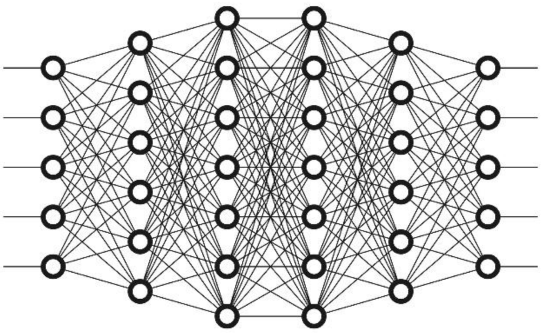
ME527

Introduction to Engineering Optimisation

2021-22 Coursework Report

Bi-Objective Optimisation of Expensive Functions

Jacob Currie – 201718558



The accompanying MATLAB scripts and relevant data produced for this coursework are uploaded alongside this report.

1577 Words

Strategy Descriptions

Part A: Non-surrogate based global search strategy

A stochastic evolutionary algorithm, specifically a controlled, elitist genetic algorithm, a variant of "Non-dominated Sorting Genetic Algorithm-II" (NSGA-II), was used as the non-surrogate global search strategy. This algorithm considers both the score or "rank", and the diversity of the population, as it uses elitism, and crowding, as selection operators. The elitism operator is a guarantee that the best solution of any generation is carried forward to the next generation, while the crowding operator encourages diversity in the population, by including population members of lower rank if they have a suitable crowding distance. The routine is described below in an algorithmic form.

1. *Start*
2. *Create initial population*
3. *Evaluate objective function at initial population*
4. *Rank initial population*
5. *Perform selection procedure*
6. *Perform crossover procedure*
7. *Perform mutation procedure*
8. *Evaluate objective function at child population*
9. *Combine child and parent population*
10. *Rank new population*
11. *Select from new ranked population the next generation*
12. *Check if stopping criteria has been met*
13. *If not met: repeat from line 5*
14. *If met:*
15. *End*

In this strategy, the selection procedure is a tournament selection where subsets of the population are compared the best member is selected. The crossover procedure is used to combine two population members into a child, in this strategy a random selection crossover procedure is used. The mutation procedure introduces small random alterations to create population diversity, here a random mutation alteration that also accounts for feasibility and bounds is used. The combination of the parent and child population is done using the elitism and crowding distance operators, to create the next generation. The initialisation of *a priori* hyperparameters such as population size is not mentioned above but was of course set at the start of the routine.

Part B: Surrogate based global search strategy

A Kriging surrogate was chosen for the surrogate global search strategy, with the variant NSGA-II genetic algorithm mentioned above used to find the minimum of surrogate. Evolution Control, specifically, individual-based control utilising a hybrid of best strategy and random strategy for individual selection, was used to update the Kriging surrogate and avoid convergence to false minima.

As this is a bi-objective optimisation, a separate Kriging surrogate model was created for each objective, for simplicity, referring to the surrogate henceforth describes an encapsulation of both single objective surrogate models combined, as the population and routine used to create both was always identical.

A number of control variables were used in Evolution Control process in an attempt to avoid convergence to false or local minima, the number of allowed generations for the genetic algorithm was increased with each “no new point” generations, allowing for more effort in finding the minimum of the surrogate model as it grew more complex or reached possible convergence. A variable that assesses the number of iterations that have passed where no new suitable points to add to the surrogate were found was used as a measure of convergence, however it is possible this is a local minimum, therefore upon convergence the iterations continue with a new initial population for the genetic algorithm also. The routine of the strategy is described in algorithmic form below.

```
1.  Start
2.  Create initial population
3.  Evaluate initial population using true function
4.  Create initial surrogate model

5.  Begin iterative surrogate improvement (Evolution Control)
6.      Generate new initial population for GA
7.      Find surrogate minimum with GA
8.      Update population using EC individual-based selection routine

9.      Check if new points are found to update surrogate:
10.             If yes:  Create new updated surrogate with new points
11.                      Reset log of iterations where no new points were found
12.             If no:   Log that no new points were found this iteration
13.
14.     Increment GA allowable generations
15.     Check if number of “no new point” generations has reached threshold:
16.             If no:   repeat from line 7
17.             If yes:  repeat from line 6 with new GA initial population
18.     Check if stopping criteria are met:
19.             If no:   continue with Evolution Control iterations
20.             If yes:
21. End
```

The individual-based control with selection subroutine is described in algorithmic form below. When comparing points, the magnitude difference of the variables in the design space is used. A control variable is used for the maximum number of points that can be added per surrogate update, as well as the number of points that have been added to the surrogate list. The random ordering of GA population point consideration is essential to eliminating bias in the choice of surrogate fitting points.

1. *Start*
2. *For each objective:*
3. *Check the minimum difference between found objective minimum and the surrogate points:*
4. *If below threshold: Continue*
5. *If above threshold: Add found point to surrogate design point list*
6. *Evaluate found point using true function and add to list*

7. *For each design space point on the found front:*
8. *Check the minimum difference between surrogate points and the front point:*
9. *If below threshold: Continue*
10. *If above threshold: Add front point to surrogate design point list*
11. *Evaluate front point using true function and add to list*

12. *Check if maximum number of added points is reached:*
13. *If no: Continue*
14. *If yes: Stop checking found front points*
15. *Check if number of added points is below a certain threshold:*
16. *If yes: For every design point in GA population (In random order):*
17. *Check minimum difference of surrogate points to selected GA point:*
18. *If below threshold: Continue*
19. *If Above threshold: Add population point to surrogate point list*
20. *Evaluate point using true function and add*
21. *Check if maximum number of added points is reached*
22. *If no: Continue*
23. *If yes: Stop checking GA population*
24. *If no:*
25. *End*

The Kriging surrogate model is a modified radial basis function approach to function approximation, with the addition of per-variable control over the width and exponent of the basis function, in this approach, a gaussian kernel, and zero-order polynomial regression model are used to construct the surrogate model.

Results

Part A: Non-Surrogate Approach

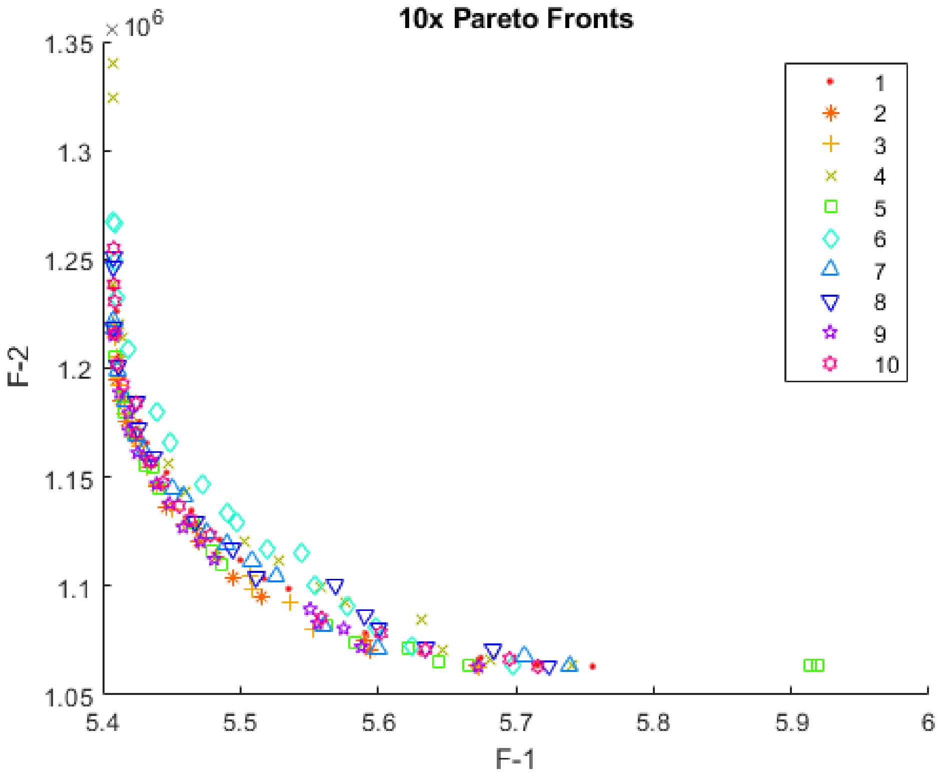


Figure 1 - Part A - Collection of 10 Pareto Fronts (Non-Surrogate)

Total time: 8 seconds (per run)

Function Evaluations: 50,000 (per run of 10)

Part B: Surrogate Approach on Auxiliary Function

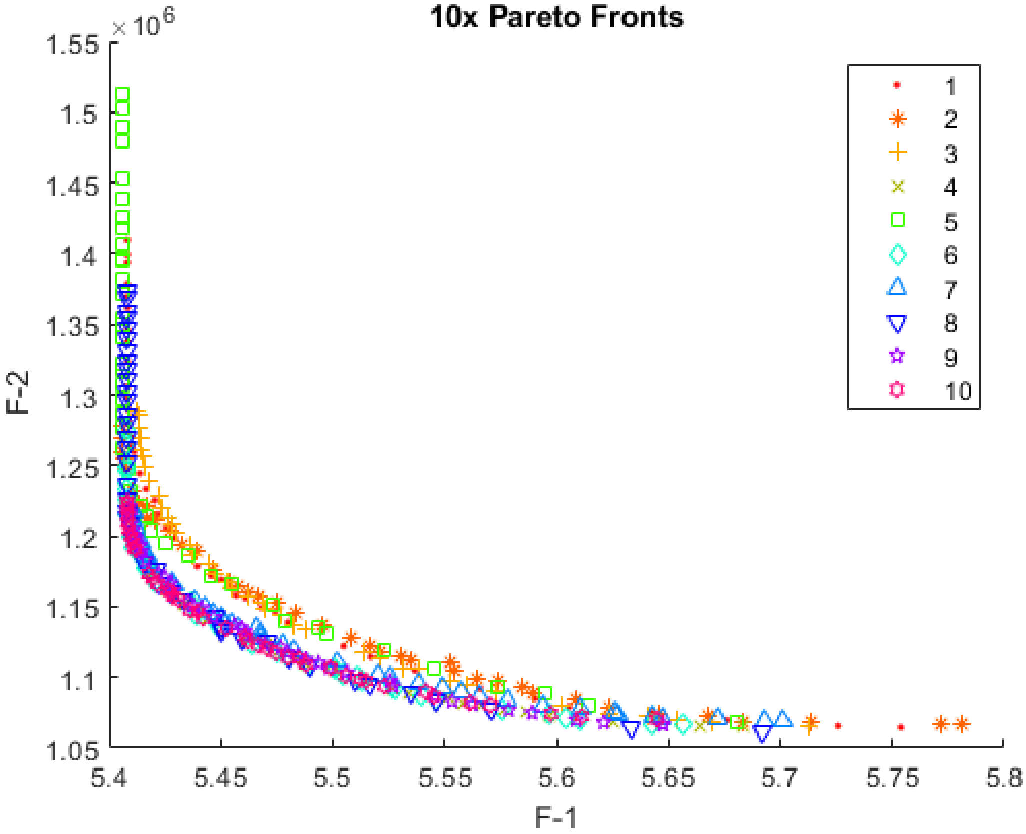


Figure 2 - Part B - Collection of 10 Pareto Fronts (Surrogate)

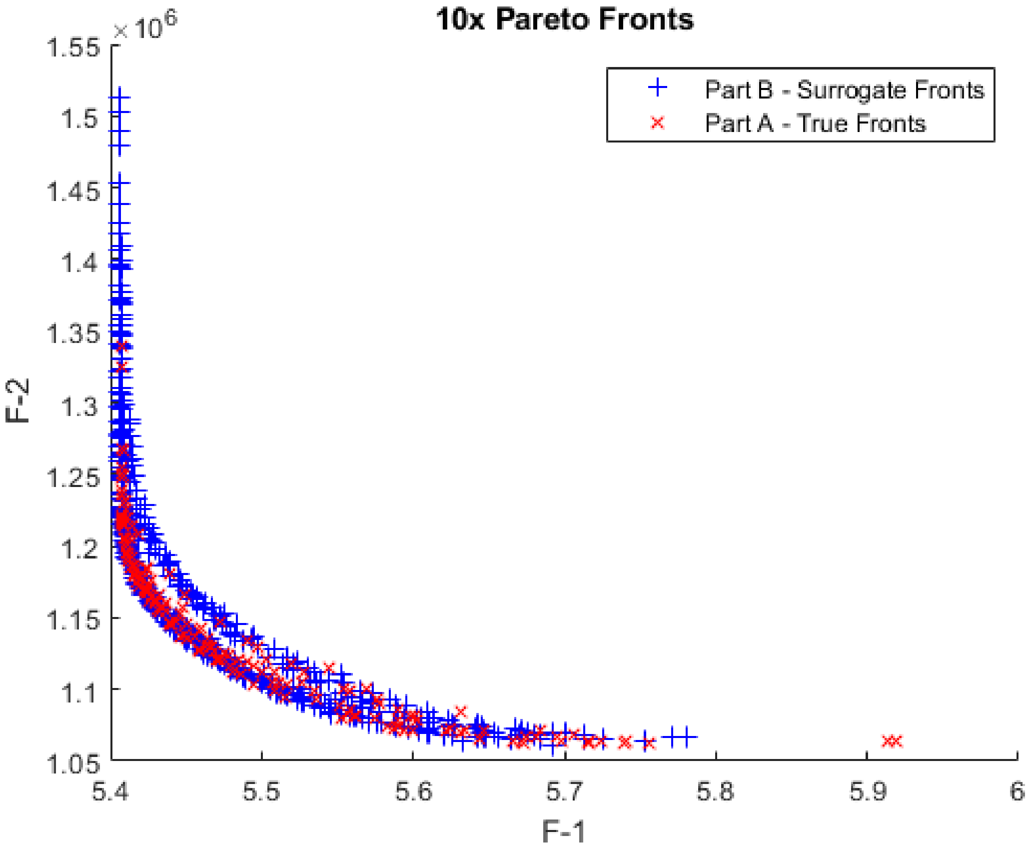


Figure 3 - Part A/B Comparison - True and Surrogate Fronts

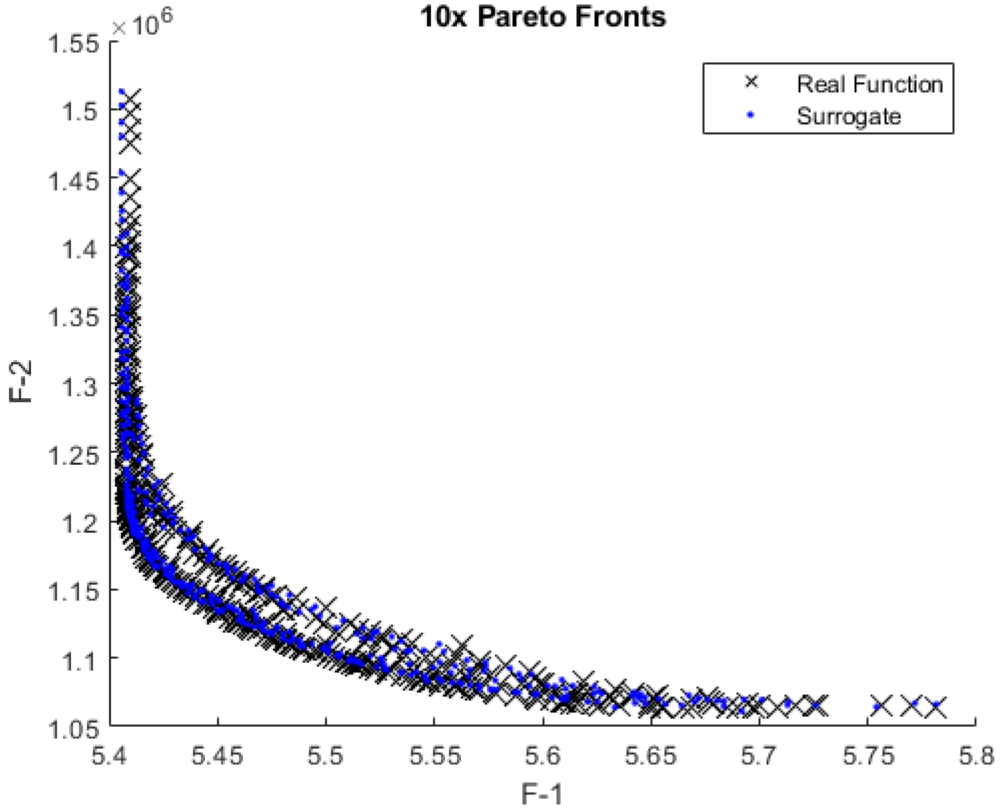


Figure 4 - Part B – Collection of Fronts Comparison (Surrogate found points evaluated on true function)

Total time: 334 seconds (average per run).

Function Evaluations: Below 300, Shown below.

Table 1 - Function Evaluations used for Part B, for each run of the 10

Run	1	2	3	4	5	6	7	8	9	10
Function Evaluations	150	246	268	197	188	177	225	196	218	237

Part C: Surrogate Approach on Expensive Function

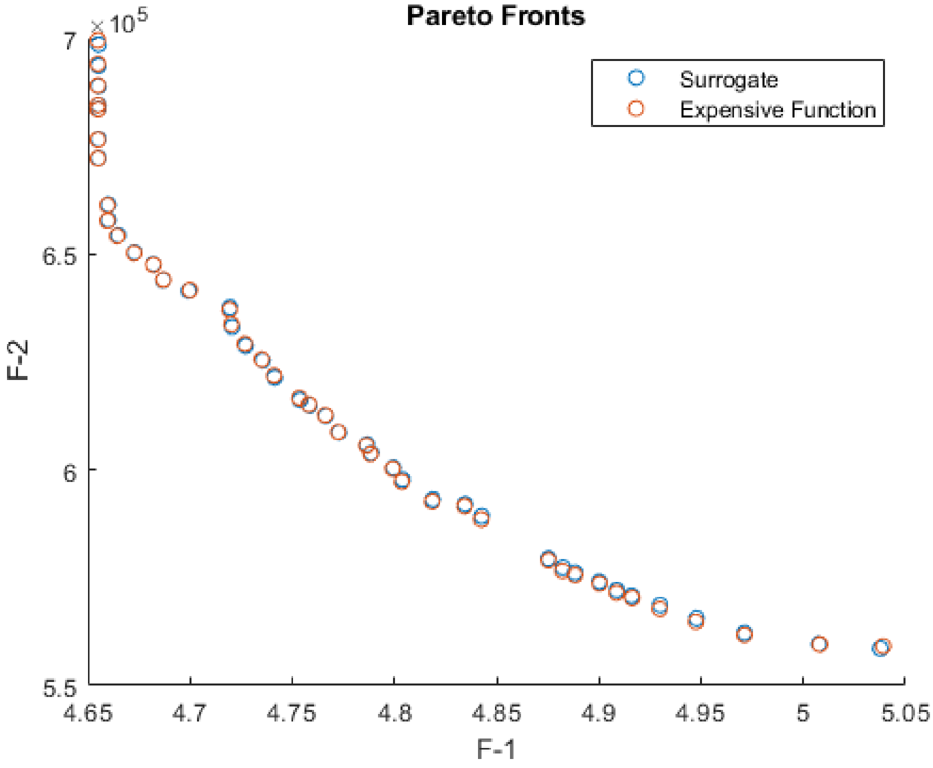


Figure 5 - Part C - Expensive Function - True and Surrogate Fronts

Total time: 79550 seconds (22.1 hours).

Function Evaluations: 280.

18 of the 280 expensive function evaluations were used for the initial design of experiment, contributing to the initial design points chosen via a Latin hypercube, these initial points could be executed in parallel, shortening the run time. A table of the exact values is shown in Appendix 1.

Some additional statistical metrics of the performance are shown in the table below.

Table 2 - Statistical Metric Results for Comparison of Surrogate and Expensive Function

Variable	Mean Square Error	Root Mean Square Error	Correlation Coefficient
F1	1.85436E-07	0.000430623	0.999996178
F2	208791.2	456.9368	0.999957

Discussion

Part A

Observing the results from part A (figure 1), the consistency across the 10 validation runs confirms that the stochastic method used is reliable for this problem, giving confidence in the results. The very fast run time of the routine can be attributed to the speed of computation of the objective function, with high number of function evaluations of 50,000, it can be said that this strategy would not scale well to more computationally expensive problems.

Part B

Taking the results from part A as the "ground truth" true front of the function, comparing the results of Part A and B in figure 3, it can be stated that the surrogate approach was successful in finding the global minimum/front, showing that the strategy is effective. Observing figure 4, a comparison between the surrogate pareto fronts, and the values of the objective function at the same design points, the clear similarity and correlation can infer that the surrogate optimisation successfully and accurately captured the behaviour of the true function. The results from the surrogate approach for this problem function show a clear collection of extreme points that do not necessarily add information to the approximation of the pareto front, rather they lie horizontally and vertically at the front's extrema, implying that while the front was found successfully, the strategy can be further improved and tuned for more efficient and optimal performance.

Part C

Observing the results in figure 5, it is clear that the strategy developed in part B, was effective in modelling the behaviour of the expensive function, it cannot be proven whether this found front is the global optimum, but the strategy was successful in accurately locating a local (possibly global) pareto front. The performance metrics in table 2 indicate just how successful the surrogate was in emulating the expensive function.

Considering the following analysis of computational cost, with a known expensive function evaluation time of 300 seconds, a basic calculation shows that 78,900 seconds, or 99.2% of the total computational time was spent evaluating the expensive function. This shows that the time spent outside the expensive function was truly trivial in comparison, highlighting the effectiveness and need for surrogate-based approaches that reduce the number of function evaluations. For perspective, the non-surrogate approach of part A would have taken almost half of 1 year to compute.

Additional

The creation of the Part B and C MATLAB routines that implement the strategies draw heavily and are sculpted from the example tutorial routines provided by Dr. Edmondo Minisci as part of this class's available resources[1], additionally, these routines make use of the DACE library for MATLAB that implements routines for the construction of Kriging models[2].

Appendix 1: Table of Part C (Expensive Function) Results Values

Input Values						Expensive Function		Surrogate	
X1	X2	X3	X4	X5	X6	F1	F2	F1	F2
0.999979	0.65375	3.12E-05	0.852177	2.12E-05	0.597672	5.039052	559131.5	5.037633	558580.2
0.999999	0.659656	2.43E-05	0.785027	0.991074	0.471	4.654902	699565.5	4.655232	698585.6
0.999986	0.656213	0.000256	0.818058	0.933067	0.505264	4.719369	637085.2	4.719501	637621.6
0.999925	0.65632	0.000103	0.812251	0.019648	0.513736	4.726848	629186.7	4.72724	628732.2
0.999979	0.65375	3.12E-05	0.852177	2.12E-05	0.597672	5.039052	559131.5	5.037633	558580.2
0.999999	0.659649	2.42E-05	0.785025	0.740541	0.470904	4.654904	676533.6	4.655243	676643
0.999999	0.65966	2.43E-05	0.785091	0.897037	0.470911	4.654903	689058.7	4.655236	688936.9
0.999954	0.65576	0.00011	0.815568	0.005042	0.5175	4.741071	621840.8	4.741409	621381.9
0.999974	0.658959	3.10E-05	0.818578	0.529634	0.476362	4.687033	644070.7	4.686905	643963
0.999973	0.654529	0.000257	0.827415	0.001708	0.584212	4.947449	564762.2	4.947697	565586.7
0.999944	0.65523	0.000136	0.815468	0.01193	0.524327	4.758152	615141.7	4.758488	614959
0.999942	0.654377	0.000172	0.817485	0.010078	0.537788	4.799154	600245.8	4.799456	600469.9
0.999999	0.65966	2.44E-05	0.785025	0.943339	0.470832	4.654906	694104.1	4.655235	693657
0.999985	0.65865	0.000103	0.797742	0.512184	0.47263	4.659619	657734.9	4.660069	657889.2
0.999978	0.654064	8.35E-05	0.834761	0.001116	0.565269	4.916217	570316.7	4.916061	570838.3
0.999985	0.659013	2.96E-05	0.802761	0.563971	0.475874	4.664386	654223.2	4.664835	654382.9
0.999901	0.655627	0.00014	0.812334	0.03374	0.510367	4.720207	633709.4	4.720596	633076.6
0.999999	0.659659	2.42E-05	0.785072	0.848616	0.470908	4.654903	684522.5	4.655238	684574.8
0.999993	0.656305	2.70E-05	0.823758	0.510995	0.480963	4.699961	641616.8	4.699395	641516.9
0.999961	0.654349	0.000109	0.816495	0.00335	0.521521	4.753465	616650.2	4.753799	616194.3
0.999968	0.654146	5.52E-05	0.831688	0.001473	0.553708	4.875345	579020.5	4.875276	579435.6
0.999939	0.655864	0.001488	0.817999	0.009446	0.525428	4.76611	612661.5	4.766387	612532.7
0.999978	0.653966	6.61E-05	0.83547	0.000669	0.586176	4.971205	561686.4	4.971195	562224.4
0.999949	0.659376	0.000185	0.816094	0.635595	0.472463	4.682084	647508.6	4.682018	647638.1
0.999971	0.654929	5.51E-05	0.806845	0.01705	0.520062	4.735078	625563	4.735462	625371.5
0.999962	0.654839	4.94E-05	0.830984	0.002232	0.558899	4.888276	575607.6	4.88828	576191.3
0.999967	0.654125	4.76E-05	0.829945	0.003624	0.566681	4.908636	571481.1	4.90873	572115.6
0.999977	0.654109	6.88E-05	0.833883	0.00141	0.560458	4.900321	573599.1	4.900179	574082.9
0.999872	0.655146	8.76E-05	0.818919	0.019471	0.549161	4.834266	591563.2	4.834525	592061.4
0.999964	0.654208	3.69E-05	0.822887	0.002614	0.580617	4.929734	567739.4	4.929964	568646
0.999955	0.655721	0.000204	0.816392	0.00404	0.540348	4.803616	597243.9	4.8039	597761.1
0.99994	0.655585	0.001452	0.817713	0.006377	0.533795	4.788293	603675.9	4.788575	603885.2
0.999999	0.659655	2.43E-05	0.78499	0.652381	0.470856	4.654906	672200.7	4.655248	672247.6
0.999979	0.653889	0.000198	0.84336	0.000734	0.594164	5.008067	559528.3	5.007582	559605.1
0.999959	0.654532	8.88E-05	0.816029	0.004913	0.529184	4.772428	608743.4	4.77276	608701
0.999937	0.656109	0.00019	0.81443	0.013945	0.55479	4.842481	588475.1	4.842686	589282.3
0.999999	0.65966	2.48E-05	0.785106	0.837519	0.47089	4.654903	683537.2	4.65524	683615.6
0.999972	0.654042	0.000982	0.821545	0.002641	0.563444	4.882129	576535.5	4.88238	577331.2
0.999976	0.659017	0.001183	0.793816	0.478445	0.477539	4.659559	661328.8	4.660032	661489.5
0.999997	0.656936	3.87E-05	0.808674	0.578674	0.478005	4.672598	650230.3	4.672823	650417.7
0.999938	0.655989	0.000913	0.816998	0.016706	0.533778	4.786318	605614.2	4.786604	605833.2

0.999944	0.656191	6.03E-05	0.822035	0.002647	0.542008	4.818466	592638	4.818676	593140.1

This table of results is included in the files uploaded alongside this report as an Excel file.

References

[1] Dr. Edmondo Minisci, “Tutorial on ‘Basic SBO’ (via Evolution Control).” University of Strathclyde - MyPlace, 2022.

[2] S. N. Lophaven, H. B. Nielsen, and J. Sondergaard, “DACE - A Matlab Kriging Toolbox,” 2002, [Online]. Available: https://omicron.dk/dace.html.