國立臺灣大學工學院工業工程學研究所

碩士論文

Institute of Industrial Engineering
College of Engineering
National Taiwan University

Master Thesis

# 以高斯過程及啟發演算法的離線最佳化模擬系統參數
## —海底渦輪機應用

# Simulation parameters optimization based on Gaussian process and metaheuristics - An application of marine turbine

秦柔
Jou Chin

指導教授：洪一薰博士
Advisor: I-Hsuan Hong, Ph.D.

中華民國 108 年 7 月
July, 2019

# 國立臺灣大學碩士學位論文
# 口試委員會審定書

## 以高斯過程及啟發演算法的離線最佳化模擬系統參數—海底渦輪機應用

## Simulation parameters optimization based on Gaussian process and metaheuristics - An application of marine turbine

本論文係秦柔君 (R06546004) 在國立臺灣大學工業工程學研究所完成之碩士學位論文，於民國 108 年 7 月 28 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

_____

_____ _____

_____ _____

_____ _____

所　長：_____

# 摘要

此部分要寫中文摘要

關鍵字： 關鍵字

# Abstract

System models that enable design space exploration are built from different information sources such as numerical simulations, physical experiments, analytical solutions and heuristics. These models, called surrogates, are nonlinear and adaptive in nature and thus suitable for system responses where limited information is available and few realizations (experiments or numerical simulations) are feasible. In this paper, the surrogate framework is applied to estimate values for unknown physical parameters of a marine turbine. For this purpose, physical experiments and numerical simulations are performed on a computer simulator. Numerical models are studied which involve four unknown parameters. Through the use of genetic algorithm and particle swarm optimization, an efficient exploration of the parameter space is performed. The objective is to determine system parameters to improve the accuracy of computer simulator. Surrogate models are built to combine information obtained from numerical simulations and experimental model measurements. The integration of several surrogate models reduces the number of large-scale numerical simulations needed to find reliable estimates of the system parameters and modify the simulation to be more accurate to the physical experiments.

**Keywords:** numerical model, surrogate model, genetic algorithm, particle swarm optimization

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Numerical simulations provide a wealth of physical insight into the system behavior and fill the gap between theory and experiment. Heermann(1990) stated methods of numerical simulation and declared that some quantities or behaviors may be impossible or difficult to measure in an experiment. With numerical simulations such quantities can be computed, but from the design perspective, they provide only pointwise information (i.e., information for single combinations of the design parameters). On the other hand, performing numerical simulations for large set of design alternatives would require considerable resources. The challenge is then to gain an understanding of the system behavior through a small number of numerical simulations. The paradigm shift that has been proposed to meet this challenge is to treat numerical simulations as computer experiments and build a meta-model (model built from another model)—called surrogate model which, in spirit, is similar to a response surface or interpolation model of the numerical simulation.

Surrogate methodologies have been proposed as an alternative to a direct function evaluation of a computationally expensive numerical simulation within an iterative search process (e.g., design optimization). The surrogate approach consists of building an interpolating model that relates the response to the parameters. Then, the computationally inexpensive surrogate substitutes the numerical simulation within the iterative process. Surrogates are meta-models built with information from numerical, experimental or other sources of data. The purpose is to replace the numerical simulation with the surrogate during optimization, robustness studies or other design related decision process and it is com-

mon to use surrogate model to solve the engineering problems. Anand et al.(2011) published paper about developing surrogate model for fuels for advanced combustion engines, Braconnier et al.(2011) provided a dynamic building of surrogate models for aerodynamic flight data generation, and Yamazaki et al.(2010) applied the gradient/Hessian-enhanced surrogate models are shown to develop aerodynamic database construction.

In summary, the surrogate approach consists of building a model based on information of a discrete set of data, with the underlying hypothesis that obtaining this information requires a considerable amount of resources (e.g., time, computation, etc.). An advantage of the surrogate modeling methodology we propose is that, through the use of different sources of information, we gain as a feedback an improved physical insight and a potentially reusable knowledge instead of just a numerical black box Mehdad et al.(2018).

In this paper, we present an improvement of this paradigm by building surrogate models that incorporate information from large-scale numerical simulations, which can be synthesized at a lower computational cost than the numerical simulation itself and the simulation data is originated from the simulator' s prediction of a marine turbine James et al.(2017). Our surrogate framework is in Gaussian process regression (GPR) which is introduced by Rasmussen(2003).This GPR model is built upon work on analysis of computer experiments.

The initial objective of this paper is to find the parameter values that minimize a chosen error measure between the physical experiment and the predictions from each numerical model considered. The surrogate model is built for the error measure from a few numerical simulations of different parameter combinations and, then, we use the surrogate model and metaheuristics to find the minimizing parameters. The surrogate models uses the data from the numerical and the physical experimental measurement. The purpose is also to illustrate the possibility of reducing the resources required (i.e., computational effort) to build the surrogate model by using all the available information about the system.

The problem investigated in this paper is highly complex and nonlinear. Traditional deterministic methods such as linear programming and non-linear programming might lead to local optima and thus become unsuitable to solve such complex problems. A review

of the literature reveals that metaheuristics, such as the Genetic Algorithm (GA) for solving multi-objective optimization Rey et al.(1994), simulated annealing (SA) for job shop scheduling Van et al.(1992), tabu search (TS) for optimal power flow Abido(2002), random walk for calculating the density of states Wang et al.(2001), gradient decent method for nonsmooth separable minimization Tseng et al.(2009), and Particle Swarm Optimization (PSO) for handling multiple objectives Coello et al.(2004).In our paper, we have successfully applied GA and PSO on marine turbine problems.

The Genetic Algorithms (GAs) are proposed based on Darwin's principle of survival of the fittest by Holland(1992) to solve larger scale combination optimization problem. It can jump out local search space to achieve optimal solutions in global space. In Genetic Algorithms, it process a population of individuals which represent search space solutions, each individual is candidate solution and population including all individuals are examined simultaneously, and quality of population are improved gradually, at last the best solution or secondary solutions are achieved by repeating employing three GA operations: selection, crossover and mutation. GA are theoretically and empirically proven to provide robust search capabilities in complex spaces, offering a valid approach to problems requiring efficient and effective search.

PSO was originally developed by Kennedy (2010), and was inspired by the social behavior of the flock of birds. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles are considered to be "flying" through the problem space searching for optimal solution. A particle's location in the multidimensional problem space represents one solution for the problem. When a particle moves to a new location, a different solution to the problem is generated.

In this paper, our main goal is to find the best parameters of the simulator in order to predict the result of the physical experiment precisely. Instead of doing the costly physical experiment from time to time, we use simulator to estimate the turbulence intensity (I) and velocity along the centerline of the wake (U) each at two influent $3\%$ and $15\%$ which are written in $I_{3\%}$, $I_{15\%}$,$U_{3\%}$, and $U_{15\%}$, respectively and these quantities of interest (QoI) are all under ten different distance. Since running the simulator by trial and error is time-

consuming, we use the Gaussian process regression as surrogate model to predict the value of QoI by learning the historical result of 81 datasets in simulator, then, apply GA and PSO to estimate the best parameters of the simulator which can make all QoI in the smallest gap compare with the physical experiment. Finally, we will display the result of two set of best parameters in comparison with the physical experiment, GPR model and simulation result in each of GA and PSO, respectively.

# Chapter 2

# Framework Development

To incorporate the surrogate model and metaheuristics into the problem of determining a set of unknown parameters in simulator, we should develop a standard in mathematical methodology to judge QoIs of simulator with those parameters tuned by PSO and GA. Thus, we propose the following general procedure. The following steps indicate the particular decisions that we have to make:

1. $\beta_p$, $\beta_d$, $C_{\epsilon 4}$, $C_{\epsilon 5}$ are the four parameters to be determined in the simulator. These parameters can come to the result of $y^S_{i,j,d}$. $y^T_{i,j,d}$ is the physical experiment estimated by turbine marine. On the other hand, the surrogate model prediction noted as $\hat{y}_{i,j,d}$. All the variables are written in Table 2.1

2. Appropriate data preprocessing can make surrogate model predict accurately. In this paper, we use clustering, outlier detection, bootstrap to clean the data.

3. There are a large number of surrogate models for solving regression problem. We can find the suitable model by testing all of the algorithms in scikit-learn cheat sheet.

4. After we select the appropriate surrogate model, it is necessary to define a quantitative measure, evaluating the error between the surrogate model predictions and numerical simulation. The mean absolute percentage error MAPE is chosen as a metric to compare surrogate model predictions with numerical simulation. In our paper, surrogate model fitting error $MAPE_{srgt/sim}$ is defined as

$$MAPE_{srgt/sim} = \frac{1}{N_{datasets}} \frac{1}{N_{points}} \sum_{m=1}^{N_{datasets}} \sum_{i} \sum_{d} \sum_{j} \left( \frac{\left| \hat{y}_{i,j,d} - y^{S}_{i,j,d} \right|}{y^{S}_{i,j,d}} \right) \tag{2.1}$$

where i noted as initial conditions 3% or 15% under two state variables j= intensity, velocity with ten spatial locations d=1.2,2,3,4,5,6,7,8,9,10. Our goal is to make the surrogate model fitting error $MAPE_{srgt/sim}$ within 10%; otherwise, the surrogate model should be rebuilt.

Table 2.1: The definition of variables

| Name | Expression |
|---|---|
| $y^{T}_{i,j,d}$ i =3%,15% j=U,I d=1.2,2,3,4,5,6,7,8,9,10 | $y^{T}_{i,j,d}$ is the state variables j of physical experiment under the initial condition i at spatial locations d. |
| $y^{S}_{i,j,d}$ i =3%,15% j=U,I d=1.2,2,3,4,5,6,7,8,9,10 | $y^{S}_{i,j,d}$ is the state variables j of simulator prediction under the initial condition i at spatial locations d. |
| $\hat{y}_{i,j,d}$ i =3%,15% j=U,I d=1.2,2,3,4,5,6,7,8,9,10 | $\hat{y}_{i,j,d}$ is the state variables j of simulator prediction under the initial condition i at spatial locations d. |
| $\beta_p$ | The first parameter of the simulator. |
| $\beta_d$ | The second parameter of the simulator. |
| $C_{\epsilon 4}$ | The third parameter of the simulator. |
| $C_{\epsilon 5}$ | The forth parameter of the simulator. |

5. After building the suitable surrogate models, we should use metaheuristics to find the best parameters which can get lowest predicted error. In this paper, predicted error $MAPE_{srgt/exp}$ describes the difference between surrogate model prediction and physical experiment and is written in

$$MAPE_{srgt/exp} = \frac{1}{40} \sum_{i} \sum_{d} \sum_{j} \left( \frac{\left| \hat{y}_{i,j,d} - y^{T}_{i,j,d} \right|}{y^{T}_{i,j,d}} \right) \tag{2.2}$$

6. The parameters predicted by metaheuristics will be set up in the simulator. After the simulation, we get 40 points of simulation result which will be verified with the surrogate model predictions. This optimal solution fitting error are also written in $MAPE_{srgt/sim}$ and is restricted to below $10\%$.If not, these 40 points should add as optimal solution points to the samples and we should go to step two again. This optimal solution fitting error $MAPE_{srgt/sim}$ is formulated as

$$MAPE_{srgt/sim} = \frac{1}{40} \sum_{i} \sum_{d} \sum_{j} \left( \frac{\left| \hat{y}_{i,j,d} - y^{S}_{i,j,d} \right|}{y^{S}_{i,j,d}} \right) \tag{2.3}$$

7. The difference between 40 points of simulation result and the physical experiment is so called Final error $MAPE_{sim/exp}$.In our case, final error must be within $23\%$ because the smallest $MAPE_{sim/exp}$ of historical simulation data is $23\%$. We should find other parameters which is better than the historical parameters; otherwise, optimal solution points to the samples will be added to the samples and the surrogate model should be rebuilt. The final error $MAPE_{sim/exp}$ is the form of

$$MAPE_{sim/exp} = \frac{1}{40} \sum_{i} \sum_{d} \sum_{j} \left( \frac{\left| y^{S}_{i,j,d} - y^{T}_{i,j,d} \right|}{y^{T}_{i,j,d}} \right) \tag{2.4}$$

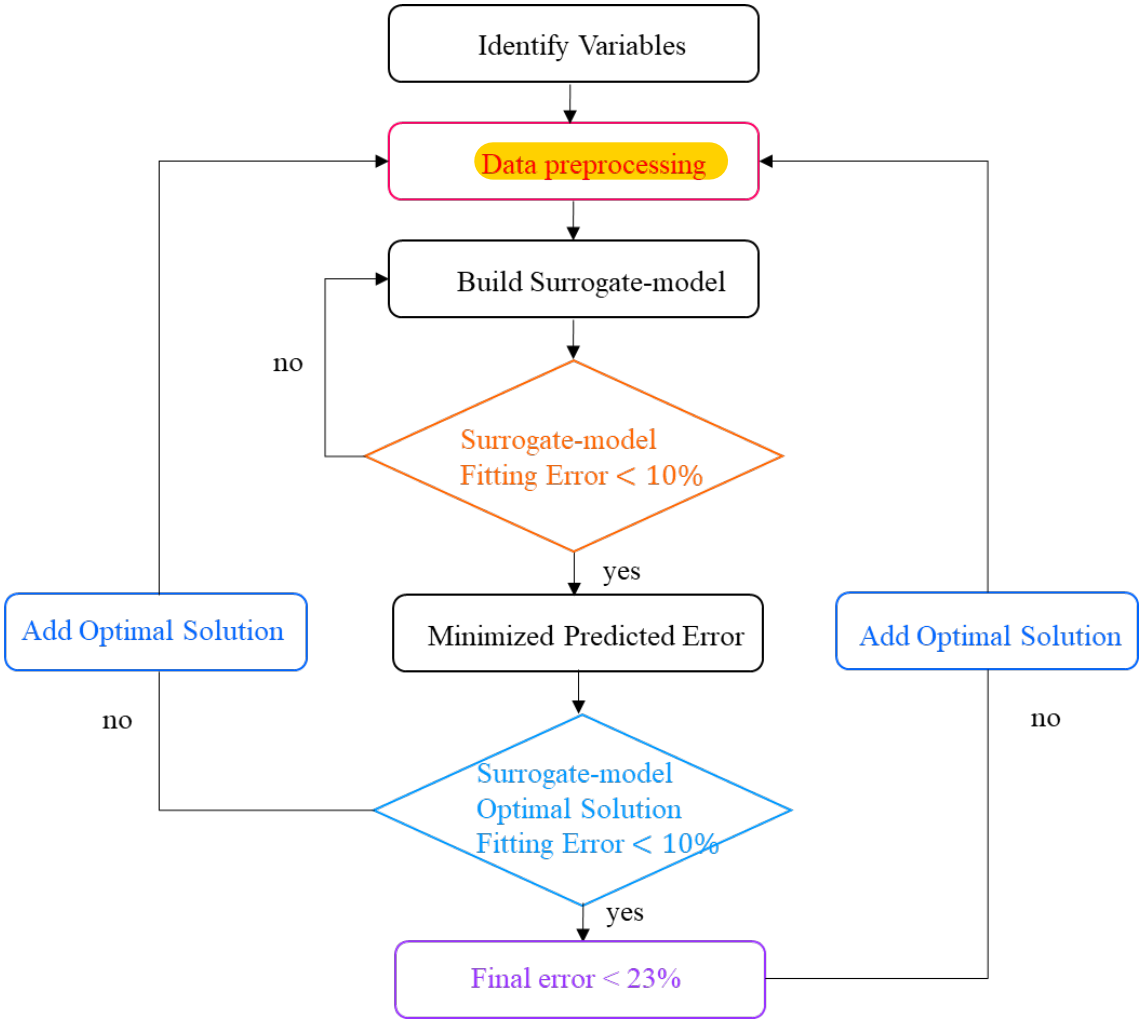



Figure 2.1: Flowchart for the surrogate-model framework

# Chapter 3

# Surrogate Model and Metaheuristics Development

In this chapter, we will introduce one type of surrogate model named Gaussian process and provide two metaheuristics to find the best parameters in order to minimize our predicted error.

## 3.1 Gaussian Process

A Gaussian process is a collection of random variables, any finite set of which have a joint Gaussian distribution. A Gaussian process is completely specified by its mean function $m(x)$ and the covariance function $k(x, x')$:

$$f(x) \sim GP(m(x), k(x, x')). \tag{3.1}$$

There is a training set $D$ of n observations, $D = \{(x_i, y_i) \mid i = 1, \ldots n\}$ where x denotes an input vector, y denotes a scalar output or target. The column vector inputs for all n cases are aggregated in the $D{\times}n$ design matrix X and the targets are collected in the vector y.

The goal of Bayesian forecasting is to compute the distribution $p(y_*|x_*, D)$ of output $y_*$ given a test input $x_*$ and a set of training points $D$. Using Bayesian rule, the posterior

distribution for the Gaussian process outputs $y_*$ can be obtained. By conditioning on the observed targets in the training set, the predictive distribution is Gaussian:

$$y_*|x_*, X, y \sim N(\hat{y}(x_*), \hat{\sigma}(x_*)). \tag{3.2}$$

where, the mean and variance are given by

$$\hat{y}(x_*) = k_*^T(K + \sigma_n^2 I)^{-1} y. \tag{3.3}$$

$$\hat{\sigma}(x_*) = k(x_*, x_*) - k_*^T(K + \sigma_n^2 I)^{-1} k_*^T. \tag{3.4}$$

where, a compact form of the notation setting for matrix of the covariance functions are: $k_* = K(X, x_*)$, $K = K(X, X)$, $\sigma_n^2$ is the unknown variance of the Gaussian noise. Gaussian process procedure can handle interesting models by simply using a covariance function with an exponential term:

$$k_y(x_p, x_q) = \sigma_f^2 \exp(-\frac{1}{2l^2}(x_p - x_q)^2) + \sigma_n^2 \delta_{pq}. \tag{3.5}$$

In addition to an exponential covariance function, there are five others in Table 3.1

Table 3.1: The definition of covariance functions

| Covariance function | Expression |
|---|---|
| ExpSineSquared | $exp(-2(\frac{sin(\frac{\pi}{p}d(x_i,x_j))}{l})^2))$ |
| DotProduct | $\sigma_0^2 + x_i \cdot x_j$ |
| Radial-basis function(RBF) | $exp(-\frac{1}{2}d(\frac{x_i}{l}, \frac{x_j}{l})^2)$ |
| Rational quadratic(RQ) | $(1 + \frac{d(x_i,x_j)^2}{2\alpha l^2})^{-\alpha}$ |
| Matern | $\frac{1}{2^{\upsilon-1}\Gamma(\upsilon)} \left(\frac{\sqrt{2\upsilon}}{l} r\right)^{\upsilon} K_{\upsilon} \left(\frac{\sqrt{2\upsilon}}{l} r\right)$ |

Equation 3.5 expresses the idea the cases with nearby inputs will have highly correlated outputs. The covariance is denoted $k_y$ as it is for the noisy targets $y$ rather than for the underlying function $f$ GP employs a set of hyperparameters $\theta$ including the length-scale $l$, the signal variance $\sigma_f^2$and the noise variance $\sigma_n^2$, $\theta$ can be optimized based on log-

likelihood framework:

$$\log p(y|X,\theta) = -\frac{1}{2}y^T(K+\sigma_n^2 I)^{-1}y - \frac{1}{2}\log|K+\sigma_n^2 I| - \frac{n}{2}\log 2\pi. \qquad (3.6)$$

Hyperparameters $\theta$ are initialized to random values (in a reasonable range) and then use an iterative method, for example conjugate gradient, to search for the optimal values.

## 3.2 Genetic Algorithm

The Genetic Algorithm(GA) is a metaheuristic that is guided for random searches so as to find an optimal solution in large spatial domain. Genetic Algorithm is a relatively new concept which is based on principles and fundamentals of natural assortment of nature. The Genetic Algorithm was first presented by John Holland. The algorithm is unaccepted and evolutionary technique which is also a global search method and main advantage of Genetic Algorithm is that it has a faster convergence. The Genetic Algorithm is considered to be utmost prevailing and fast optimization technique for a large solution spatial domain. Genetic Algorithm is used in different areas of study such as Artificial Intelligence, Facial Recognition, Computer Graphics and Signal Transmission.

The optimization problem solution is produced by employing natural selection of the acceptable entities. The optimization process is performed by natural conversation of genetic material amid the offspring's and parents are formed from those adapted genes from parents. The fitness of offspring's is calculated and only the fittest entities are approved to breed and rise. So, this paper will explain how this natural selection process is adapted by Genetic Algorithm and selection of best optimized solution.

Genetic Algorithm (GA) is a meta-heuristic search algorithm based on the evolutionary. The main idea is derived the behavior of reproduction animal, consist of selection, crossover and mutation. Genetic Algorithms represent an intelligent exploitation of a random search. It has been applied to solve optimization problem for many years. The original Genetic Algorithm is consists of the following component:

1. generate an initialize of the populations P($t$) randomly.

2. the fitness function/objective function of the populations are evaluated.

3. the next generation is produced by selecting two current populations, P1 and P2 (roulette wheel).Apply one-point crossover operator to P1 and P2 with crossover rate (Pc) to obtain a child chromosome C1 and C2.

4.apply mutation operator to C1 and C2 with mutation rate Pm($t$) to produce D($t$) and evaluate D($t$).

5.select P($t+1$) and the new generations D($t$) by their fitness ranking.

6. if we reach the maximum stopping criteria then stop;otherwise, return to Step 1.

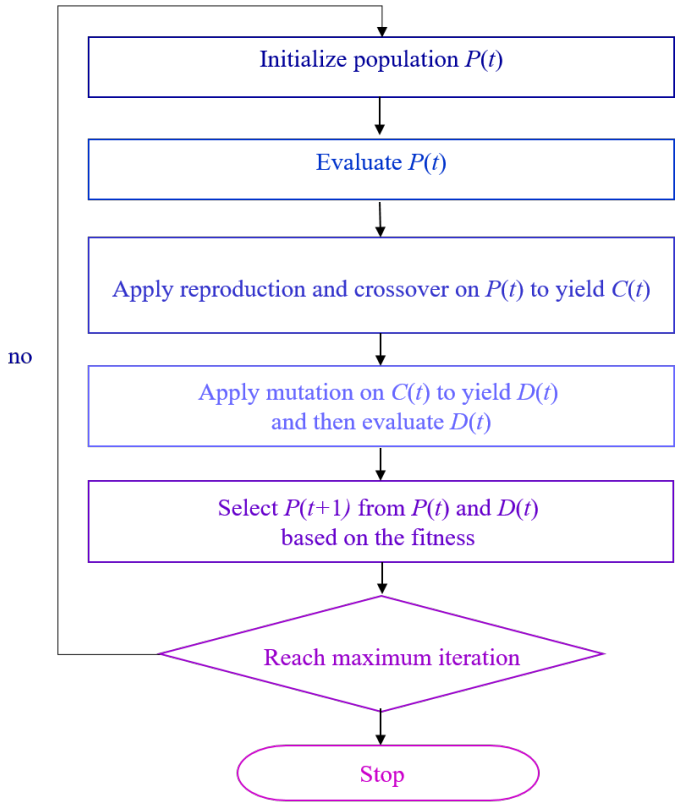



Figure 3.1: Flowchart for implementation of genetic algorithm

## 3.3 Particle Swarm Optimization Algorithm

PSO is initialized with a population of random solutions and searches for optima by updating generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution it has achieved so

far. This value is called $p_{best}$. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. When a particle takes all the population as its topological neighbors, the best value is a global best and is called $g_{best}$. After finding the two best values, the particle updates its velocity and positions with following equation:

$$\left.\begin{aligned} v_{id} &= w v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \\ x_{id} &= x_{id} + v_{id} \end{aligned}\right\} \tag{3.7}$$

$v_{id}$ is velocity of the $i^{th}$ particle in $d$ dimension space; $x_{id}$ is location of the $i^{th}$ particle at the $d$ dimensions; $r_2$ and $r_2$ are random number distributed uniformly in (0,1). $c_1$ and $c_2$ are learning factors, usually $c_1$=$c_2$=[1.8, 2]; $p_{id}$ is $p_{best}$ and $p_{gd}$ is $g_{best}$.

# Chapter 4

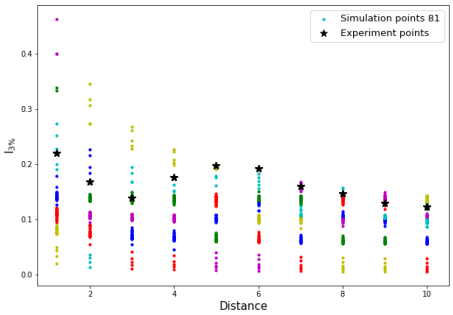
# Mathematical Analysis

## 4.1 Data preprocessing

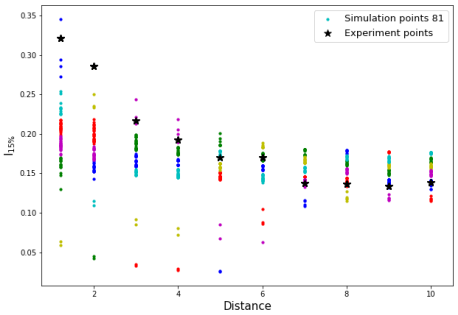
According to the physical experiment of Sandia National Laboratories (SNL), there is a set of state variables under ten different spatial distances. Instead of doing physical experiment, which is quite repetitive, time-consuming, expensive and environmentally damaging, we apply computer simulator and surrogate model to predict the result of the physical experiment under different parameters precisely. However, some of the data generated from the simulator are not as moderate as our thought so we should do data preprocessing to enhance the accuracy of the prediction. Figure 4.1 shows the distribution of the physical experiment and simulation, namely numerical experiments before data cleaning process.

As Figure 4.1 shows, the horizontal axis represents ten distance points, while the state variables $U_{3\%}$, $U_{15\%}$, $I_{3\%}$, $I_{15\%}$ are presented in the vertical axis of each sub-graph individually. We apply k-means clustering algorithm to the data under different distances and divide them into six groups. The group which is the closest to the physical experimental points (marked with black pentagram) is chosen and extracted to be our new set of data.

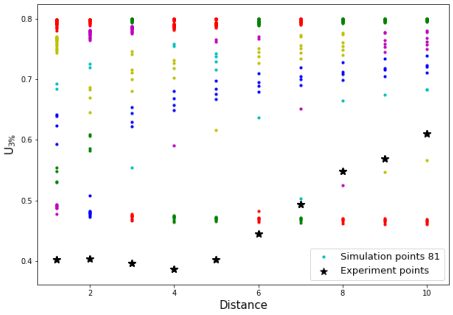
In order to avoid over-fitting, the whole datasets should be divided into 70% of training set and 30% of testing set. Training set is used to build the surrogate model and testing set is to verify the accuracy of the model. In this paper, the number of 97 datasets are separated into 81 datasets for training and 16 datasets for testing. After k-means clustering, we reduce 81 datasets to 36 datasets and retrieve them to be our surrogate model training



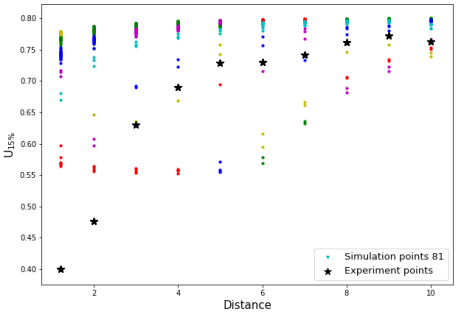

(a) The distribution of $I_{3\%}$ experiment.




(b) The distribution of $I_{15\%}$ experiment.




(c) The distribution of $U_{3\%}$ experiment.

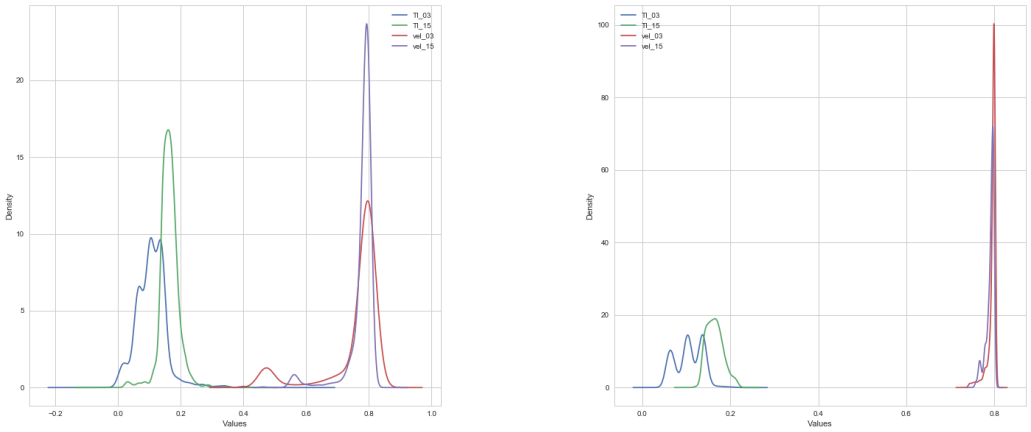



(d) The distribution of $U_{15\%}$ experiment.

Figure 4.1: The distribution of the physical experiment and the numerical experiment.

dataset.

As figure 4.2(a) shows, the retrieved dataset doesn't obey Gaussian distribution so it is necessary to introduce univariate boxplot Frigge et al.(1989) in order to remove the outliers Shevlyakov et al.(2013).




(a) Distribution of state variables with outliers. (b) Distribution of state variables without outliers.

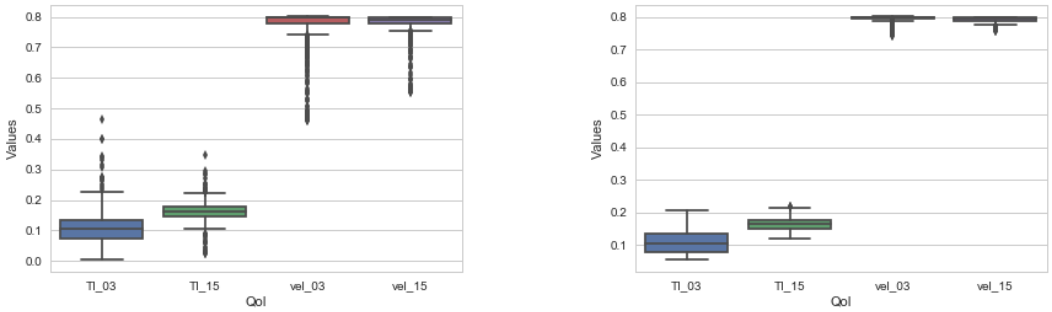
Figure 4.2: Using boxplot to remove the outliers

A univariate boxplot is specified by five parameters: the two extremes, the upper UQ (75th percentile) and lower LQ (25th percentile) quartiles and the median (50th percentile).The lower and upper extremes of a boxplot are defined as

$$x_L = max\left\{x_{(1)}, LQ - \frac{3}{2}IQR\right\}, x_U = min\left\{x_{(n)}, UQ + \frac{3}{2}IQR\right\}. \tag{4.1}$$

Different streams of data are compared via their respective boxplots in a quick and convenient way. It is a common practice to identify the points which are located beyond the extremes (maximum and minimum) as outliers, and mark them in the corresponding boxplots Figure 4.3.

We wish to remove all the outlier points, but once the points have been removed the UQ, LQ, median and two extremes will be changed. It is difficult to retrieve all the points that are between $x_L$ and $x_U$, so now we still have some outlier points in $U_{3\%}$, $U_{15\%}$, $I_{3\%}$, and $I_{15\%}$.




(a) Before boxplot removal (b) After boxplot removal

Figure 4.3: The application of boxplot removal

Table 4.1: The distribution of boxplot removal

(a) Before boxplot removal

| Value | $U_{3\%}$ | $U_{15\%}$ | $I_{3\%}$ | $I_{15\%}$ |
|---|---|---|---|---|
| $x_L$ | 0.4972 | 0.7552 | -0.0549 | 0.0799 |
| LQ | 0.6787 | 0.7797 | 0.0606 | 0.1389 |
| Median | 0.7968 | 0.7929 | 0.119 | 0.1601 |
| UQ | 0.7997 | 0.796 | 0.1376 | 0.1782 |
| $x_U$ | 0.9812 | 0.8205 | 0.2531 | 0.2372 |

(b) After boxplot removal

| Value | $U_{3\%}$ | $U_{15\%}$ | $I_{3\%}$ | $I_{15\%}$ |
|---|---|---|---|---|
| $x_L$ | 0.7693 | 0.7851 | -0.0029 | 0.1022 |
| LQ | 0.7876 | 0.7908 | 0.0675 | 0.1402 |
| Median | 0.7987 | 0.7947 | 0.1292 | 0.1658 |
| UQ | 0.7998 | 0.7965 | 0.1379 | 0.1782 |
| $x_U$ | 0.8181 | 0.8022 | 0.2083 | 0.2162 |

Besides, we use bootstrap method that involves iteratively resampling the 36 datasets with replacement. After this method, we increase the number of datasets and is about to train our model with 100 datasets.

## 4.2 Surrogate Model Development

In this section, we will use equation 2.1 as the evaluative criteria to evaluate the performance of the surrogate model. Luckily, scikit-learn, which is a library containing different kinds of package in the machine learning field, provides us with a cheat-sheet that illustrates four aspect of machine learning methodology such as regression, dimensionality reduction, clustering, and classification. In our case, regression is suitable for the training the surrogate model. The samples in our research are only 810 points which are less than

the number of 100K so it is reasonable to use Lasso, ElasticNet, SVR, RidgeRegression and EnsembleRegressors to build the surrogate model.

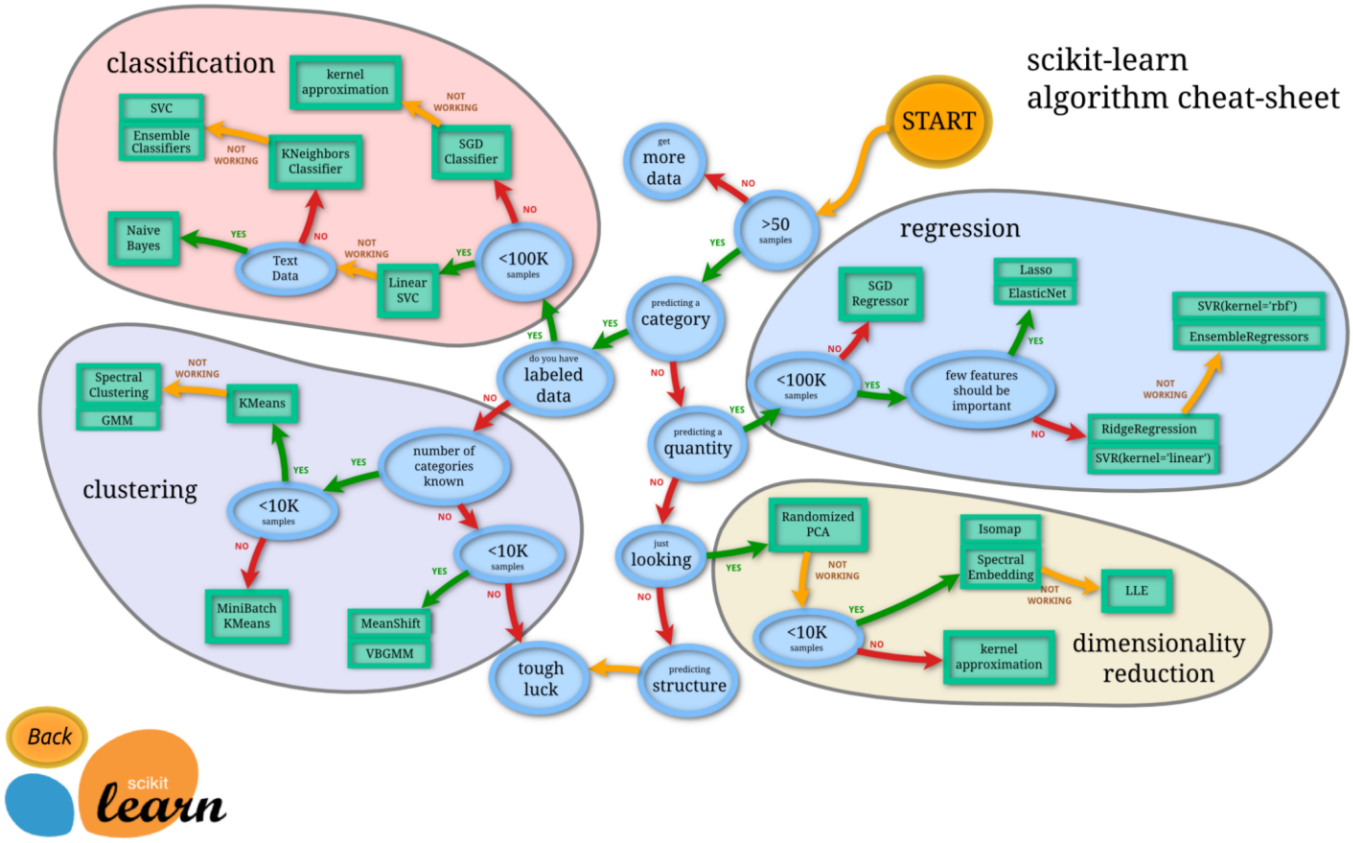



Figure 4.4: Mind map of machine learning

The figure presents information about the total percentage of MAPE in $U_{3\%}$, $U_{15\%}$, $I_{3\%}$ and $I_{15\%}$ predicted by different machine learning models.

Before the data is analyzed, it is difficult to know which is the best one from these twenty-seven models. Through trial and error method, Gaussian model with five kinds of different kernel functions is the fittest, compared with three types of Support Vector Regression (SVR), Lasso and Ridge Regression, Decision tree and so on. All the total MAPE values of Gaussian models are under 10% and $U_{3\%}$ always takes the lowest percentage within these four state variables.

On the contrary, the highest value of MAPE is Multiple Layer Perceptron where MAPE is almost 170% more than seventeen-fold in comparison with Gaussian model.

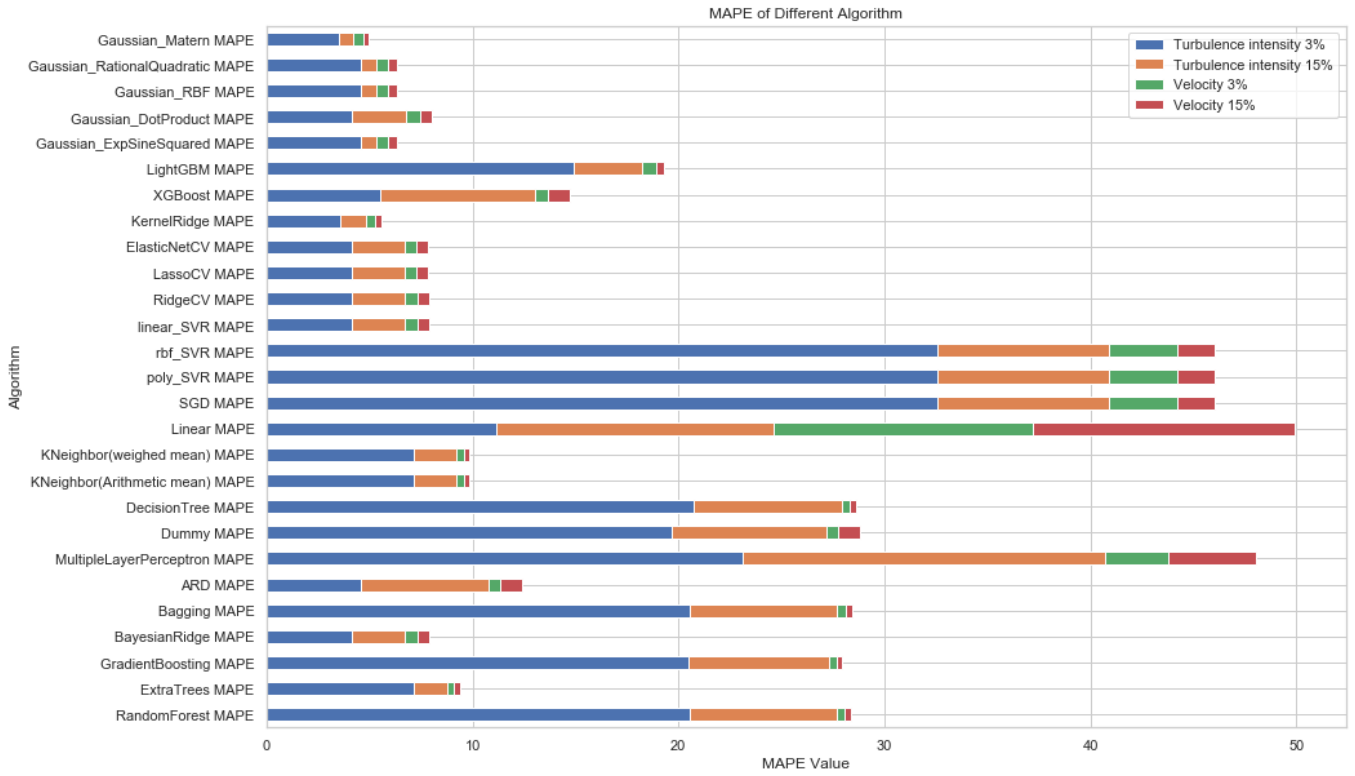



Figure 4.5: The MAPE of different kinds of machine learning regression

Overall, the Gaussian model is noticeably lower in the MAPE of $U_{3\%}$, $U_{15\%}$, $I_{3\%}$ and $I_{15\%}$, so the whole dataset will be trained by this model.

## 4.3 Building Gaussian Process Regression

The training set, which contains 100 datasets and is close to the physical experimental points, is about to build Gaussian model.

The Gaussian Process Regression (GPR) surrogate models are constructed using Python version 3.6.4 with Numpy version 1.14.3 and Pandas version 0.22.0. The GPR is trained by scikit-learn 0.19.1 and the graph is drawn by Matplotlib version 2.1.2. The processor is GPU GeForce GTX 1060 which is extremely faster than the traditional computer equipped with CPU.

The GPR model is trained to minimize the fitting error between the surrogate model and simulator by adjusting the kernel and alpha. The kernel with the lowest MAPE should

be found by testing. In contrast, alpha can be preciously predicted by grid search method which is tuned in the range $10^{-5}$ to $10^{5}$. Table 4.2 shows five different kernels of GPR in $U_{3\%}$, $U_{15\%}$, and $I_{15\%}$ with alpha 0.00001 and the remaining $I_{3\%}$ with alpha 0.0001.

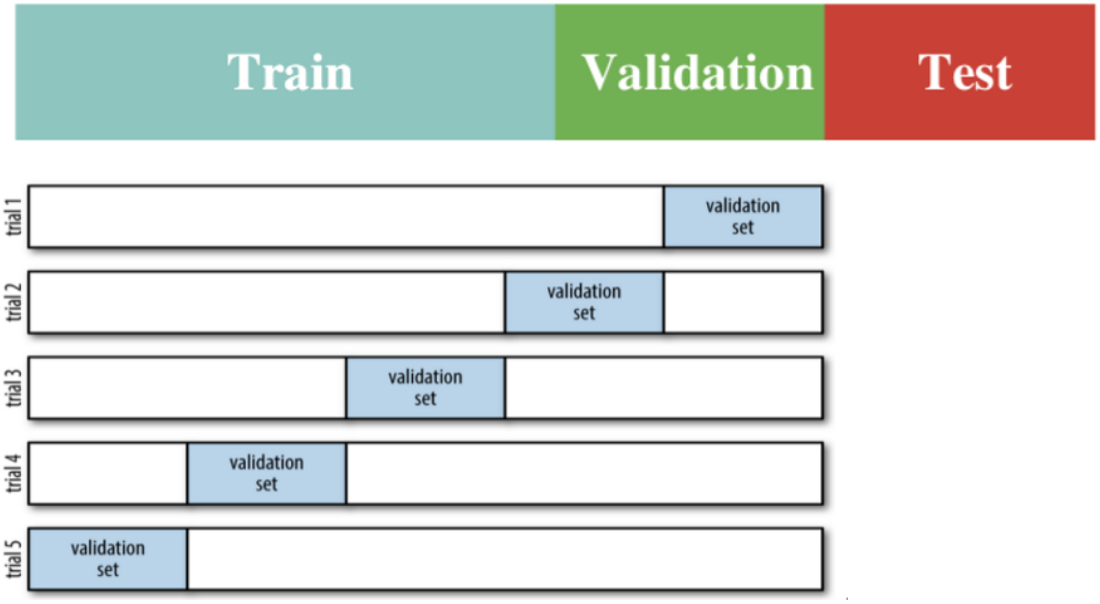



Figure 4.6: Five-fold cross-validation

Surrogate-model fitting error $MAPE_{srgt/sim}$ is an indicator of the performance of five kernels. All the $MAPE_{srgt/sim}$ values are estimated by five-fold cross-validation(see Figure 4.5)—that is, we split the training data(70%) into five sets and use each in turn as validation set to evaluate the model fit on the other four-fifths of the training set. Repeating the validation across different subsets gives us a better idea of the performance of the algorithm in advance. After doing five-fold cross-validation in training set, the GPR model is built and should be tested. Therefore, testing set (30%) plays an important role to test whether the GPR model is precise or not.

Table 4.2: Summary of the covariance functions

| QoI | ExpSineSquared | DotProduct | RBF | RQ | Matern |
|---|---|---|---|---|---|
| $U_{3\%}$ | 5.6% | 3.2% | 5.6% | 5.6% | 2.5% |
| $U_{15\%}$ | 0.6% | 0.6% | 0.6% | 0.5% | 0.6% |
| $I_{3\%}$ | 18.8% | 14.3% | 18.8% | 18.8% | 15.9% |
| $I_{15\%}$ | 1.2% | 2.8% | 1.2% | 1.2% | 1.2% |
| Mean | 6.55% | 5.2% | 6.55% | 6.5% | 5% |

Table 4.2 presents the outcome of the evaluation of testing set (30%)of five different Gaussian kernels. It is shown that the lowest $U_{3\%}$ lays on Matern(2.5%). RationalQuadratic has the lowest number of $U_{15\%}$(0.5%). DotProduct contributes the lowest $I_{3\%}$(14.3%). Each of the ExpSineSquared, RBF, RationalQuadratic, and Matern are all responsible for 1.2% in $I_{15\%}$. There is a narrow gap of MAPE in each of the kernels, so we choose Matern, which is the lowest MAPE in the average of Qol , to build surrogate models.

## 4.4 Finding parameters with Metaheuristic Algorithm

# Chapter 5

# Conclusion

This paper illustrates the use of the Gaussian surrogate methodology to determine a set of unknown physical parameters based on experimental information. We show the potential of using this methodology to reduce the burden of solving marine turbine problems when computationally-expensive numerical simulations are required. Surrogate models are built for an error metric, which compares the numerical model predictions to the experimental model measurements. By minimizing these surrogates, the values of the unknown physical parameters are estimated.

The application of the surrogate methodology to the marine turbine problem provides a framework to enhance the integration of different information sources into the model building process. The advantage of this integration is that none of the information available about the system is neglected when building a model, which reduces the amount of information required from resource-intensive sources such as detailed numerical simulations. 增加數據分析的結論 The numerical models chosen perform well in predicting the experimental model measurements. In fact, the numerical model with four parameters agrees with the measurements from the experimental model with an error of the experimental uncertainty.

# Bibliography


[1] M. Abido. Optimal power flow using tabu search algorithm. *Electric power components and systems*, 30(5):469–483, 2002.

[2] K. Anand, Y. Ra, R. D. Reitz, and B. Bunting. Surrogate model development for fuels for advanced combustion engines. *Energy & Fuels*, 25(4):1474–1484, 2011.

[3] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, and P. Sagaut. Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers & Fluids*, 40(1):195–209, 2011.

[4] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3):256–279, 2004.

[5] M. Frigge, D. C. Hoaglin, and B. Iglewicz. Some implementations of the boxplot. *The American Statistician*, 43(1):50–54, 1989.

[6] D. W. Heermann. Computer-simulation methods. In *Computer Simulation Methods in Theoretical Physics*, pages 8–12. Springer, 1990.

[7] J. H. Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[8] J. Kennedy. Particle swarm optimization. *Encyclopedia of machine learning*, pages 760–766, 2010.

[9] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[10] J. rey Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence*, volume 1, pages 82–87. Citeseer, 1994.

[11] G. Shevlyakov, K. Andrea, L. Choudur, P. Smirnov, A. Ulanov, and N. Vassilieva. Robust versions of the tukey boxplot with their application to detection of outliers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6506–6510. IEEE, 2013.

[12] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

[13] P. J. Van Laarhoven, E. H. Aarts, and J. K. Lenstra. Job shop scheduling by simulated annealing. *Operations research*, 40(1):113–125, 1992.

[14] F. Wang and D. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

[15] W. Yamazaki, M. Rumpfkeil, and D. Mavriplis. Design optimization utilizing gradient/hessian enhanced surrogate model. In *28th AIAA Applied Aerodynamics Conference*, page 4363, 2010.