are supposed to give the enhanced objectives values '+' in Figure 6. At the same generations (30th → Figure 6a and 35th → Figure 6b), the standalone NSGA-II optimiser was running and exploring much further regions in the decision variable and objective space ('*' in Figures 6 (a, b, c)). The results in Figure 6c illustrate the final fronts achieved at the 50th and last generation of the optimisation by NSGA-II ('*') and NSGA-II/ICA ('+') with the ICA operating from the 30th until the 35th generation. The NSGA-II/ICA with the ICA active for 6



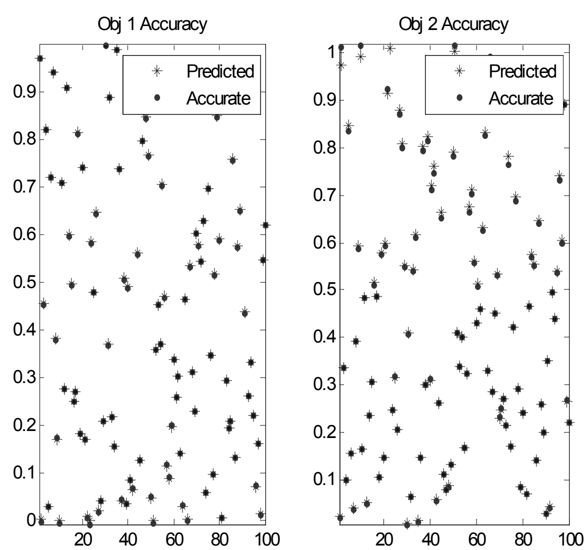
(a) ZDT1: 30th generation



(b) ZDT1: 35th generation



(c) ZDT1: 50th generation



(d) Results Accuracy at the 50th generation

Figure 7. NSGA-II and NSGA-II/ICA Optimising ZDT1

generations has achieved the true Pareto front for the ZDT3 discontinuous test function within 50 generations. The exactitude of the resulting results was precise and accurate (just some minor bars in Figure 6d) because of the correction period after turning off the ICA which has allowed the optimiser to readjust the mapping accuracy between the decision variables and their corresponding objective values. On the other hand, it was clear that the standalone NSGA-II has achieved a much further and sub optimal front, which is a standard result usually achieved within 50 generations. Similar to the experimental scenario presented in Figure 6, Figure 7 (a, b, c) shows the fronts achieved by NSGA-II and NSGA-II/ICA for the convex test function ZDT1. The NSGA-II/ICA was consistently, over 10 runs of the optimisers, achieving the true Pareto fronts of the two test functions deployed in this study within 50 generation. The standard NSGA-II on the other hand was achieving much lower quality fronts within the same amount of time. The final prediction error in the ZDT1 optimisation framework is shown in Figure 7 (d) and illustrates a very good level of accuracy.

Additional experiments were carried to investigate the advantages of the ICA when hybridized with a different MOEA solving the same multiobjective optimisation problems. The ICA operator was hybridized with the Strength Pareto Evolutionary Algorithm (SPEA2) [14], and the scenarios presented in Figure 6 and Figure 7 were executed contrasting the performance of the standalone SPEA2 and the SPEA2/ICA. Compliant results illustrating the enhanced performance of the SPEA2 optimiser hybridized with an active ICA (SPEA2/ICA) running for 6 generations and optimising the same test functions were observed.

In the previously described scenarios, the utility of the *informed convergence accelerator* was demonstrated. It did not make any significant difference whether the ICA was plugged and executed within the cycle of SPEA2 or NSGA-II. The only difference that could be inferred from the underlying MOEA is the extent and utility of the diversification mechanisms, which are proper to a certain MOEA.

6. CONCLUSIONS

Plugging the ICA into a state of the art MOEA such as NSGA-II or SPEA2 has presented enhanced results outperforming the results achieved by these standalone optimisers for a set of test functions deployed in this work. The concept of introducing ‘mirage’ solutions to guide the search towards promising areas of the space of interest (most commonly the objective space) was introduced and shown beneficial. The use of NN as a trained operator storing useful information between runs of a stochastic optimisation algorithm, such as evolutionary algorithms, is beneficial and has showed great use for enhancing the convergence speed of MOEAs. The ICA can be particularly convenient for optimisation problems under uncertainty, where noise and uncertainty are tolerable within certain relaxation intervals. Future work will focus on ameliorating the predictive capabilities of the deployed NN, and devising learning strategies for modeling multimodal objective functions. Future work will also include validating the advantageous effect of the ICA on real world applications and problems dealing with higher number of objectives.

7. REFERENCES

[1] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Fonseca, V. G. d. "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Trans. on Evolutionary Computation*, vol. 7, pp. 117-132, 2003.

[2] Farina, M. "A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm," Proc. *Congress on Evolutionary Computation (CEC'2002)*, Piscataway, New Jersey, 2002.

[3] El-Beltagy, M. A., Nair, P. B., and Keane, A. J. "Metamodeling Ttechniques for Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations," in the Proc. *of the Genetic and Evolutionary Computation Conf. GECCO-99*, San Francisco, CA, 1999.

[4] Adra, S. F., Hamody, A. I., Griffin, I., and Fleming, P. J. "A Hybrid Multi-Objective Evolutionary Algorithm Using an Inverse Neural Network for Aircraft Control System Design," Proc. *IEEE Congress on Evolutionary Computation (CEC'2005)*, Edinburgh, Scotland, 2005.

[5] Gaspar-Cunha, A., and Vieira, A. S. "A Hybrid Multi-Objective Evolutionary Algorithm Using an Inverse Neural Network," Proc. *Hybrid Metaheuristics, First International Workshop, HM 2004*, Valencia, Spain, 2004.

[6] Gaspar-Cunha, A., Vieira, A. S., and Fonseca, C. M. "Multi-objective optimization: Hybridization of an evolutionary algorithm with artificial neural networks for fast convergence," in *Fourth EU/ME Workshop on Design and Evaluation of Advanced Hybrid Meta-Heuristics*, Nottingham, U.K, 2004.

[7] Deb, K., and Goel, T. "Controlled Elitist Non-Dominated Sorting Genetic Algorithms for Better Convergence," Proc. *the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, 2001.

[8] Mezura-Montes, E. E., Coello, C. A. C., and Vela'zquez-Reyes, J. "Increasing Successful Offspring and Diversity in Differential Evolution for Engineering Design," Proc. *Adaptive Computing in Design and Manufcature (ACDM 2006)*, Bristol, UK, 2006.

[9] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," in *Parallel Problem Solving from Nature - PPSN VI*, Berlin, 2000.

[10] Bishop, C. M. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.

[11] Nabney, I. T. *Netlab: Algorithms for Pattern Recognition*. London: Springer Verlag, 2002.

[12] Zitzler, E., Deb, K., and Thiele, L. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, pp. 173-195, 2000.

[13] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems*, vol. 9, pp. 115-148, 1995.

[14] Zitzler, E., Laumanns, M., and Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Evolutionary Methods for Design, Optimisation and Control*, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: CIMNE, 2002, pp. 95 - 100.