Bi-Objective Optimisation of Expensive Functions

ME527: Introduction to Engineering Optimisation

Week 11 Coursework 2024

Josef Manlutac

201922513

Date: 28/03/2024

1 Non-surrogate Based Global Search Strategy

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) was chosen as the non-surrogate based global search strategy as it is a sophisticated evolutionary algorithm known to be applied to multi-objective global optimisation problems. NSGA-II is an improvement on NSGA-I as it extends the original implementation by providing a form of elitism and calculates a crowding distance within the selection process. Elitism allows the progression of optimal solutions to be carried to following generations and thus preserving quality. Meanwhile, the crowding distance (niching) mechanism enables sufficient exploration within the search space by preferring individuals that are more isolated from other points within the solution space. These selector operators encourage diversity while keeping strong candidates to create new generations with.

ALGORITHM: NON-DOMINATED SORTING GENETIC ALGORITHM II	
1	***Initialise*** *parent population* P_0 *with* N *samples.*
2	*Evaluate the fitness of each individual in* P_0
3	*Sort* P_0 *based on non-domination to identify different fronts*
4	***While*** *generation count < maximum generations*
5	*Select parents from current population* P_i *based on rank and crowding distance*
6	*Applying recombination and mutation on some parents to create offspring* Q_i *population of size* N
7	*Combine population* P_i *and* Q_i *to form* R_t
8	*Rank new population* R_t *based on non-dominated sorting*
9	*Let* F *be a subpopulation of individuals with the same rank*
10	*Generate new population* $P_{i+1} = P_{i+1} \cup F_j$ *and* $i += 1$ *until* $\lvert P_{i+1} \rvert + \lvert F_j \rvert < N$
11	*Select* $N - \lvert P_{i+1} \rvert$ *individuals with large crowding distance from* F_j *and add to* P_{i+1}
12	*Generate offspring* Q_{t+1} *from* P_{i+1} *using tournament selection, recombination and mutation*

This strategy employs tournament selection meaning that the population undergoes a competitive selection process to identify the individuals with the best fit. The recombination mechanism which is characterised by random pairing and the exchanging of parts of the 'genome' result in 'breeding' new individuals for the next generation. Mutation is also a key process for selecting the new population as it introduces a degree of randomness which then ensures diversity and adaptability.

2 Surrogate Based Global Search Strategy

The surrogate-based strategy implemented is combining an Artificial Neural Network (ANN) for the surrogate model with a multi-objective genetic algorithm. This strategy attempts to leverage the ability of ANNs to approximate complex, nonlinear relationships between the objective functions which is crucial when the problem is completely black box. ANNs are particularly well suited when the evaluated functions have no explicit mathematical formulations and thus are used as surrogate models. These models have the ability to capture the behaviour of any expensive of computationally intensive function which enable the solution space to be explored more efficiently.

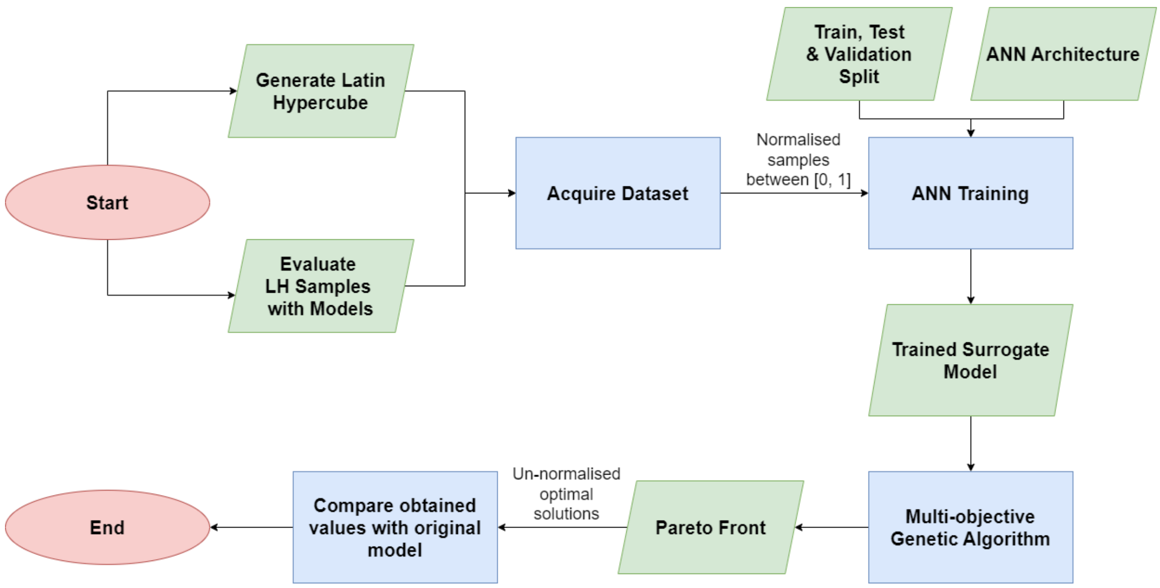

Figure 1 Simplified flowchart diagram of surrogate-based strategy.

The multi-objective genetic algorithm, implemented in MATLAB with the `*gamultiobj*` function, was utilised to obtain the Pareto front from the surrogate model achieved by the ANN. This algorithm was used due to having capabilities for handling multiple objectives functions as well as being able to maintain diversity while moving towards the optimal solutions, and thus obtain the Pareto front based on the fitness values.

ALGORITHM: MULTI-OBJECTIVE GENETIC ALGORITHM

Randomly generate an initial population of solutions
Calculate the fitness of each individual based on the objective functions
***Sort** the individuals using a non-dominated sorting approach*
    Solutions are classified into different fronts based on dominance ranking
***Select** parents to produce offspring*
    Select based on ranking and crowding distance
Pair parents and apply recombination operators to produce new offspring
Apply mutation to the offspring based on percentage distribution
Combine parents and offspring populations to create new generation
***If** new generation exceeds max population*
    Select individuals based on ranking and crowding distance
***Repeat** until max generations reached*
***Return** non-dominated front = Pareto front*

A key factor in employing this strategy is a direct response to the constraint imposed by the problem of 300 function evaluations. Therefore, a method in which the efficacy is maximised for each evaluation was taken. The ANN model makes complete use of this restricted dataset to understand and approximate the relationship between the two objective functions. It is however acknowledged that the production of the dataset has large upfront computational costs, the reusability and versatility of this dataset allows an optimisation strategy that does not go above 300 function evaluations in any respect, other than validating the returned values.

3 Results

3.1 Part A – Find Global Minima of Objective Functions

The method used to find the global minima of each objective function was the Genetic Algorithm, which is implemented using the `*ga*` built-in MATLAB function, where the maximum generations, size of each population, and recombination rate were defined to have control over the search and obtain the best values.

Table 1 Global minima of objective function 1 and computational costs.

F1 Minima		Function Evaluations		Run Time (s)	
0.5219		476050		5.681	
X1	X2	X3	X4	X5	X6
8.002	-39.254	94.151	-785.993	4711.872	-39197.894

Table 2 *Global minima of objective function 2 and computational costs*.

F1 Minima		Function Evaluations		Run Time (s)	
4754.9		467500		5.761	
X1	X2	X3	X4	X5	X6
-7.859	35.001	-157.022	786.100	-2356.668	47116.894

3.2 Part B – Non-surrogate based Strategy on Auxiliary Model

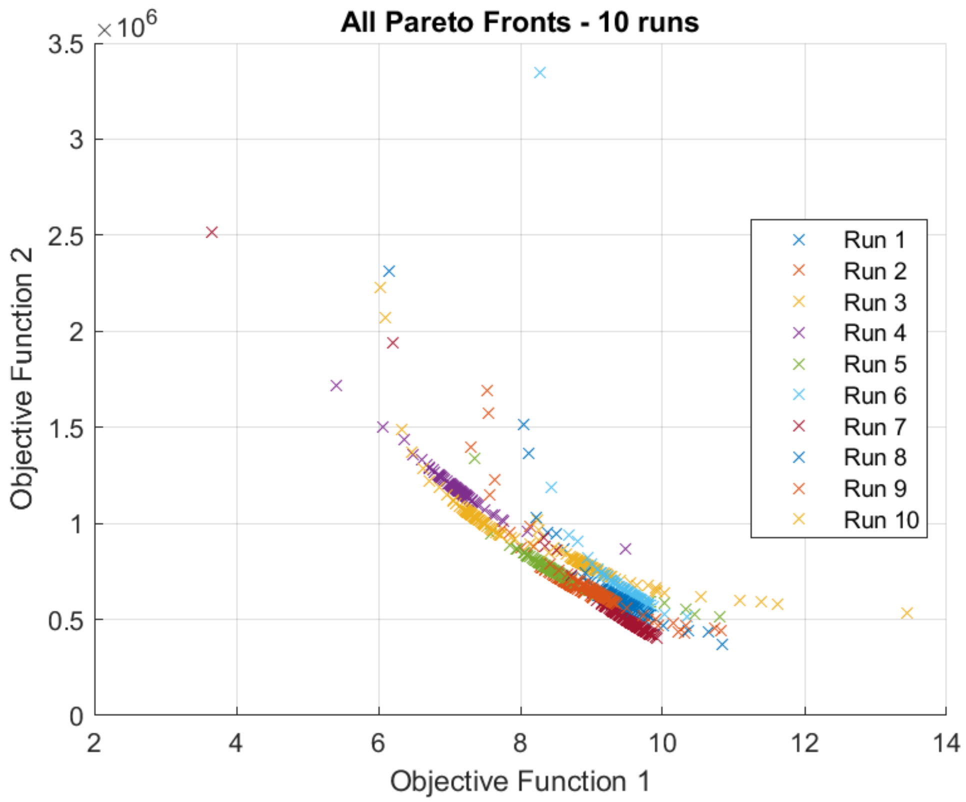

Figure 2 Plot of all Pareto fronts obtained across all runs of non-surrogate NSGA-II strategy.

Table 3 Computational cost across all runs for non-surrogate NSGA-II strategy.

Total Run Time, 10 Runs (s)	Total Function Evaluations
1419.865	301200

The total run time was 23 minutes and 40 seconds, giving an average of 2.5 minutes per run.

Table 4 Function evaluations per run for non-surrogate NSGA-II strategy.

Run	Function Evaluations per Run
1	30120
2	30120
3	30120
4	30120
5	30120
6	30120
7	30120
8	30120
9	30120
10	30120

3.3 Part C – Surrogate based Strategy on Auxiliary Model

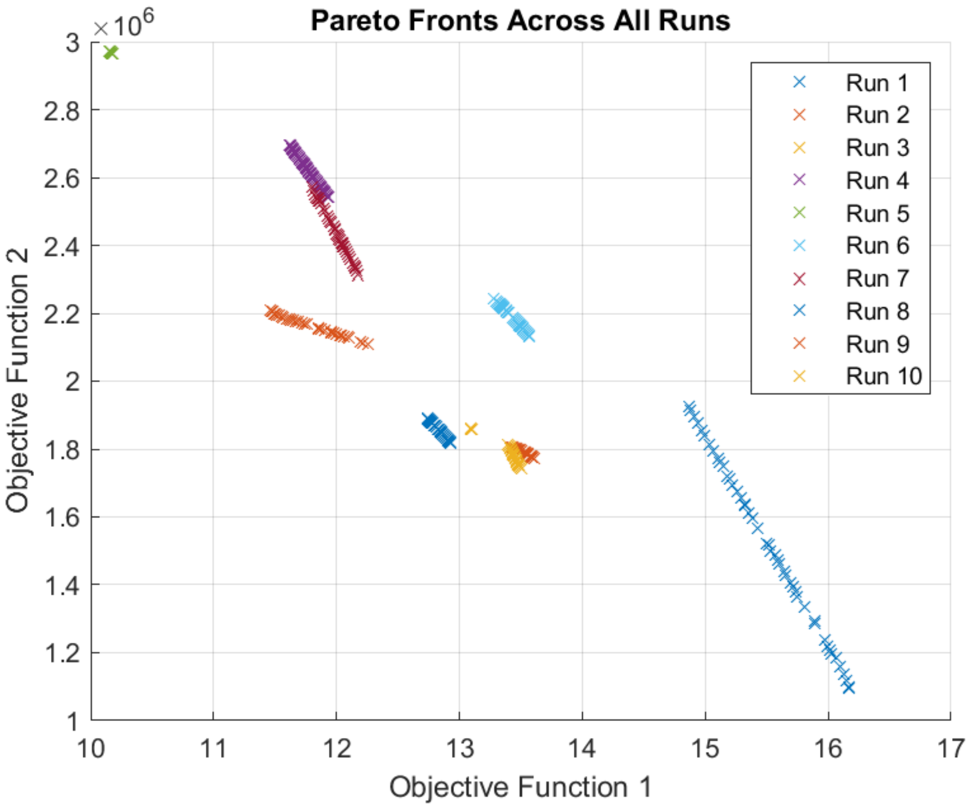

Figure 3 Plot of all Pareto fronts obtained across all runs of surrogate ANN strategy.

The total run time was 11 minutes and 52 seconds, giving an average of 1.2 minutes per run.

Table 5 MSE between predicted and testing values

Run	Mean Squared Error
1	3.07E+11
2	2.38E+11
3	1.92E+11
4	1.70E+11
5	2.95E+11
6	3.16E+11
7	1.73E+11
8	4.71E+11
9	2.39E+11
10	1.17E+11

Table 6 MAE between predicted and testing values

Run	Mean Absolute Error
1	2.69E+05
2	2.56E+05
3	2.32E+05
4	2.30E+05
5	2.81E+05
6	3.00E+05
7	2.26E+05
8	3.70E+05
9	2.68E+05
10	1.75E+05

Table 7 Computational cost across all runs for surrogate ANN strategy.

Time to Gather Dataset (s)	Total Run Time, 10 Runs (s)	Total Function Evaluations
4103.802	712.299	3000

Table 8 Function evaluations per run for surrogate ANN strategy.

Run	Function Evaluations per Run
1	300
2	300
3	300
4	300
5	300
6	300
7	300
8	300
9	300
10	300

3.4 Part D – Surrogate based Strategy on Expensive Model

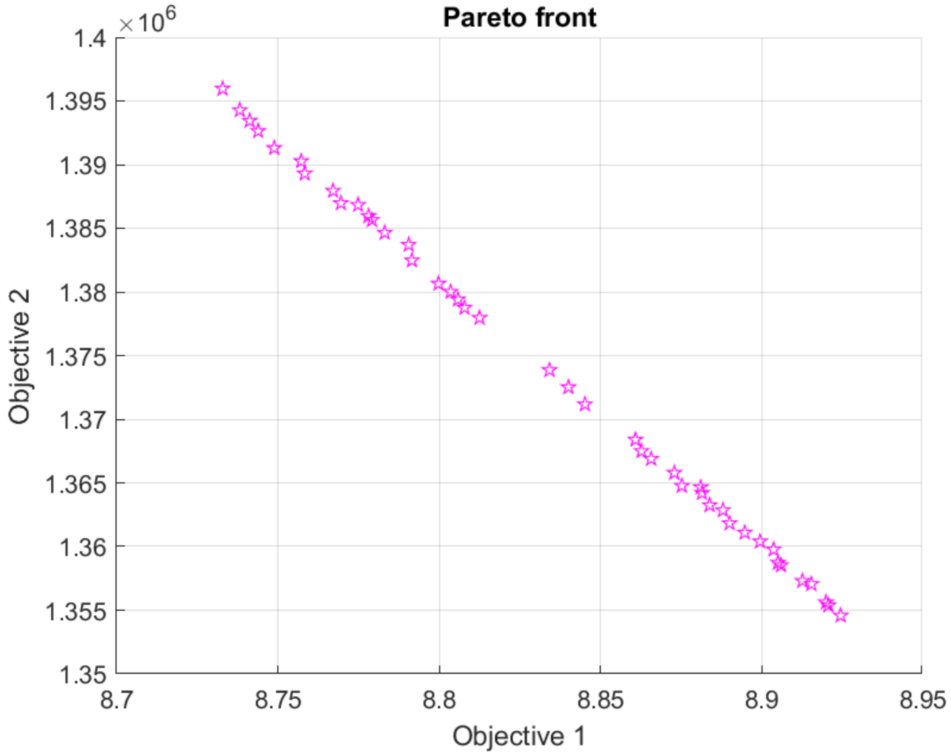

Figure 4 Pareto Front obtained from surrogate model.

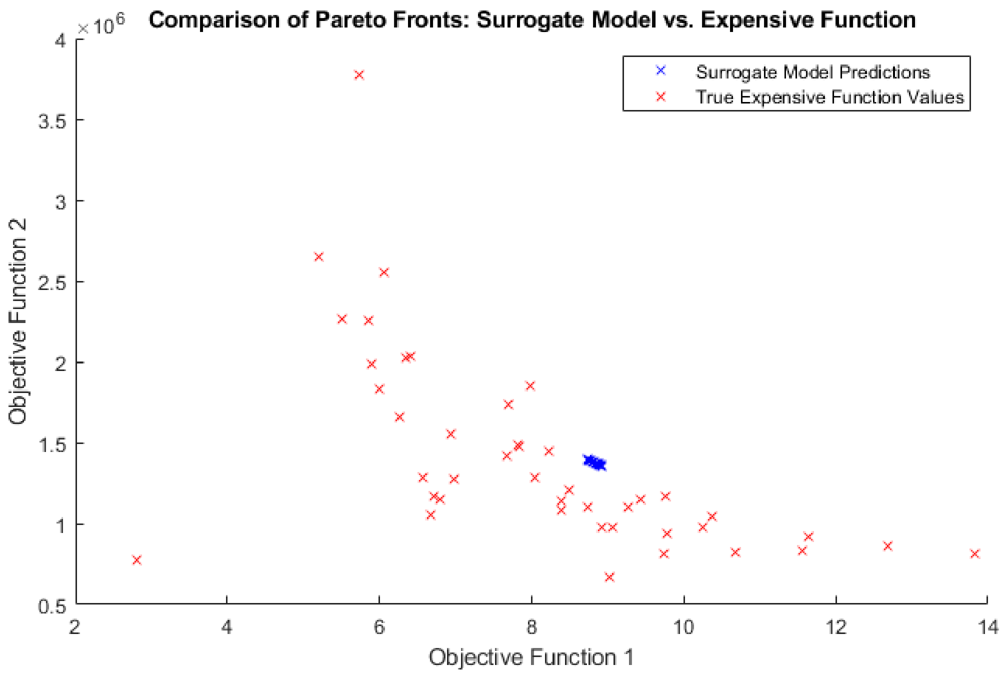

Figure 5 True values versus surrogate model Pareto front.

Table 9 Computational costs of surrogate strategy

Total Run Time, 10 Runs (s)	Total Function Evaluations
3900.194	300

The total run time was 1 hour and 5 minutes, to validate the Pareto front values. The time to obtain the dataset was not recorded official using MATLAB, however, it was approximately 12 Hours as parallel processing via `*parpool*` was employed to speed up the function evaluations.

4 Discussion and Conclusion

4.1 Finding the Global Minima of each Objective Function

The results from this routine were obtained after systematically varying the maximum generations, number of individuals in a population and the recombination percentage, this allowed for manual searching of the solutions space. It was found that minima gathered in Table 1 and Table 2, were substantially better than the initial minima obtained from the routine. However, it is important to highlight that these minima values would have large trade-offs in the opposing objective function as is highlighted in later searches.

4.2 Non-surrogate based Strategy on Auxiliary Model

The results obtained from the NSGA-II algorithm exhibit a degree of accuracy as there is clustering within the same region, which suggests that the "best" Pareto front lies within that region. However, there are points which are outlier to this clustering, although this may be attributed to the inherent stochastic nature of NSGA-II and the crowding mechanism integral to the searching process. This shows that NSGA-II is successful in converging towards regions of the Pareto front that offer an optimal trade-off between the objective functions whilst maintaining diversity with the isolated individuals. This approach lends well in comparison to the surrogate model in the context of the auxiliary model as the non-surrogate method has the ability to exploit the true objective space without the need of approximating the model.

4.3 Surrogate based Strategy on Auxiliary Model

The results obtained from the ANN surrogate model paired with the multi-objective genetic algorithm, aimed to approximate the Pareto front efficiently. However, the results obtained exhibit far less clustering than the non-surrogate model. Although, this may be potentially due to several factors of the implementation process and the limits of the dataset (300 samples for each run). In terms of a limited dataset, this limits the generalisation and approximation of the model which in turn leads to inaccuracies in the predictions, overfitting or underfitting, and overall reduced model complexity. Another factor that could have led to a less than optimal surrogate model is the architecture of the ANN and hyperparameter tuning, a thorough investigation of these configuration settings would be required to improve the overall model. The training algorithm used `*trainrp*` was chosen as from initial testing it showed good approximation compared to other training algorithms available, however, after running the routine over all the runs, it did not return satisfactory results.

From the statistical metrics, the surrogate models seem to perform better in terms of the MAE metric compared to the MSE metric suggesting that the predictions are closer to the real value but may not perform very well on outliers.

4.4 Surrogate based Strategy on Expensive Model

The results from the surrogate strategy on the expensive model also returned unsatisfactory results shown in Figure 5. However, there is an underlying trend that can be seen from the true values. Again, this could have been improved in the same matter as the auxiliary surrogate model, although, it may require a completely different configuration as the relationship between the auxiliary and expensive function is completely unknown.

5 Appendix

The NSGA-II script was based on this implementation of the algorithm [2]. The scripts are run ready given the following packages are installed: Global Optimisation Toolbox, Deep Learning Toolbox, Parallel Computing Toolbox.

6 References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182-197, Apr. 2002. [Online]. Available: https://doi.org/10.1109/4235.996017 (Accessed 28/03/24)

[2] Yarpiz, "NSGA-II (Non-dominated Sorting Genetic Algorithm II)," Yarpiz.com. [Online]. Available: https://yarpiz.com/56/ypea120-nsga2 (Accessed 28/03/24)