University of Exeter coursework header sheet

ECMM102 Group Project (Meng) (A, TRM1+2 2016/7)

023706

1027442

620038192

Coursework: Individual contribution to the group achievement
Submission Deadline: Thu 4th May 2017 12:00
Personal tutor: Dr Hao Qin

Marker name:	G_Tabor
Word count:	7484

First marker's comments	Indicative mark
Second marker's comments	Second mark
Moderator's comments	Agreed mark

I2 Report

Implementation of Bayesian Optimiser on CFD Modelled Draft Tube

Wan Lun Carol Ng

2016

4th year MEng Group Project

I certify that all material in this thesis that is not my own work has been identified and that no material has been included for which a degree has previously been conferred on me.

Signed... [illegible]

College of Engineering, Mathematics, and Physical Sciences
University of Exeter

I2 Report

ECMM102

Title: Implementation of Bayesian Optimiser on CFD Modelled Draft Tube

Word count: 7484

Number of pages: 31

Date of submission: Thursday, 04 May 2017

Student Name: Wan Lun Carol Ng

Programme:

MENG Electronic Engineering and Computer Science

Student number: 620038192

Candidate number: 023706

Supervisor: Dr. Gavin Tabor

Abstract

This individual project is under the Optimisation of Hydraulic Draft Tubes through CFD and Machine Learning group project, which aimed to optimise the geometry of the draft tube for a hydraulic turbine. This report covers the steps taken to implement Efficient Global Optimisation (EGO) on Computational Fluid Dynamics (CFD) modelled draft tube.

A more sophisticated and effective global optimisation tool is wanted in the engineering field which could work well with computationally expensive and time consuming simulations like CFD. For this reason, Bayesian optimisation was proposed for its capability to reach an optimal solution with fewer function evaluations. This is achieved by building a probabilistic surrogate model that approximates the objective function as new data is available and directs efficient function evaluation accordingly. With a project aim of optimising a CFD-modelled draft tube with Bayesian optimisation, EGO was implemented on a simple benchmark Branin function, followed by the development of an interface which facilitates automated optimisation happens between EGO and CFD. The geometry representation required for this design problem was computed from the novel Catmull-Clark subdivision curve. The optimiser together with the automated interface were tested on PitzDaily CFD tutorial case before it was brought to optimise the CFD-modelled draft tube. The performance of this optimiser on Branin function and PitzDaily case showed EGO is a more powerful optimisation tool in contrast to Genetic Algorithms. Despite the inaccuracy in model predictions, the optimiser was still capable to optimise the draft tube by 13.57% with pointwise and four control points within 187 function evaluations.

Keywords: Bayesian Optimisation, Catmull-Clark Subdivision Curve, Automated CFD-based Optimisation

Table of contents

1. Introduction and background

As technology advances and machine learning algorithms are becoming more mature, optimisation is heavily implemented in engineering problems to help improve current design by exploring the possibilities over the available search space. For this reason, the field of research on design optimisation using Computational Fluid Dynamics (CFD) emerged and is currently expanding.

Conventionally, CFD design optimisation is trial-and-error based. The geometry changes by constant manual amendment based on the limited amount of previously CFD-analysed results with the use of Computational Aided Design software [1]. In other words, the whole process of CFD design optimisation is time consuming and highly dependent on personal engineering experience. Hence, machine learning algorithms, such as Evolutionary Algorithm, are introduced to CFD design optimisation to search for good solutions within an acceptable time.

Yet, not all machine learning tools work equally-well with CFD-based optimisation. Consider the cost of CFD in terms of both time and computational cost, the total number of evaluations required by the well-known Genetic Algorithm (GA) to optimise a single problem makes GA less appealing. In short, a more sophisticated machine learning optimisation algorithm is wanted as a better alternative for this type of optimisation. On top of the machine learning algorithm for efficient optimisation, geometry representation and parameterisation are essential in automated CFD optimisation. This project looked into the use of novel Catmull-Clark spline geometry representation in automated CFD-based optimisation with efficient global optimisation.

The aim of this individual project was to implement a machine learning tool, Bayesian optimiser, which is capable to optimise the draft tube design from automated CFD simulation. This worked towards the main aim of this group project, which was to optimise a draft tube for low head hydraulic turbine application. Four main objectives were established to achieve this aim. To begin with, a Bayesian optimiser which is capable to optimise simple and benchmark test (Branin function) effectively had to be built. Once an effective optimiser was made ready, the next objective was to prepare an interface for the communication between the optimiser and the CFD process to allow automation, which includes the geometry representation and optimisation parameters. The third objective was to investigate whether Bayesian optimiser or GA is performing better in CFD optimisation case. Eventually, optimisation was undertaken to maximise the efficiency of hydraulic draft tube.

Following the project introduction and background provided above is literature review of previous work. The report will then elaborate the key concepts relevant to this project. Methodology and design of the optimiser will be outlined next, the report will subsequently present and analyse the collected result. Eventually, discussion and conclusions on what has been learnt and future work are presented. Project management and contribution to group functioning are included at the end of the report.

2. Literature review

2.1. Bayesian Optimisation

Emerged as a solution for joint optimisation and global optimisation, Bayesian optimisation is becoming more popular recently. While the room for improvement in massive complex integrated systems design problems is enormous due to the incompetence and challenges in tuning the vast amount of distributed tuneable configuration parameters, Bayesian optimisation is expected to bring optimisation forward [2]. In various automated design problems, such as automatic machine learning [3] [4] and experimental design [5], Bayesian optimisation showcased its potential to enhance the level of automation in myriad of design choices and give instant improvement and innovation.

The framework of Bayesian optimisation is sequential [6] model-based and composed of two fundamental components to allow dynamical design choices to be made as new evidence appears [7]. The first component is a probabilistic surrogate model, followed by a loss function as the second component. The probabilistic surrogate model is an observation model that captures the mechanism of data generation incorporated with prior beliefs about the distribution and behaviour of the unknown objective function. While the probabilistic surrogate model is constantly being refined as more function evaluations are done and the Bayesian posterior distribution are updated, the most efficient and informative sequence of next evaluation search point could be computed by minimising the expected loss.

Benefited from its framework, Bayesian optimisation is an appealing global optimiser. First, the use of probabilistic model extends its application beyond ordinary search spaces to combinational search spaces with categorical and conditional inputs. Second, its highly data efficient framework makes full use of the information from previous optimised evaluations to direct efficient search, hence it is helpful in optimising problems with costly evaluations of the objective function f, where f is nonconvex and multimodal. Third, the loss function could be chosen to automatically leverage exploration and exploitation in sequential queries [7] [8]. Altogether, Bayesian optimisation is a potential strategy to practically global optimise engineering problems which are known to have strict limits on time or cost imposed from the extensive simulation run time.

2.2. Probabilistic Surrogate Model

The probabilistic surrogate model holds the beliefs about the model and directs optimisation [7]. Consider an unobserved model parameter w and available, observed data D, the priori distribution $p(w)$ captures a priori beliefs on the possible values of w before observation. The posteriori distribution $p(w|D)$ which represents the updated beliefs after data observation could be calculated from Bayes' rule as in equation 1.

$$p(w|D) = \frac{p(D|w) \cdot p(w)}{p(D)} \tag{1}$$

where $p(D|w)$ is the likelihood model and $p(D)$ is the marginal likelihood.

Despite more literature and research were focused on the acquisition function, it was suggested the choice of surrogate model plays a primary role before acquisition function [7]. For problems with independently varying components, simple linear models, e.g. with Thompson Sampling [9], could deal with straightforward choices whereas generalised linear models could handle response variables flexibility through link function [10]. An alternative to linear regression model is Design and Analysis of Computer Experiments (DACE) [11] stochastic process model, which models the deterministic output as the realisation of stochastic process and provide prediction as well as its uncertainty from the statistics. While linear regression model focuses on regression and makes assumptions about errors, DACE model focuses on error correlation and makes assumptions about regression [8]. This leads to a general difference in regression model aims to describe the exact function whereas DACE model aims to describe the typical function behaviour.

2.3. Acquisition Functions

Minimisation of loss function, or cut down of wasteful arbitrary searches, becomes particularly crucial when it comes to costly objective function evaluations. Acquisition functions which are comparatively inexpensive to evaluate are developed as strategies of sorting the search sequence in Bayesian optimisation. Among the many selection strategies developed, a popular tribe is to utilise the posterior model, i.e. select the next query point x_{n+1}base on previously evaluated sample data D_n. This tribe of acquisition functions is designed to trade off exploration of the search space and exploitation of current promising areas. Traditional improvement-based acquisition function, optimistic acquisition functions, the more recently developed information-based approaches and the entropy search portfolio are the examples of this type of acquisition function. [7] In fairness, no acquisition function can outperform the others over all problem instances. It has been empirically observed that the preferred strategy can change at various stages of the sequential optimisation process. [7]

The maximum probability of improvement (MPI), a.k.a. the P-algorithm, is an improvement-based acquisition function suggested by Kushner in 1964 [12]. This algorithm considers the probability of improvement $PI(x) = P\big(f(x) \geq f(x^+)\big) = \Phi\left(\frac{\mu(x)-f(x^+)}{\sigma(x)}\right)$, where $x^+$is the point where the current maximum is observed, and $\Phi(\cdot)$ is the normal cumulative distribution function. A potential issue of this formulation is points which potentially enormously greater than $f(x^+)$ will potentially be selected over points which could improve the current best but with a higher uncertainty. In short, this function leads to pure exploitation. For this reason, a trade-off parameter $\xi \geq 0$ is added which modifies $PI(x) = \Phi\left(\frac{\mu(x)-f(x^+)-\xi}{\sigma(x)}\right)$. Maximum expected improvement (EI) is an alternative to MPI which has a similar mathematical framework as specified in section 3.1.2.

2.4. Parallelisation

With achievable multi-core parallelism in modern computation, running Bayesian optimisation parallelly would further boost the efficiency by reducing overall run time. The several parallelisation approaches proposed could be classified into two class: inherently sequential and truly parallel [7]. Sequential approach is to hold a set of current observations and pending experiments, which the pending experiments could be evaluated parallelly, followed by data augmentation [2]. While the truly parallel alternative approach also generates a set of candidate search points, these candidates are generated simultaneously based on GP-UCB to support a range of exploration and exploitation [13].

2.5. PitzDaily

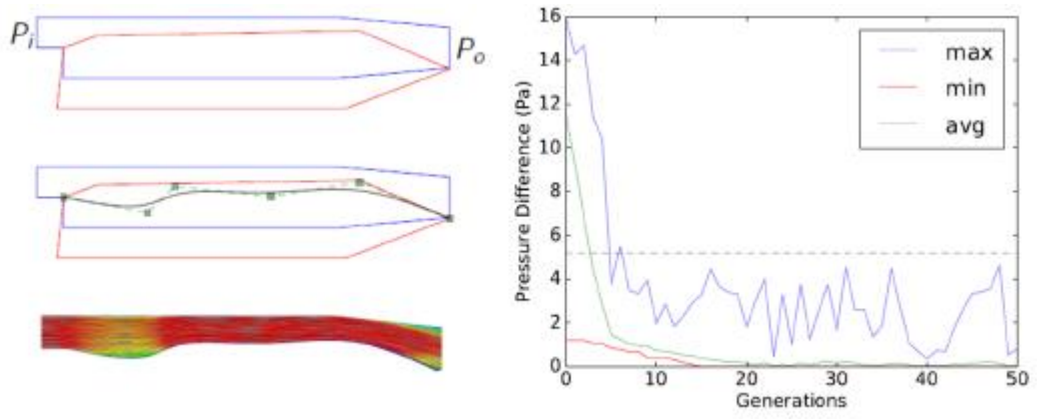


Figure 2-1 Outcome of single-objective optimisation of reducing the pressure drop in PitzDaily tutorial case with GA. [1]

Previously, automated design optimisation on the PitzDaily tutorial using CFD together with GA was done [1]. PyFoam 0.6.5 acted as the bridge between Python libraries and OpenFOAM. Since their objective was to minimise the pressure difference between the inlet and outlet, their chosen cost function was $f = \min(|p_i - p_o|)$, where p_i and p_o are the averaged pressures over the inflow and outflow boundary condition faces. By alternating the lower wall of the geometry using sub-division curves, their optimiser achieved an optimised shape which reduced the base pressure drop from $5.22Pa$ to $4.7e^{-6}Pa.$ The optimised shape and the performance of the optimiser are shown in Figure 2-1. Catmull-Clark subdivision spline curve as the geometry representation and the use of PyFoam as an automated procedure implemented in the interface between the optimiser and CFD simulation in this project were inspired by their work.

3. Theoretical background and design

3.1. Efficient Global Optimisation

Efficient global optimisation (EGO) is a well-known version of Bayesian optimisation in the experimental design literature, which is often associated with the use of GP prior to approximate the objective function [7] and with EI as the chosen acquisition function. This optimisation algorithm is particularly suitable for modelling nonlinear and multimodal functions [8]. Below outlined the EGO algorithm:

Algorithm: Efficient Global Optimisation

```
Model Preparation Stage:
    1. Generate ( $no. of\ dimension \times 11 - 1$) [8] initial samples using Latin hypercube
       sampling method.
    2. Evaluate initial samples function values.
    3. Construct GP model with data $\boldsymbol{D_n}$
Optimisation Stage:
    4. For n = 1, 2, …, budget, do:
    5. Select new $\boldsymbol{x_{n+1}}$ by optimising acquisition function $\boldsymbol{EI}$
           $\boldsymbol{x_{n+1}} = \boldsymbol{argmin_x\ EI}(\boldsymbol{x};\ \boldsymbol{D_n})$
    6. Evaluate objective function to get $\boldsymbol{y_{n+1}} = \boldsymbol{f}(\boldsymbol{x_{n+1}})$
    7. Augment data $\boldsymbol{D_{n+1}} = \{\boldsymbol{D_n}, (\boldsymbol{x_{n+1}}, \boldsymbol{y_{n+1}})\}$
    8. Update GP model
    9. End for
```

3.1.1. Gaussian Process

Gaussian process (GP) is a powerful method to perform Bayesian optimisation. As an extension of multivariate Gaussian distributions of a function, GP is a distribution over infinite possible outputs which any finite subset of observations generated follows a multivariate Gaussian distribution [6] [14]. Therefore, GP models are commonly used as mathematical approximations of time-consuming numerical simulators, where all observed data are generalised and treated as a sample from a GP [2]. The intuitive similarity between each pair of data points could be specified by performing kernel trick k [7]. Since both the predictions and marginal likelihood depends on the kernel matrix K, a complex feature mapping matrix is no longer required.

The probabilistic surrogate model used in this project's EGO was constructed by `GPy` [15], a GP framework in Python. The choice of kernel function was the popular Matern 5/2, which is known to offer better performance in high dimension problems [16]. The optimise function in GPy automatically maximises the likelihood of all hyperparameters in the model to construct the most likely model; the predict function computes the mean prediction and the associated variance. These values were used later in the calculation of expected improvement as well as in the model's diagnostic test.

3.1.2. Expected Improvement

Expected Improvement (EI) is an attractive selection strategy designed to balance local and global search, first recorded in literature in 1978. [8] The concept of EI is to determine the uncertainty at $f(x)$ from the mean prediction $\hat{y}$ and standard deviation s returned from the DACE predictor, and its standard error, which reflects the probability that $f(x)$ will be better than the current best f_{best}. In Bayesian Optimiser, this EI function is with respect to the predictive distribution of the Gaussian process [12]. The equation below shows how EI of a minimising problem can be obtained from the expected value at x:

$$E[I(x)] \equiv E[\max(f_{min} - Y, 0)], \text{ where Y is } Normal\ (\hat{y}, s^2).$$

Performing integration by part gives:

$$E[I(x)] = (f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{s}\right) + s\phi\left(\frac{f_{min} - \hat{y}}{s}\right) \tag{2}$$

where f_{min} is the current minimum among evaluated values, $\Phi(\cdot)$ is the cumulative distribution function and $\phi(\cdot)$ is the probability distribution function of the standard normal distribution. [8] [12] It could be derived from equation 2 that a higher EI will be resulted from smaller $\hat{y}$ and larger s [8], i.e. when the predicted function value is small and when the uncertainty is high, thus the trade-off between exploitation and exploration is achievable.

3.2. Geometry Representation

In this project, a simpler and easier approach was adopted to start off. Instead of optimising the entire draft tube structure which could be particularly difficult as a starting point, this project has decided to optimise the lower wall of the draft tube only. As a more straight-forward approach than generation a subdivision surface, the geometry of the lower wall was represented by an extrusion of a subdivision curve formed by one to five control points as specified in section 3.5.

3.2.1. Catmull-Clark Subdivision Curve

Catmull-Clark subdivision was introduced in Computer Aided Design in 1978. It generalised a set of subdivision rules on bicubic B-spline patch subdivision that produces arbitrary control-points meshes. On top of producing a resulting surface which is continuous and smooth in tangent and curvature at ordinary points, the surface is also guaranteed to be continuous in tangent at extraordinary points [17]. For this reason, this method is particularly useful in generating smooth three-dimensional objects.

In the process of subdivision, each segment of the curve is replaced by two segments. Consider the cubic B-spline subdivision mask, $\frac{1}{4}(1 \quad 2 \quad 1)$,

$$\begin{bmatrix} p_1'' \\ p_2'' \\ p_3'' \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{bmatrix}\begin{bmatrix} p_1' \\ p_2' \\ p_3' \end{bmatrix}, \tag{3}$$

Where $\begin{bmatrix} 4 & 4 & 0 \\ 1 & 6 & 1 \\ 0 & 4 & 4 \end{bmatrix}$ is known as the local subdivision matrix S [18] [19].

Equations 4 and 5 below are two underlying rules concluded from equation 3 which determine the position of the new odd and even vertices respectively.

For odd vertices:

$$p_{2i+1} = \frac{p_i + p_{i+1}}{2} \tag{4}$$

For even vertices:

$$\mathrm{p}_{2\mathrm{i}} = \frac{1}{8}\mathrm{p}_{\mathrm{i}-1} + \frac{6}{8}\mathrm{p}_{\mathrm{i}} + \frac{1}{8}\mathrm{p}_{\mathrm{i}+1} \tag{5}$$

With black crosses indicating the control points, Figure 3-1 illustrates how a smooth Catmull-Clark subdivision curve used in the CFD simulation was generated after five iterations of subdivision.

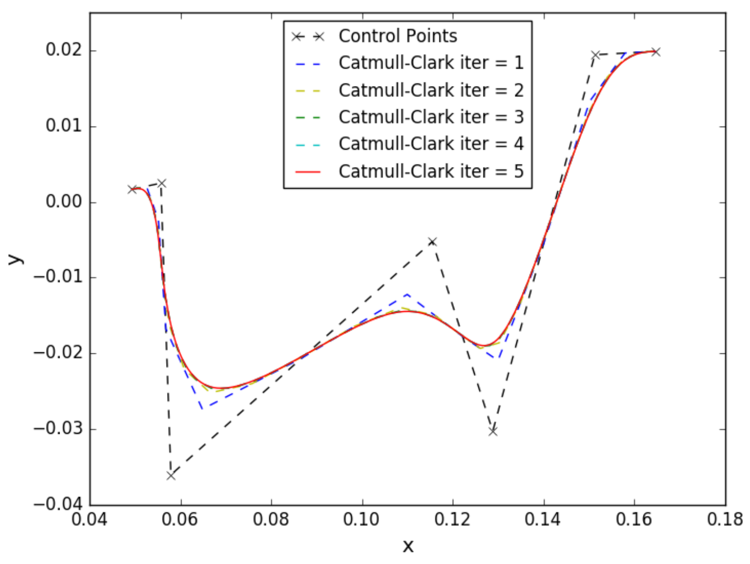


Figure 3-1 Demonstration of Catmull-Clark subdivision.

3.3. Constraints

There were three constraints applied to the geometry representation of the bottom spline:

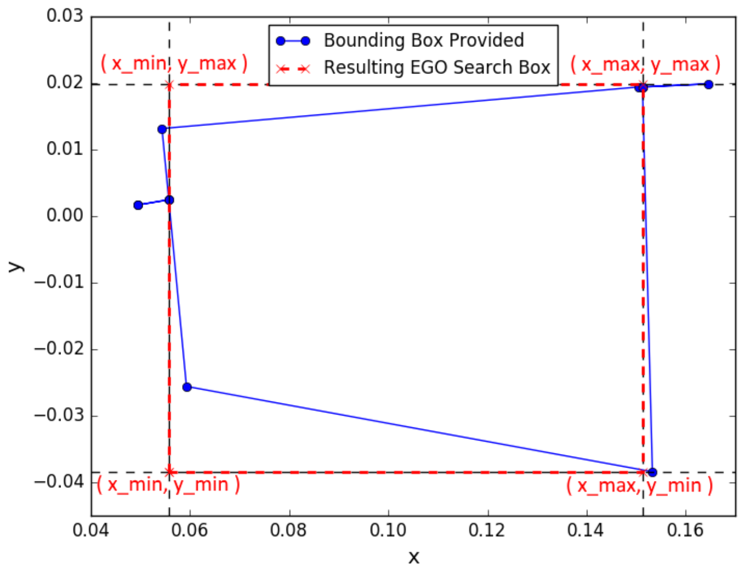


Figure 3-2 Illustration of search box derived from bounding box.

First, a bounding box was essential to define the search area and the boundary of the control points where the resulting spline will be bounded to. A simple technique to set up a search box was illustrated in Figure 3-2. The boundary of x-dimension was the x-coordinate of the two fixed points which mark the entry and exit of the boundary box provided; the lower and upper boundaries of y-dimension were set to the minimum and maximum y-coordinates of the boundary box.

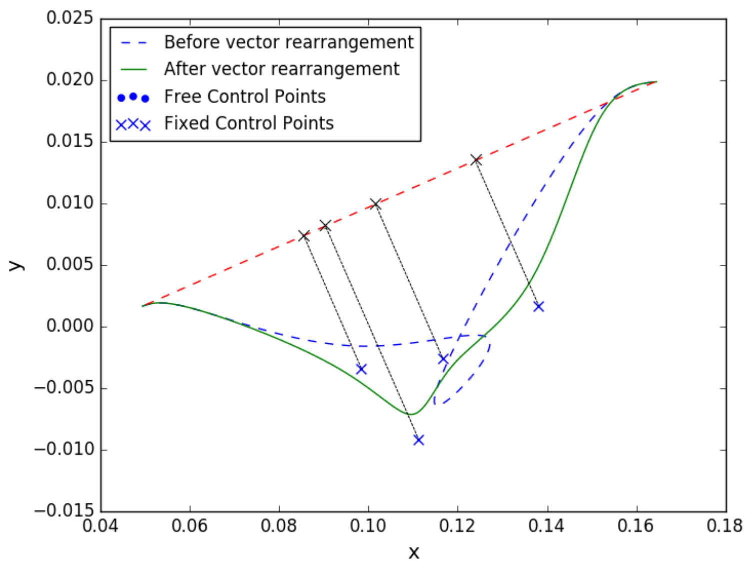


Figure 3-3 Illustration of vectors rearrangement.

Second, with the consideration of both the practicality and the robustness of CFD meshing techniques, it was exceptionally important that the splines must not be intersecting. This could be achieved by rearranging the vectors to make sure the control points are sorted before the spline is generated. Figure 3-3 helps illustrate the vector rearrangement strategy. The freely movable control points were projected orthogonally onto the red dashed line which connects the fixed control points at the two ends. The order could then be sorted out from their order along the line. Following the rearrangement of the freely generated control points within the boundary box, the fixed control points which mark the entrance and exit of the boundary box were added to the beginning and the end of the list of control points respectively to generate a smooth, non-intersecting spline.

Third, an extra fixed point was added to each side of the boundary box as a measure to prevent extreme geometries being generated at the boundary of the search box which the CFD meshing techniques could not handle.

3.4. Genetic Algorithm

Genetic Algorithm (GA), a stochastic optimisation technique inspired by evolution was used to optimise the acquisition function, hence determine the location of control points that worth to search next. The GA implemented in this project ran on algorithm 2 below with parameter settings as listed in Table 1.

Algorithm 2: Genetic Algorithm

```
1.  Generate random initial solution population
2.  Evaluate initial population
3.  for generation n = 1, 2, …, n, do:
4.      Select two parent solutions by tournament selection
5.      Generate two child solutions by crossover
6.      Amend child solutions by mutation
7.      Evaluate child solutions
8.      Update the population
9.  end for
```

Table 1 Parameters used in the optimiser to optimise EI, the acquisition function, with GA.

Parameter	Population Size	Tournament Size	Number of Iterations	Mutation Rate	Crossover Rate
Value	100	3	500	0.1	0.9

3.4.1. Crossover Strategy

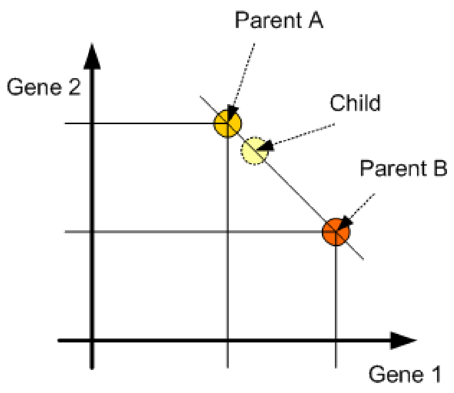


Figure 3-4 Illustration of line recombination crossover [20].

Line recombination crossover was used in the GA optimiser. Unlike traditional crossover which exchanges gene between parents, line recombination crossover belongs to the blend recombination operator which blends the genes [21]. The child's gene was computed from a randomly generated factor on the line linking the two parents' genes as illustrated in Figure 3-4.

3.4.2. Mutation Strategies

As an exploration operator [22], it is more ideal if the mutation operator mutates a gene around its parent value rather than mutates a gene to a completely random location, especially when the parent solutions are already located at near-optimal. In other words, mutation with normal distribution is more favourable than randomisation with uniform distribution. To achieve this, each mutated gene was generated as a normal continuous random variable truncated to the normalised range of its corresponding upper and lower boundaries. The `rvs` method in the `truncnorm` object from `scipt.stats` module [23] was used for this purpose.

Figure 3-5 below shows the relationship between the resulting probability density of the random variables generated by calling the `rvs` method on a gene vector bounded between -5 and 10 and the location of the current gene as well as the scale. It can be seen from the graph that the generated samples are more concentrated at the current gene when the scale is lower. This indicates the degree of changes introduced in mutation can be controlled by the scale. The scale used in this project's GA was selected to be one-fifth of the boundary range.

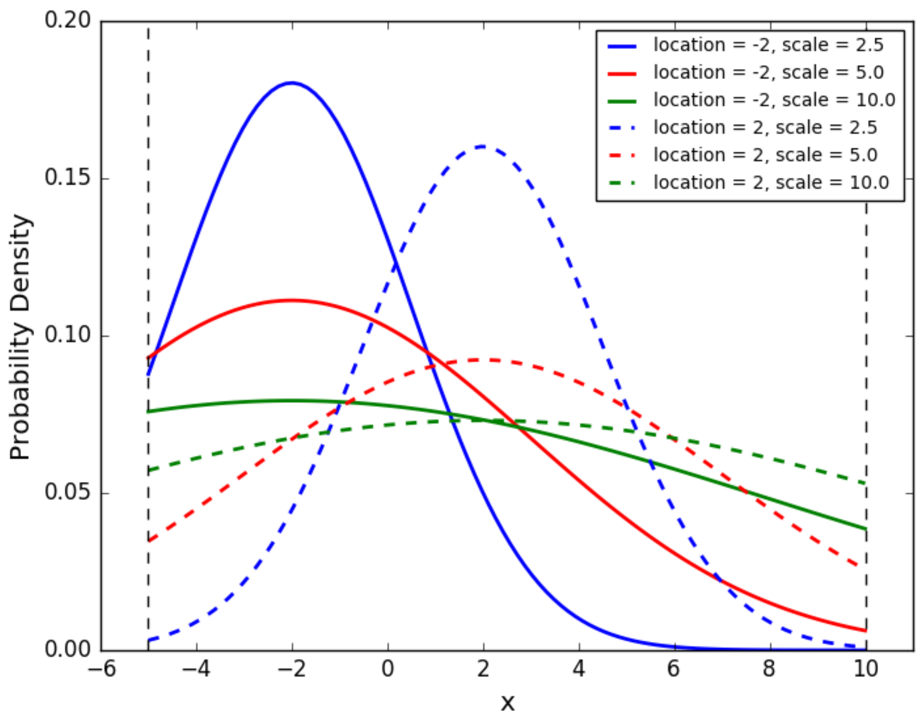


Figure 3-5 Demonstration of scipt.stats.truncnorm.rvs with bounded within the black dashed line.

3.5. Designed Optimisation Test Cases

The optimiser was implemented on three optimisation problem as listed below:

1. Branin function

   The performance of EGO against GA was studied by minimising this benchmark global minimisation problem. Both optimisers ran 10 times on this minimisation problem to ensure the EGO optimiser programed is functioning properly and performing well before moving on to a more complicated, CFD-based PitzDaily case. Both optimisers were set to have initial samples of size 21 and have a total budget of 100 function evaluations.

2. PitzDaily case

   Based on previous work [1], this optimisation case was used as a guide to the automated CFD-optimisation. The objective function of the problem case is to minimise the pressure difference between the inflow and outflow boundaries, which gives a minimising objective function f(x) $= \min(|P_{out} - P_{in}|)$.

   The novel Catmull-Clark spline, generated from three to five control points, was used here to represent the geometry of newly altered design. 10 runs of EGO and GA on this PitzDaily case were called with the purpose to study the performance of EGO against GA on more complex optimisation problems. The budget on function evaluation was set to be $(\,no.of\ dimension \times 11 - 1) + 125$.

3. Draft Tube Model

   Following the testing on geometry and parameter setup in PitzDaily, the optimiser was run to optimise the draft tube. The objective function is to maximise the pressure recovery factor Cp, which is directly related to the pressure difference between the inflow and outflow boundary conditions P_{in} and P_{out}, resulting in an objective function f(x) = max($P_{out} - P_{in}$). Since the returned function value from the CFD simulation was $P_{in} - P_{out}$, therefore the objective function of this draft tube optimisation problem was equivalent to f(x) $= \min(P_{in} - P_{out})$.

The bounding box set for this optimisation case was shown in Figure 3-6, which only optimise bottom of the diffuser and was chosen with the consideration of the level of difficulty and whether it is achievable within the project time frame.

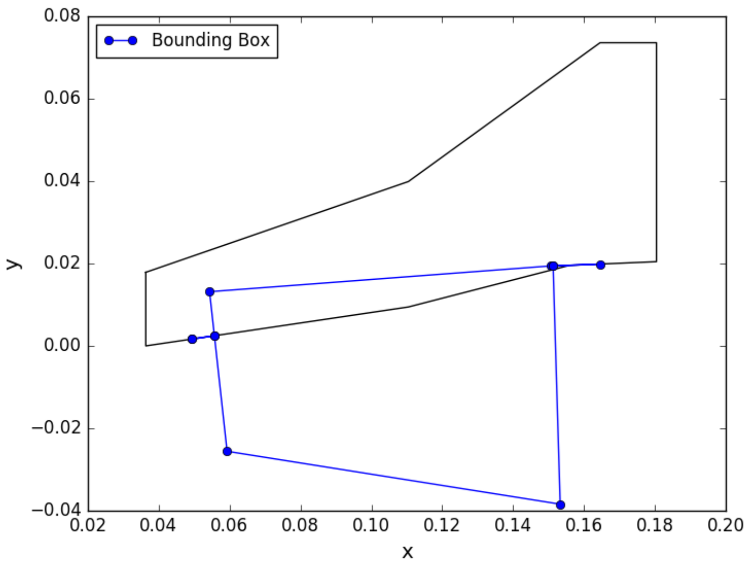


Figure 3-6Bounding box in draft tube case.

Different CFD meshing techniques and models were chosen to run the optimiser with different number of control points: blockmesh [24] was run on one and two control points, cfMesh [25] was run on three control points and pointwise [26] was run on four control points. The budget on function evaluation was set to be $(no.of\ dimension \times 11 - 1) + 100$. To facilitate the automation, the Catmull-Clark spline would be passed to CFD in a form of .stl file if cfmesh was used, whereas it would be passed as a .csv list of subdivided control points if blockmesh or pointwise was used.

3.6. Parallel Evaluation of Initial Samples

Despite Bayesian optimisation is a serial algorithm, the evaluation process could be speeded up by parallel evaluation of independent initial samples using the `multiprocessing` module in Python library. A pool of workers was instantiated depending on the number of cores available on the machine, then each worker would pick up an initial sample and evaluate the sample in their own directories until no initial sample was left unevaluated. The results were combined and returned to the optimiser for model fitting once all initial samples were evaluated. Since the machine which the optimiser ran on has 12 cores, the performance of utilising eight cores on PitzDaily case was first studied before the multiprocessor was implemented on the draft tube case with four cores.

4. Presentation of analytical results and final constructed product

4.1. *Branin Function*

4.1.1. Expected Improvement

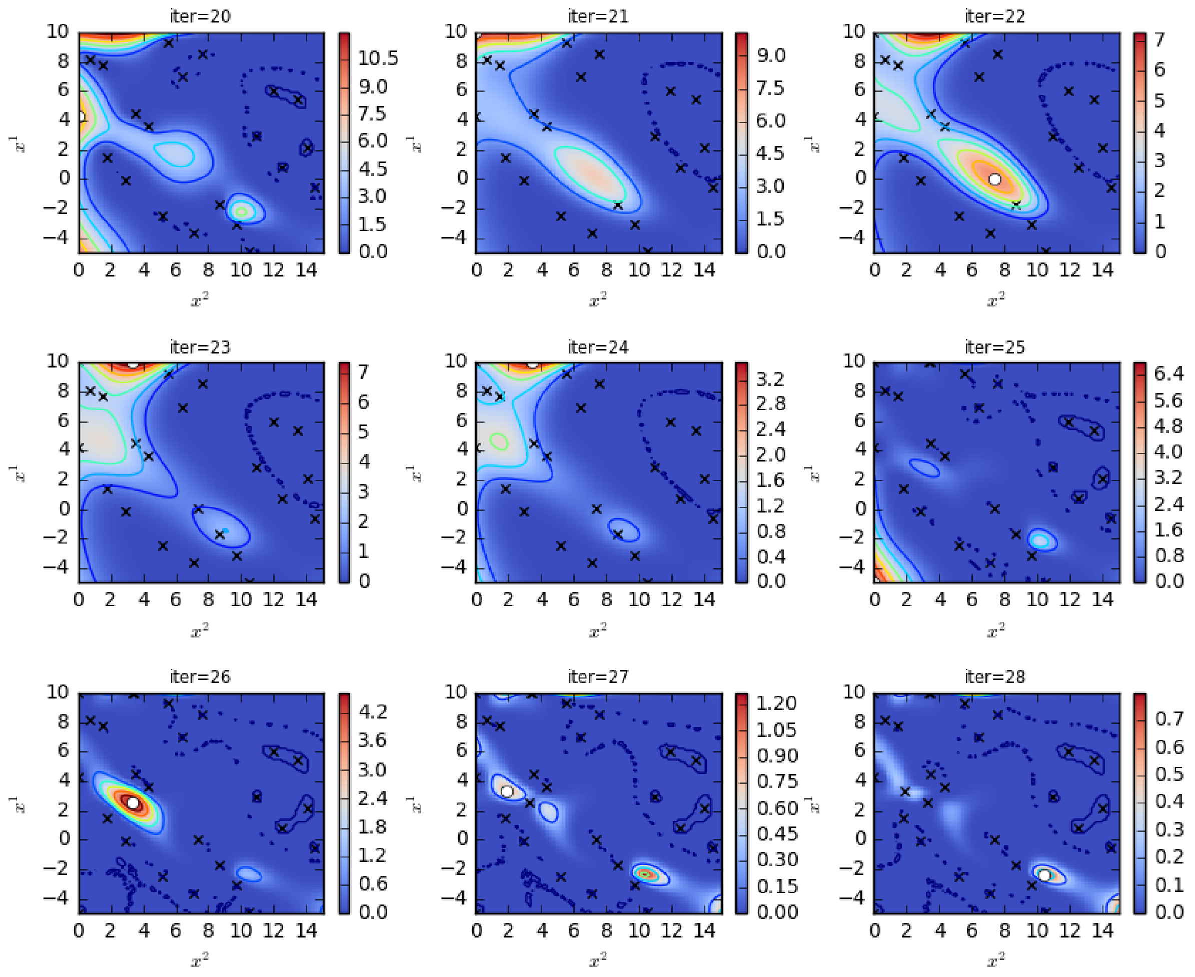


Figure 4-1 Expected Improvement (EI) of the surrogate model over the first nine iterations after evaluation of initial samples.

How the next search point is located and how the EI of the surrogate model changes after each evaluation are demonstrated in Figure 4-1. The black crosses mark all previously evaluated points and the white dot indicates the point chosen to be the point the be searched in the next iteration. From this plot, previously evaluated points (black crosses) were checked to have a very low EI and stochastic global optimiser GA's was also proved to be capable to locate next search point (white dots) in area with highest computed EI. Therefore, this graph indicates both the EI acquisition function and the GA used to optimise the next search point were functioning well.

4.1.2. Optimiser Performance

Improvement over Iteration

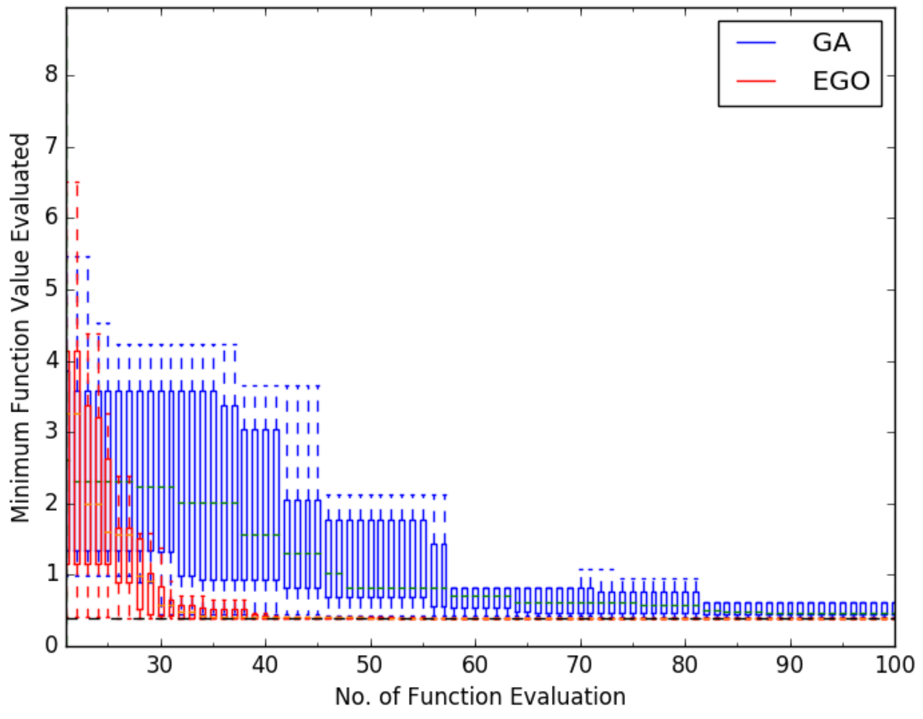


Figure 4-2Performance of GA vs. EGO over Iterations (after initial samples) on Branin Function.

Table 2 Test results of GA and EGO on Branin function over 10 runs.

Optimiser	Minimum evaluated within 100 iterations	#Evaluations to current minimum	Error (%)	Minimum #evaluations required for global minimum +1%
GA	0.40551670556239827	97	1.92	beyond 121
EGO	0.39789668994551697	74	2.44×10^{-3}	26

Figure 4-2 compares the minimum function values minimised by GA and EGO over function evaluations after the initial samples. These error bars are generated from 10 sets of data and the dashed black line marks the global minimum of 0.397887 in Branin function. With the optimisation stage set to after 21 function evaluations, the minimum function value evaluated in EGO went down drastically in contrast to the steady reduction in GA. Together with the important results listed in Table 2, especially from the percentage error of the minimum the two optimisation minimised away from the known global minimum as well as the minimum number of evaluation required for locating a point which is less than 1% difference from the global minimum, it can be said that EGO outperformed GA in this global minimisation benchmark Branin function.

Diagnostic Test on Model Performance

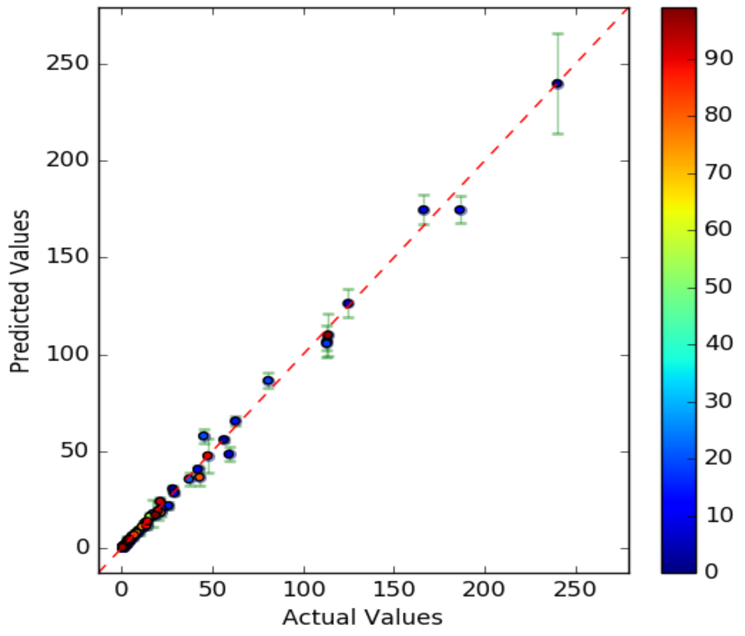


Figure 4-3 Diagnostic test on EGO surrogate model: 5-fold cross-validated prediction value vs. actual function value.

Figure 4-3 shows the diagnostic test for the GPy model fit to the Branin test function, where the 5-fold cross-validated predictions are plotted against the actual function values. This diagnostic test was carried out with 100 function evaluations in EGO to show how well was the approximation made by the surrogate model at the end of the optimisation process. The plot shows the resulting model was good because the points lie closely to the 45° red dashed line. From the optimiser performance over iterations and the diagnostic test performance, the overall performance of this EGO on this simple test function was satisfactory.

4.2. *PitzDaily*

4.2.1. Optimiser Performance

Improvement over Iteration

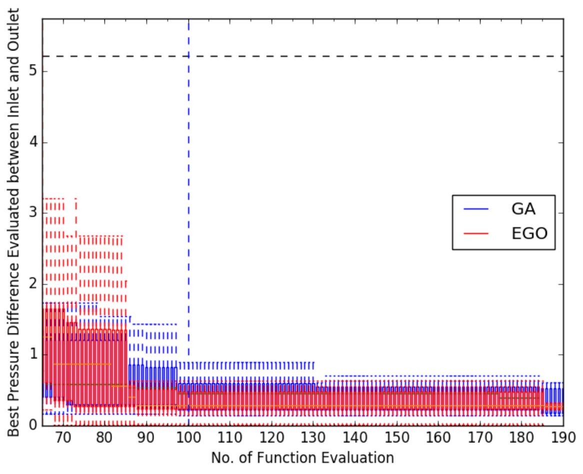


Figure 4-4 Performance of GA vs. EGO on PitzDaily with three control points over 190 iterations. The blue vertical dashed line marks the end of evaluation on GA initial population.

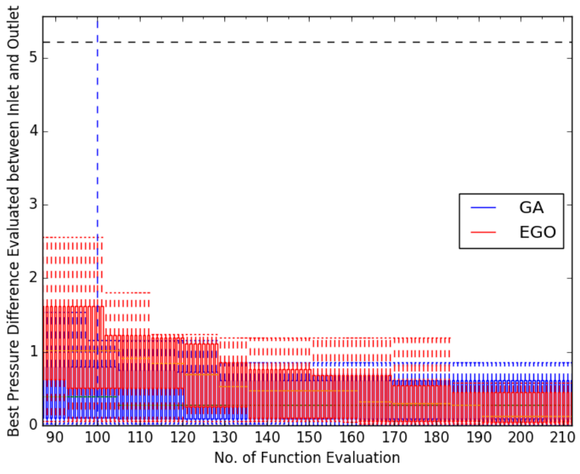
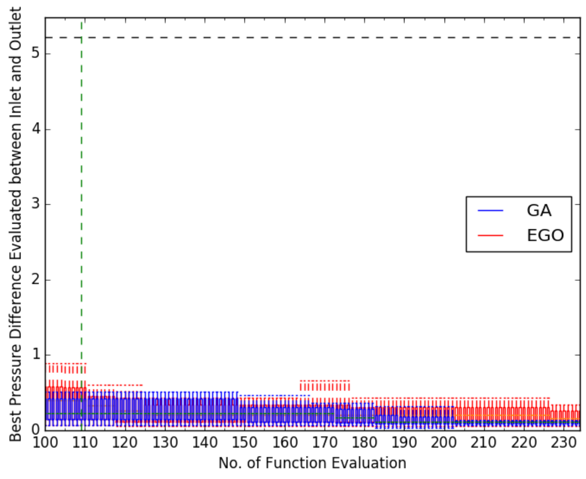


Figure 4-5 Performance of GA vs. EGO on PitzDaily with four control points over 212 iterations. The blue vertical dashed line marks the end of evaluation on GA initial population.



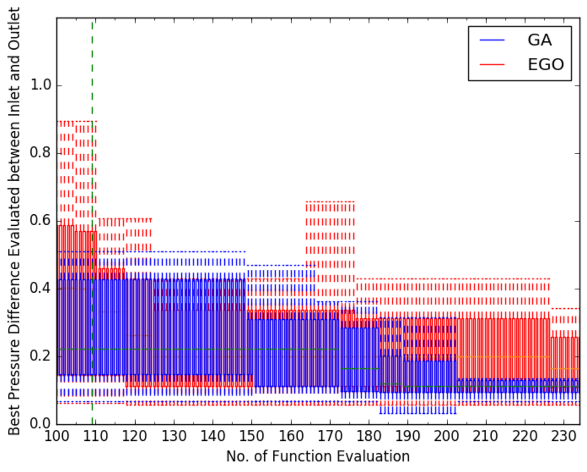


Figure 4-6 Performance of GA vs. EGO on PitzDaily with five control points over 234 iterations: (left) with base pressure difference indicated by black dashed line; (right) enlarged without marking the base pressure difference.; the green vertical dashed line marks the end of EGO's initial evaluation.

Figure 4-4 to Figure 4-6 show the performance of EGO with GA on the PitzDaily case after the initial evaluations. In general, the initial samples generated from uniform random values and using Latin hypercube sampling method both lead to a significant improvement compared to the base pressure at 5.22. Moreover, the randomly selected initial samples in GA generally gave better evaluated function values than the initial samples generated using Latin hypercube sampling method. This might suggest the response surface of the objective function is multimodal so that the evenly spread out random samples were easily located to an area that give improved values.

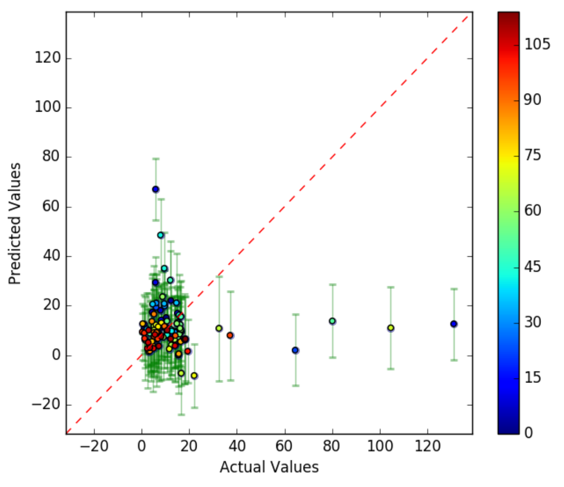
Table 3 Results of GA and EGO on PitzDaily case over 10 runs.

Optimiser	**# Control point**	**Budget (# evaluation)**	**Median of minima evaluated within budget**	**Median #evaluation to budget minimum**	**Median improvement (%)**
GA	3	190	0.27122983001500001	77.5	94.80
	4	212	0.27973230419200001	47.5	94.64
	5	234	0.11173763898200001	161	97.86
EGO	3	190	0.27684091636500002	78.5	94.70
	4	212	0.12673965014835001	131.5	97.58
	5	234	0.1642040301345	96	96.85

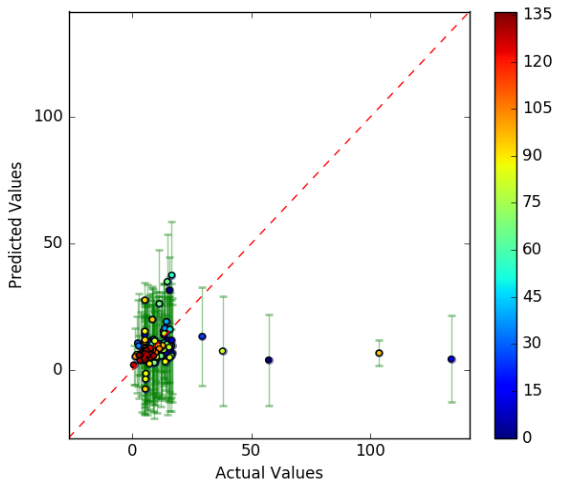
Despite the optimisation stage of GA started off with better improved initial populations, Figure 4-4 and Figure 4-5 share a common trend which EGO outperformed GA by the end of the budget and therefore proved the EGO optimised better than GA. This is particularly noticeable in Figure 4-4 where there is a considerable improvement made by EGO at iteration 186. However, it is unexpected that GA was capable to minimise the objective function noticeably earlier than EGO within budget in Figure 4-6. Nevertheless, the median number of evaluation required to get to the minimum evaluated within the budget in Table 3 reveals the majority of minima GA found within the budget where within its 100 initial samples whereas EGO was able to further improve the initial samples in the optimisation stage. Consider 125 evaluations were carried out following the initial samples regardless of the number of dimension of the optimisation problem, it is conjectured that EGO requires more function evaluations to locate the optimum in the optimisation stage when the problem increases in dimension. The results of diagnostic test on model performance in the next section might give more clue on what's happening in EGO with five control points.

Another observation is new design with larger improvement from base case in terms of a lower pressure difference between the inlet and outlet were generated from more control points. This might due to more control points could be used to specify the details required to optimise the design.

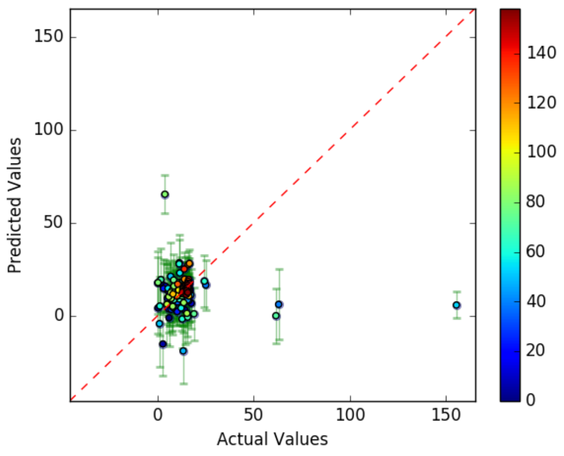
Diagnostic Test on Model Performance



(a)



(b)



(c)

Figure 4-7 Diagnostic Tests for PitzDaily case, 10-fold cross-validated predictions vs. actual function values: (a) three control points with 115 function evaluations; (b) four control points with 137 function evaluations; (c) five control points with 159 function. The colour indicates the evaluation order of each observation point.

The diagnostic tests on PitzDaily with 10-fold cross validation were shown in Figure 4-7. Comparing to the highly satisfactory model prediction performance in Branin function (Figure 4-3), which the predictions lie closely to the 45° red dashed line and indicate accurate predictions, the performance of model prediction in PitzDaily was unsatifactory. In all three surrogate, regardless of the number of control points designated in the problem case and the number of dimension, about 5% of the model predictions were outlying which they were predicted to be small but turned out to be large.

Although the model performance was unsatisfactory, the colour distribution of the observation points pointed out EGO was capable to locate and guide later searches towards the minimal area. The fact that being able to arrange later search within the optimal area in tests with three and four control points helps explain why EGO was capable to optimise better in Figure 4-4 and Figure 4-5. The inaccuracy in model prediction when optimising with five control points is presumably a reason for the poor optimising performance of EGO in Figure 4-6, for a misspecified prior or poorly chosen kernel function would require more function evaluation to produce acceptable posterior estimates [7]. Therefore, it might worth investigate how to tune the model's hyperparameters to fit the response surface better.

4.2.2. Multiprocessor

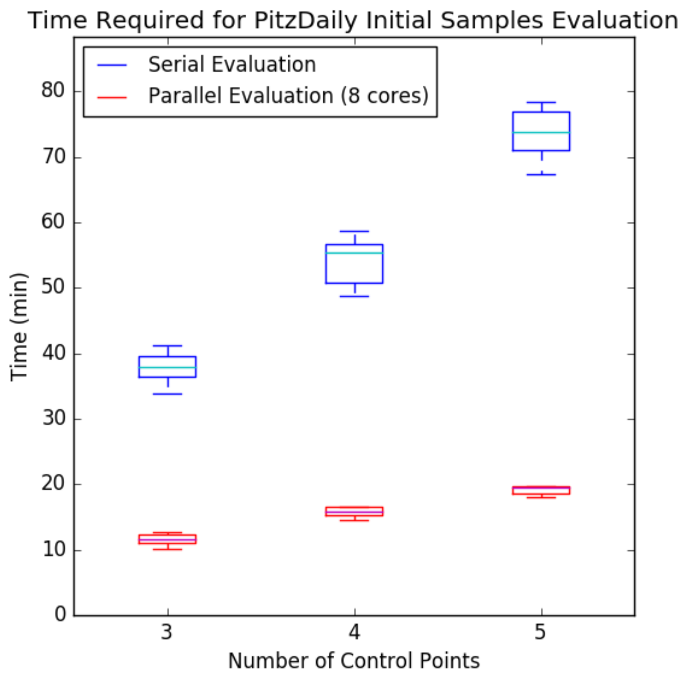


Figure 4-8 Time required to evaluate PitzDaily initial samples serially and parallelly over eight cores, the evaluation time saved by running with three, four and five control points parallely are 69.425%, 70.899% and 74.059% on average respectively.

Figure 4-8 presents error bars of time taken to evaluate initial samples evaluation of PitzDaily serially and parallelly over nine sets of initial sample data. This graph shows how running the initial evaluation parallelly with the multiprocessor can significantly reduce the evaluation time required before moving on to the optimisation stage, and the amount of time saved was more substantial as the number of control points, and hence the number of initial samples required, goes up.

4.3. *Draft Tube*

4.3.1. Boundary Box

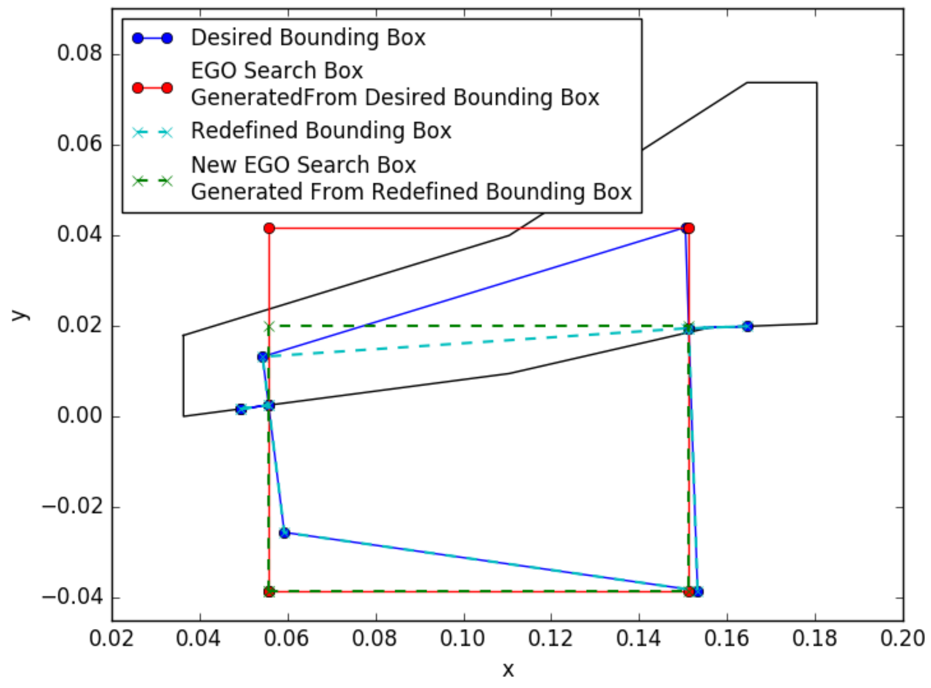


Figure 4-9 Bounding Box and the resulting EGO search box.

The changes in bounding box and its corresponding EGO search box are illustrated in Figure 4-9. Amendment was made to the original blue bounding box suggested by the CFD team because the corresponding search box set up using the method adopted by this EGO optimiser, outlined in red solid line, would potentially generate control points that lie outside the draft tube. This problem was resolved by redefining a smaller bounding box which is outlined in cyan dashed line. However, this implies part of the desired search space and hence, some possible optimum solution, would be sacrificed.

4.3.2. Optimised Shape

Figure 4-10 - Figure 4-13 present the 2D structure of the draft tubes optimised using different CFD meshing techniques and different number of control points. It is interesting to find out the optimised shape from the range of one to four control points both altered the similar area of the draft tube, but the optimised geometry carried more details as the number of control points used increased. Figure 4-14 illustrates the optimised draft tube in Figure 4-13 in 3D structure.

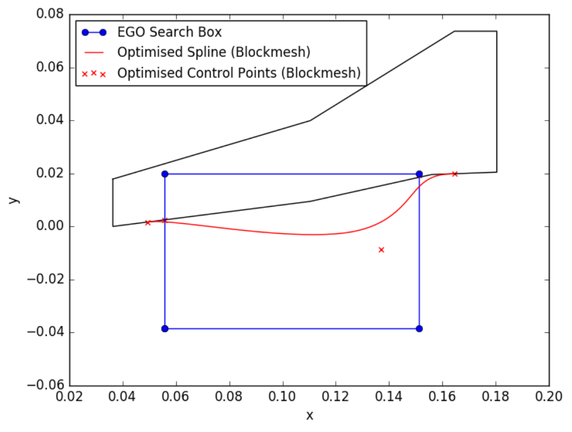


Figure 4-10 Optimised control points and spline (blockmesh with one control point).

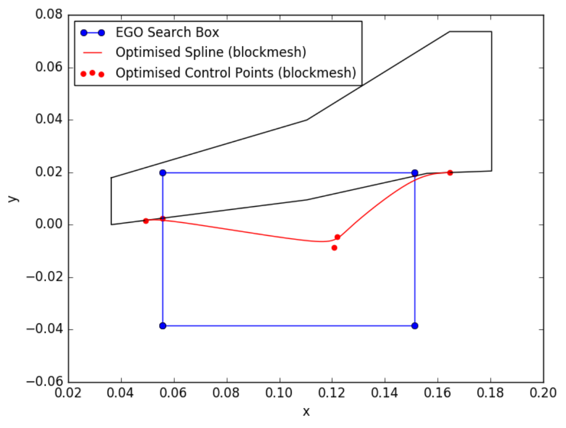


Figure 4-11 Optimised control points and spline (blockmesh with two control points).

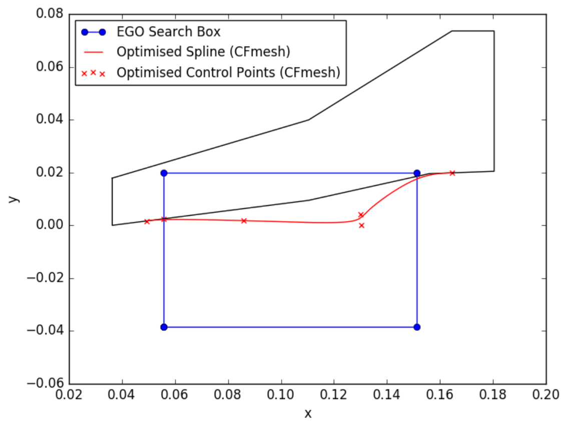


Figure 4-12 Optimised control points and spline (cfMesh with three control points).

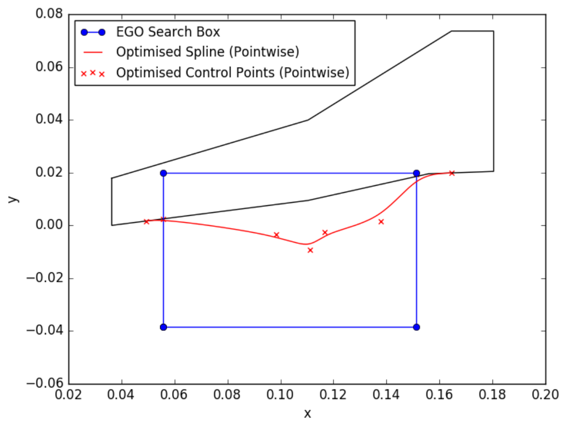


Figure 4-13 Optimised control points and spline (pointwise with four control points).

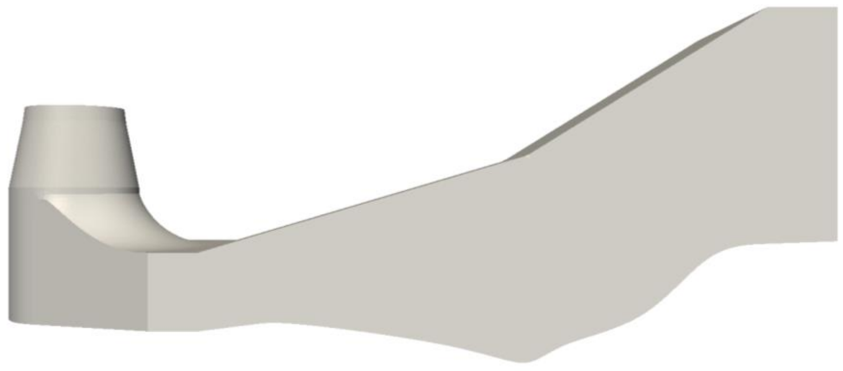


Figure 4-14 Optimised shape of draft tube using pointwise with four control points in ParaView. [26]

4.3.3. Optimiser Performance

In the limited project time scope, the optimisation process on each CFD mesh basically could only be ran once. To study the EGO optimiser performance better and be able to draw conclusive statement, optimisation should be repeated for more runs. In addition, the optimiser is expected to optimise the problem to a higher extend if more function evaluations could run.

Improvement over Iteration

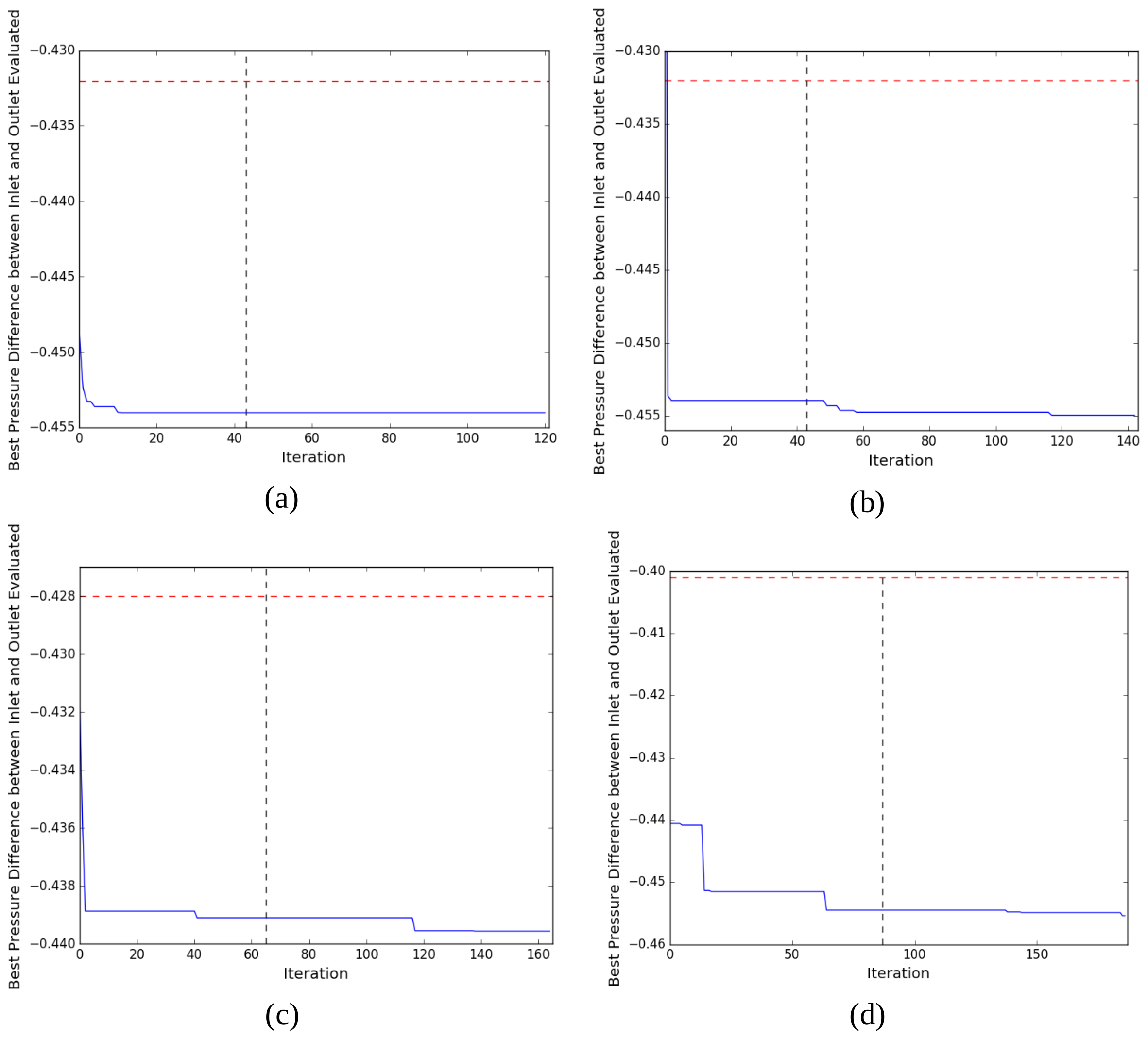


Figure 4-15 Best pressure drop over Iterations, a: blockmesh with one control point; b: blockmesh with two control points; c: cfMesh with three control points; d: pointwise with four control points. The red dashed line marks the base pressure difference of the original draft tube corresponded to the meshing technique; the black dashed line marks the end of initial evaluations.

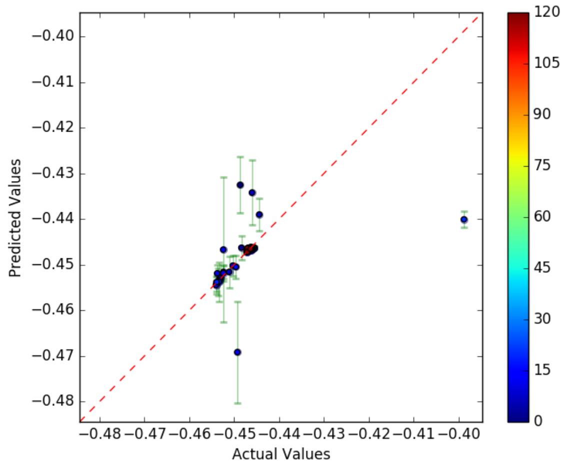
Figure 4-15 proves the automated optimisation of CFD-modelled draft tube with EGO was functioning properly with all three CFD meshing techniques prepared. In all runs, the remarkable improvements were made within the initial samples, and EGO slightly optimised the draft tube design further in the optimisation process, except the one control point draft tube optimisation with blockmesh. The optimisation trends in suggest the draft tubes could presumably be further optimised if more function evaluations could run.

Although each meshing technique was responsible for optimisation problem set up with different number of control points, Table 4 shows the percentage improvement achieved increased with the number of control points in general. The remarkable low improvement from the three-control points optimisation with cfMesh and the outstanding improvement from the four-control points optimisation with pointwise suggest the optimisation result is closely related to the meshing technique used.

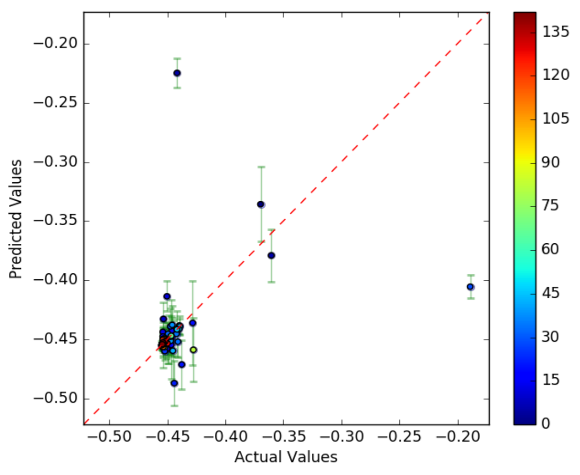
Table 4 Draft tube optimisation results.

Meshing Technique	#Control Points	Base Pressure Difference	Optimised Pressure Difference	#Evaluation to Optimum	Improvement (%)
blockmesh	1	-0.432 [24]	-0.45404299999999997	11	5.10
blockmesh	2	-0.432	-0.45497520000000002	117	5.32
cfMesh	3	-0.4280 [25]	-0.43956129999999999	138	2.70
pointwise	4	-0.401 [26]	-0.45542300000000002	185	13.57

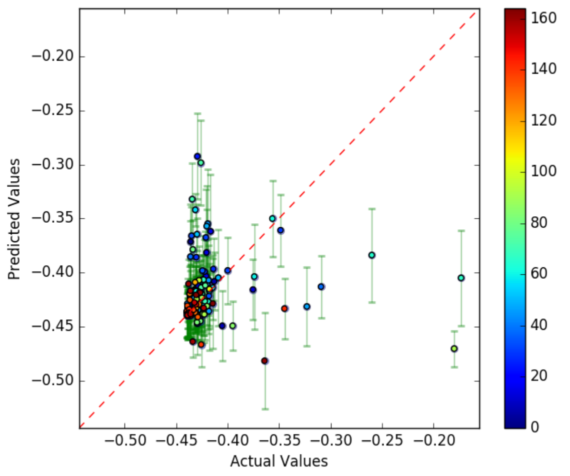
Diagnostic test on model performance



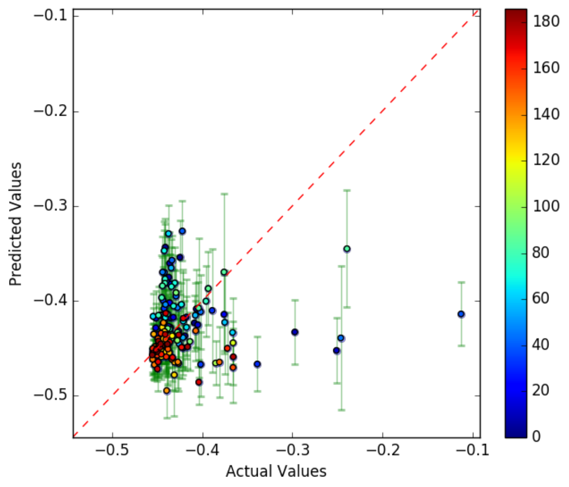
(a)



(b)



(c)



(d)

Figure 4-16 Diagnostic Tests for CFD draft tube case ran with different meshing methods, 10-fold cross-validated predictions vs. actual function values: (a) one control point with 121 function evaluations using blockmesh; (b) two control points with

137 function evaluations using blockmesh; (c) three control points with 165 function evaluations using cfMesh; (d) four control points with 187 function evaluations using pointwise. All points are coloured according to their sampling order.

Although the predicted data points do not lie on the 45° red dashed line which indicates accurate model predictions and there are some outlying data points in all diagnostic tests, the predictions roughly follow the same direction as the 45° line. Despite the models were deficient in covering the whole response surface, it is noteworthy that the model prediction was more accurate at the near-optimal area, which could potentially be sufficient for minimisation purposes. Besides, it is noticed that the model prediction was more accurate and less uncertain at cases with less dimensions by comparing the four plots in Figure 4-16. The model predictions in this CFD-modelled draft tube case were also better in comparison to the model predictions in previous PitzDaily case.

While most of the recent searches were located around the minimal area, the red and orange outlying observation points in the model for the optimisation case of cfMesh with three control points are eye-catching. A potential cause of this unexpected observation is some of the CFD simulation were not converged during the automated optimisation process [25]. Therefore, a modification on cfMesh setting might be required to ensure the function evaluation results are converged. Further study on optimisation with partially converged function would also be helpful in improving the performance of this automated optimiser.

4.4. Final Constructed Optimiser

The program codes of the EGO, GA, the layout which transform geometry design optimisation problem into parameters of the optimisers, and the multiprocessing class are available via the link below:

https://universityofexeteruk-my.sharepoint.com/personal/wlcn201_exeter_ac_uk/_layouts/15/guestaccess.aspx?folderid=01ac47e1996fd49cbb581865eb3526dcb&authkey=AQnWaGT154PdVjb9NI4itkU&expiration=2017-10-31T07%3a58%3a15.000Z

5. Discussion and conclusions

5.1. Discussion and Conclusions

This report studied and compared the performance of EGO built with `GPy` using kernel `Matern52` as the probabilistic surrogate model and EI as the acquisition function on the benchmark Branin function and previous work on automated CFD-based PitzDaily optimisation with GA. It has been shown that EGO is more powerful in global optimisation than GA in both test cases, with EGO exceptionally outperformed GA in the simple Branin function. However, EGO was less efficient than GA within the evaluation budget when optimising PitzDaily with five control points. A potential cause could be kernel function did not match well with the objective function, which would require more function evaluations to give reliable predictions. Although the GP model was not able to accurately model the response surface, EGO is still a functioning optimiser which is capable to direct searches in the optimal area.

The optimised draft tube presented implies the automated CFD-based optimisation with EGO using Catmull-Clark spline parameterisation was a success. This is very encouraging as this project only optimised the diffuser section of draft tube, the improvement is presumably more significant if the entire draft tube could be optimised.

5.2. Future Work

Future work on this project could have two directions. In terms of the overall performance of EGO, further research could be done on the implementation of EGO with expensive constraints. Also, to further extend the parallelism of this optimiser as well as to make EGO a more efficient and powerful optimisation tool, it worth to research more on the topic of parallel Bayesian optimisation.

Apart from improving the overall performance of EGO, geometry representation of solutions could be another area to work on which aids the implementation of automated CFD-based optimisation. A better search box generation mechanism could be adopted which the EGO search such that the search box could have arbitrary shape to expand and utilise the search space. This could potentially be achieved by setting the control points that fall outside the search boundary to have zero EI value as a penalty of violating the constraints.

6. Project management

Within the project period, regular group meeting was held weekly with project supervisor. Each group member took turn to be the chair person and the secretary in the meeting. The meeting minutes were stored in a shared group folder on Google drive.

The project was split into four main stages: preliminary work; EGO implementation; EGO with PitzDaily; and EGO with CFD-based draft tube. The preliminary work was fundamentally research on Bayesian optimisation and EGO as well as GP, which was taken place in the first three weeks of the project. Stage two, EGO implementation, was carried out almost parallelly with stage one but it eventually took around six weeks. Stage three, adaptation of EGO to PitzDaily case, lasted approximately two and a half weeks after EGO was tested. The final stage, the construction and implementation of automated EGO on CFD-modelled draft tube, was carried out since March until the end of project.

7. Contribution to group functioning

This project worked closely with Burns, Gowans and Hardy from the CFD sub-team. The CFD sub-team was responsible for development of CFD meshing techniques which would remain robust in automated design process and the construction of accurate CFD draft tube model to be optimised. blockmesh, pointwise and cfMesh as well as the corresponding draft tube model developed and prepared by Burns, Gowans and Hardy respectively were used with the optimiser in this project. In addition, this project worked collaboratively with the CFD team to build an automated interface between the optimiser and CFD parts. While the CFD team worked on automatic meshing, set simulation running and return the simulation results, this piece of work was to program a functioning EGO optimiser which generates a .stl file or a list of control points of the altered bottom spline in .csv file as required by each CFD meshing technique. Altogether, our works worked towards a successful interface for automated CFD-based optimisation and its implementation optimised the draft tube with appreciable improvement.

It was expected that Gilbert's work on surface sensitivity and gradient information could be implemented as an adjoint function to this Bayesian optimiser, and Hutching's work on three-dimensional representation of the draft tube could contribute to optimisation of the entire draft tube geometry instead of limiting to the diffuser.

8. References

[1] S. Daniels, A. Rahat, G. Tabor, J. Fieldsend and R. Everson, “A review of shape distortion methods available in the OpenFOAM framework for automated design optimisation,” Exeter.

[2] J. Snoek, H. Larochelle and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *Advances in Neural Information Processing Systems,* vol. 25, pp. 2960-2968, 2012.

[3] J. S. Bergstra, R. Bardenet, Y. Bengio and B. Kégl, “Algorithms for Hyper-Parameter Optimization,” *Advances in Neural Information Processing Systems 24,* pp. 2546-2554 , 2011.

[4] M. Hoffman, B. Shahriari and N. Freitas, “On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning,” in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, Reykjavik, 2014.

[5] X. Fern, J. Azimi and A. Jalali, “Hybrid Batch Bayesian Optimization,” in *Proceedings of the 29 th International Conference on Machine Learning*, Edinburgh, 2012.

[6] M. A. Osborne, R. Garnett, and S. J. Roberts, “Gaussian Processes for Global Optimization,” in *3rd International Conference on Learning and Intelligent Optimization (LION3),* 2009.

[7] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. de Freitas, “Taking the Human Out of the Loop: A Review of Bayesian Optimization,” *Proceedings of the IEEE,* vol. 104, no. 1, pp. 148 - 175, January 2016.

[8] D. R. Jones, M. Schonlau and W. J. Welch, “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization,* vol. 13, no. 4, pp. 455-492, 1998.

[9] W. R. Thompson, “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples,” *Biometrika,* vol. 25, no. 3/4, pp. 285-294, 1933.

[10] J. A. Nelder and R. W. M. Wedderburn, “Generalized Linear Models,” *Journal of the Royal Statistical Society. Series A (General),* vol. 135, no. 3, pp. 370-384, 1972.

[11] J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn, “Design and Analysis of Computer Experiments,” *Statistical Science,* vol. 4, no. 4, pp. 409-435, 1989.

[12] E. Brochu, V. M. C. Cora and N. de Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,” 14 12 2010. [Online]. Available: https://arxiv.org/abs/1012.2599v1. [Accessed 11 04 2017].

[13] F. Hutter, H. H. Hoos and K. Leyton-Brown, “Parallel Algorithm Configuration,” *Learning and Intelligent Optimization,* vol. 7219, pp. 55-70, 2012.

[14] M. Ebden, “Gaussian Processes: A Quick Introduction,” 29 August 2015. [Online]. Available: https://arxiv.org/abs/1505.02965. [Accessed 17 February 2017].

[15] “Gaussian processes framework in python,” [Online]. Available: https://github.com/SheffieldML/GPy. [Accessed 1 February 2017].

[16] N. Durrande , D. Ginsbourger and O. Roustant , “Additive Kernels for High-dimensional Gaussian Process Modeling,” Technical Report, 2010.

[17] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design,* vol. 10, no. 6, pp. 350-355, 1978.

[18] D. James, “Subdivision Overview,” 2009. [Online]. Available: http://www.cs.cornell.edu/courses/cs4620/2009fa/lectures/01subdivision.pdf. [Accessed 24 April 2017].

[19] D. Fussell , “Subdivision curves,” 2010. [Online]. Available: http://www.cs.utexas.edu/users/fussell/courses/cs384g-fall2011/lectures/lecture17-Subdivision_curves.pdf. [Accessed 24 April 2017].

[20] B. Hadorn, “Genetic Algorithm,” 13 September 2016. [Online]. Available: http://www.xatlantis.ch/index.php/education/zeus-framework/49-genetic-algorithm#line_recombination. [Accessed 3 May 2017].

[21] Y.-H. K. Kim and Y. Yoon, “The Roles of Crossover and Mutation in,” in *Bio-Inspired Computational Algorithms and Their Applications,* InTech, 2012, pp. 65-82.

[22] D. Ortiz-Boyer, C. Herv´as-Mart´ınez and N. Garc´ıa-Pedrajas , “CIXL2: A Crossover Operator for Evolutionary Algorithms,” *Journal of Artificial Intelligence Research,* vol. 24, pp. 1-48, 2005.

[23] “scipy.stats.truncnorm — SciPy v0.18.1 Reference Guide,” [Online]. Available: https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.stats.truncnorm.html. [Accessed 11 April 2017].

[24] P. Burns, “The development of an automatic numerical draft tube model,” Exeter, 2017.

[25] S. Hardy, “Automatic Mesh Generation for Shape Optimisation of a Draft Tube Using cfMesh,” Exeter, 2017.

[26] J. Gowans, “Automatic structured and unstructured mesh generation using Pointwise,” Exeter, 2017.