Dual Mass Flywheel Design Optimization Based On Differential Equation – Neuroregression Modeling Approach

Submitted to the Graduate School of Natural and Applied Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in Mechanical Engineering

by

Ümmühan Gümüş

ORCID 0000-0002-5054-6859

January, 2022

This is to certify that we have read the thesis **Dual Mass Flywheel Design Optimization Based On Differential Equation – Neuroregression Modeling Approach** submitted by **Ümmühan Gümüş**, and it has been judged to be successful, in scope and in quality, at the defense exam and accepted by our jury as a MASTER'S THESIS.

APPROVED BY:

Advisor:	**Assoc. Prof. Dr. Levent Aydın** İzmir Kâtip Çelebi University	……………..

Committee Members:

Assist. Prof. Dr. Ebubekir Atan İzmir Kâtip Çelebi University	……………..
Assoc. Prof. Dr. Gözde Değer Sarı Manisa Celal Bayar University	……………..

Date of Defense: January 25, 2022

Declaration of Authorship

I, **Ümmühan Gümüş**, declare that this thesis titled **Dual Mass Flywheel Design Optimization Based On Differential Equation – Neuroregression Modeling Approach** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for the Master's / Doctoral degree at this university.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. This thesis is entirely my own work, with the exception of such quotations.
- I have acknowledged all major sources of assistance.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signature:

Date: 25.01.2022

Dual Mass Flywheel Design Optimization Based On Differential Equation – Neuroregression Modeling Approach

Abstract

Torque fluctuations occur during combustion cycles of a four-stroke engine, and these fluctuations are transferred to the drivetrain of a vehicle as torsional vibration. Hence, the noise and vibration in gears clash, ping, and load change vibration cause jerky motion and reduced driving comfort. In vehicles, it is desirable to transmit this vibration generated by the engine to the drivetrain of the vehicle at the lowest level. Different technologies are used for this purpose. One of them is Dual Mass Flywheel (DMF). DMF is a complex system that includes rotational inertia, torsional stiffness, and damping properties. In this thesis, a comprehensive study is presented in order that the transmission of vibration generated in the engine of a heavy vehicle to the drivetrain as minimum amplitude. It is aimed to obtain the best DMF model. Firstly, a simplified mathematical model based on a system of an ordinary differential equation (ODE) form is used to show the positive effect of the DMF on the powertrain. The ODE system is solved for different input parameters' values, and then a data set is generated utilizing the full factorial design. A neuro-regression model is constituted using these data, and then the optimization of system parameters is performed by using stochastic technics *Differential Evolution*, *Nelder–Mead*, *Simulated Annealing,* and *Random Search*. As a result of the study, it is observed that the improvement is over 28%.

Keywords: Dual Mass Flywheel, Mathematical Modeling, Optimization

Diferansiyel Denklem – Nöroregresyon Modelleme Yaklaşımı Tabanında Çift Kütleli Volan Optimum Tasarımı

Öz

Dört zamanlı bir motorun yanma çevrimleri sırasında tork dalgalanmaları oluşur ve bu dalgalanmalar aracın aktarma organlarına burulma titreşimi olarak geçer. Sonucunda dişli vuruntusu, gövde uğultusu ve yük değişim titreşimi şeklinde kendini gösteren bu gürültü ve titreşim, sarsıntılı çalışmaya ve sürüş konforunun azalmasına neden olur. Araçlarda motordan kaynaklanan bu titreşimin, aracın aktarma organlarına en düşük seviyede iletilmesi istenmektedir. Bunun için farklı teknolojiler kullanılmaktadır; bunlardan biri Çift Kütleli Volan (ÇKV)'dır. ÇKV, dönme ataleti, burulma sertliği ve sönümleme özellikleri içeren karmaşık bir sistemdir. Bu tezde, ağır vasıta bir aracın motorunda oluşan titreşimin aracın aktarma organlarına en düşük genlikle aktarılması amacıyla belirlenen bir problemin çözüm aşamalarını içeren bir örnek çalışma sunulmaktadır. Bir ÇKV'nin güç aktarma organları üzerindeki olumlu etkisini göstermek için adi diferansiyel denklem sistemi formundaki, ideal durumlar için basitleştirilmiş bir matematiksel model kullanılmıştır. Bu adi diferansiyel denklem sistemi, Mathematica programı kullanılarak farklı girdi değerleri için çözülmüş ve daha sonra "tam faktöriyel" tasarım yaklaşımıyla bir veri seti elde edilmiştir. Bu veri değerleri ile bir nöroregresyon modeli oluşturulmuş ve daha sonra stokastik optimizasyon yöntemleri *Differential Evolution*, *Nelder–Mead*, *Simulated Annealing* ve *Random Search* kullanılarak sistem parametrelerinin optimizasyonu gerçekleştirilmiştir. Çalışma sonucunda, iyileşmenin diferansiyel denklem çözüm sistemi ile elde edilen en küçük değere oranla %28'in üzerinde olduğu görülmüştür.

Anahtar Kelimeler: Çift Kütleli Volan, Matematiksel Model, Optimizasyon

To my family…

Acknowledgment

First and foremost, I would like to express my endless thanks to my academic advisor Assoc. Prof. Dr. Levent Aydın, who has always supported me, who has been working hard on me for about 8 years. He never withheld his knowledge from me, rather he worked patiently to teach something more. I will always be proud to work with him and to be his student.

I should state that I am grateful to my family who supported me in my nearly 20 years of education under all circumstances, who always stood behind me and who did not spare their moral support.

Lastly, I am very thankful to my beloved friend Emre Özeğdemir for his friendship, caring, support, and motivation.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ODE	Ordinary Differential Equation
DE	Differential Evolution
SA	Simulated Annealing
NM	Nelder–Mead
RS	Random Search
DMF	Dual Mass Flywheel
NVH	Noise, Vibration, and Harshness
ANN	Artificial Neural Networks
DMTVDA	Dual Mass Torsional Vibration Dynamic Absorber

List of Symbols

$M_e(t)$	Engine torque
$\varphi_1(t)$	Absolute angle of rotation of the primary flywheel
$\dot{\varphi}_1(t)$	Absolute angular speed of rotation of the primary flywheel
$\ddot{\varphi}_1(t)$	Absolute angular acceleration of the primary flywheel
$\varphi_2(t)$	Absolute angle of rotation of the secondary flywheel
$\dot{\varphi}_2(t)$	Absolute angular speed of rotation of the secondary flywheel
$\ddot{\varphi}_2(t)$	Absolute angular acceleration of the secondary flywheel
$\varphi_v(t)$	Absolute angle of rotation of the output
$\dot{\varphi}_v$	Absolute angular speed of rotation of the output
k_1	Torsional stiffness coefficient of the primary flywheel
c_1	Torsional damping coefficient of the primary flywheel
k_2	Torsional stiffness coefficient of the secondary flywheel
c_2	Torsional damping coefficient of the secondary flywheel
$M_v(t)$	Output torque
Y	Nonlinear regression model
f	Function
X	Vector of p predictors
β	Vector of k parameters
ε	Error term
$P(E)$	The Probability of Achieving the Energy Level (E)
k	The Constant of Boltzmann
T	Temperature

J^T	Jacobian
J	Traditional Jacobian
W	Diagonal weighting matrix
λ	Damping parameter
I	Identity matrix
h	Perturbation
y	Measured point set
$\hat{y}$	Fitted function

Chapter 1

Introduction

1.1 Objectives and Motivation

Every time an engine fires, a shock occurs, which is commonly known as “Noise, Vibration, and Harshness” (NVH). This is transmitted to the transmission via the crankshaft. Furthermore, any moving parts bolted to the engine will contribute to the overall NVH.

The torsional vibration emitted by today's engines increases by lighter engines contributing to better fuel economy, combined with higher combustion pressures to lower emissions, and conventional clutch plate dampers cannot meet customer comfort expectations.

If not addressed, NVH will enter the gearbox, causing premature wear and failure. It will then be transmitted into the vehicle's cabin, resulting in reduced driver comfort and, in the case of vehicles used every day, the possibility of injury.

Companies that operate fleets must be more cautious than ever before about the vehicle in which they send their employees out in an increasingly litigious society. Similarly, they will be acutely aware of the cost of vehicle downtime and should do everything possible to avoid such situations.

The Dual Mass Flywheel is a product of vehicle evolution. It is a solution to the problem of excessive vibration caused by modern vehicle standards.

In this thesis, the relative displacement $\varphi_1 - \varphi_2$ value minimization study of Dual Mass Flywheel design transfers the vibration from the engine to the driveline at the lowest amplitude. Optimization of dual mass flywheel design has been accomplished by four different stochastic algorithms, Differential Evolution (DE), Simulated Annealing (SA), Random Search (RS), and Nelder–Mead (NM).

The objectives of this study can be considered as follows; (i) Defining the phenomenon by the mathematical model in the best way. (ii) Verifying the mathematical model as statistically. (iii) Defining the objective function, constraints, and design variables to solve the optimization problem. (iv) Comparison of four different stochastic optimization algorithms.

It should be noted that defining the best mathematical model is not simple. In the current study, the results of different literature studies were examined, and optimization process studies were carried out to show the deficiencies in these studies, report the problems, and make improvements in the results. If we briefly mention why the results of these literature studies are examined and why these studies are carried out, the R^2 value cannot be evaluated by using only one or two regression models as a decision criterion. It is not usually the case that a strong R^2 correlation equates to a fine match for acceptable structures. Determining the basic actions of the phenomenon requires fresh modeling experiments using alternative regression techniques. Furthermore, the models established in the academic research have no limited control. For this reason, limitedness must be controlled if the engineering parameter ranges relate to the selected models.

The organization of the thesis study is as follows;

This study consists of 4 parts. In Chapter 1, why this topic chosen and the importance of the study has been mentioned. Studies on dual-mass flywheel design as Theoretical and Experimental, Experimental, Theoretical, and Numerical have been mentioned in literature. A brief description of the problem studied in the thesis is presented. Chapter 2 surveys the fundamental mathematical instruments that will show throughout this study. The main areas enclosed in this study are the design of experiments, regression analysis, neuro-regression approach, and boundedness concept. The main aim of the chapter is to show the basic and modern approaches to the modeling and design of engineering structures in a concise manner. In Chapter 3, four different stochastic algorithms, Differential Evolution (DE), Simulated Annealing (SA), Random Search

(RS), and Nelder–Mead (NM) used in the study, are explained. In Chapter 4, general information about optimization is given, and it is explained how they are used in the Mathematica program which variations of these optimization methods. Chapter 5 describes the optimization work of the dual-mass flywheel, which ensures that the vibration is transmitted at low amplitude as it is transmitted to the vehicle's driveline. The differential equation system is solved numerically to model the engineering problem at this stage. The results obtained are converted into data in the form of variations depending on the inputs and expressed with the neuroregression approach.

Short Problem Description:

The mathematical model of the problem and the motor torque are given in Equation 1.1 and Equation 1.2. The φ_1 and φ_2 in the equations represent the angular displacements of the first and second flywheels, respectively. Its derivatives represent angular velocities, and second derivatives also represent angular accelerations. It is shown as a free body diagram in Figure 1.1.

$$J_1\ddot{\varphi}_1 + c_1(\dot{\varphi}_1 - \dot{\varphi}_2) + k_1(\varphi_1 - \varphi_2) = M_e(t)$$
$$J_2\ddot{\varphi}_2 + c_1(\dot{\varphi}_2 - \dot{\varphi}_1) + k_1(\varphi_2 - \varphi_1) + c_2(\dot{\varphi}_2 - \dot{\varphi}_v) + k_2(\varphi_2 - \varphi_v) = 0 \quad (1.1)$$

$$M_e(t) = M_0 + M_1 * \sin(w_e t + \alpha_1) \quad (1.2)$$

where α_1 is phase angle, M_0 is constant torque, M_1 represents the wave's amplitude. Lastly, it is assumed that the output side of the system does not have any vibration.

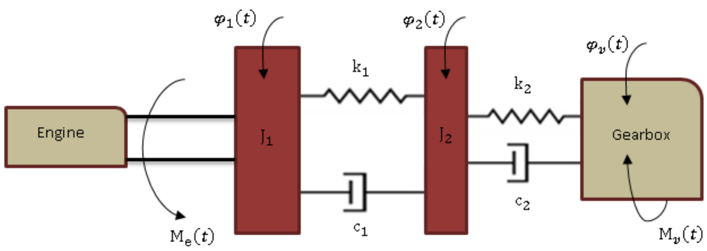


Figure 1.1: Free body diagram of DMTVDA model

In order to solve the optimization problem, the objective function $(\varphi_1 - \varphi_2)$ and design variables $(J_1, J_2, k_1, k_2, c_1, c_2)$ are defined.

1.2 Literature Survey

One of the crucial factors affecting vehicle performance is the torsional vibrations that occur in the clutch system, and these vibrations also cause noise in the power transmission. In today's technology, vehicle designers use dampers, couplings, and similar products to reduce torsional vibrations or provide isolation [1]. Consequently, it becomes apparent that conventional damping elements do not provide sufficient vehicle comfort due to the limitations in damping and limited volumes in the installation. For these reasons, Dual Mass Flywheel appeared in the 1980s. In this respect, DMF is a highly efficient vibration damping element, widely used in passenger and commercial vehicles, more preferred in diesel vehicles [1]. The resonance value for torsion in traditional single mass flywheels is between 700 and 2000 rpm. DMF aims to reduce the torsional resonance of the clutch system below the engine resonance. This is provided by having an additional mass on the transmission input shaft side. DMF also aims to reduce the hysteresis in the system and to dampen the power fluctuations from the motor shaft [2]. The inclusion of DMF also increases driver comfort and allows the engine to run at lower speeds, resulting in lower fuel consumption [3].

Studies on the related area in the literature can be examined in three main sections.

1.2.1 Theoretical and Experimental

Subsequent studies have been based on various methods to investigate the effect of DMF on the transmission system, and mathematical models have been established. In order to see the accuracy, the results of the experimental setups were compared. As the torsional stiffness changes depending on the rotational speed, it also affects the damping characteristic of DMF. Therefore, the effect of torsional stiffness is one of the topics studied [3–6].

A study aims to further improve the counter-torque of a DMF with single-stage stiffness at large torsional angles. Besides, it is used to model the contact between the friction bearing and the secondary flywheel [3]. In a different study, it was found that the two stages of torsional stiffness were calculated considering the frictional forces. For Circumferential Arc Spring Dual Mass Flywheel (DMF-CS), the repetitive formulas of the springs in the system were calculated by a different method, including friction forces [4]. There are also studies on multistage DMF design methods on torsional vibration control and the compatibility of kinetic parameters. Studies have shown that the amplitude of the angular velocity in the gearbox input shaft is reduced by using three-stage DMF, and torsional vibrations can be effectively controlled [5].

The friction characteristic of the arc springs is another essential point to be emphasized. For this reason, studies have been carried out on this subject [4, 7]. The study by Kim et al. 2006, it is performed a discrete analysis approach for DMF performance and efficiency. The arc spring is modeled as an n-discrete element and is placed between the flywheels, and each element contains components such as mass, spring, and nonlinear friction elements. In order to explain the viscous friction and Stribeck effect due to relative sliding velocity, a nonlinear element model is defined. To define the arc spring friction behavior, they have introduced LuGre's model. MATLAB Simulink is also often used in such studies [7–9].

Alternatively, a new DMF structure with variable stiffness has been investigated in order to determine the characteristics of the power transmission system in [6]. The resonance speeds of the torsional stiffness in orders (first and second) have been analyzed, and the model is transformed into fourth-order linear algebraic equations. Since DMF cannot be defined as a linear dynamic model, a suitable model should be selected to determine the ignored nonlinear properties, creating a nonlinear dynamic model [10, 11]. The researchers consider the nonlinear stiffness, and the parameters in the Bouc–Wen model (the hysteresis caused by friction torque) are estimated by obtaining dynamic test data. When the Levenberg-Marquardt and Gauss-Newton methods were compared, the first method showed more efficient and accurate results [10]. In the Coulomb method, a quasi-steady state approach was used to simplify the complexity of the model for arc springs [11]. In the DMF model, the motor torque's observability was studied using the Ab-initio method [12]. In another study, optimization of DMF and clutch system was emphasized. It was found that the theoretical studies were compatible with the experimental models, and vibration was reduced by the optimization of the parameters required for the design [10]. Many methods have been used in this subject, and the theoretical studies have agreed well with experimental studies and contributed to its development. Therefore, the contribution of the studies in this field is significant.

1.2.2 Experimental

By establishing test benches and models, the effects of torsional vibrations on the transmission of a dual-mass flywheel (DMF) and noise and vibration improvements were investigated [2, 13–15]. In a study involving many powertrain components, torsional vibration signals were obtained by equal angles sampling and order analysis method. The torsional vibration data are processed and analyzed with the help of the MATLAB® program under constant speed conditions, and the order signal of the engine excitation is obtained [13]. In another study, a test setup was established to investigate the effect of DMF on impact-induced clonk noise. Extensive measurements using DMF and the conventional clutch system showed significant differences in vibration and noise values in the driveline, resulting in differences in the number of modes of both parameters [2]. Dual Clutch Transmission (DCT) DQ250 was used to observe and verify the damping ability of the DMF, and the experimental setup was established. In this experiment, various conditions were created and tested for the vehicle, such as an idle, working, and climbing condition. According to the obtained test data, different engine revs and loads affect the damping ability of the DMF. As the load and rev on the engine increase, the dual mass flywheel has been observed to have a better damping ability [14]. Displacement measurements on DMF have been developed with a linear variable differential transformer (LVDT) mounted on a vehicle. Unlike other experimental studies, the data are obtained using sensors, power and Bluetooth TM node connection and transmit the data via Bluetooth technology. It is possible to obtain more direct results with this measurement system developed for DMF.

1.2.3 Theoretical and Numerical

In some studies, there are theoretical and numerical approaches to investigate the effects of a dual-mass flywheel on vibration, including mathematical models created using only various methods [1, 9, 16–19]. For example, in [16], the linear approach is preferred by simplifying the model of DMF-CPVAs setup. The equation of motion for the setup has been derived based on Lagrange's method. After analyzing the setup's ability, it is observed that the utilization of CPVAs on the DMF is better on vibration isolation than damping vibrations at a specific frequency. The idle speed of a heavy vehicle was considered in [1, 9]. In these researches, the vehicle dynamics were determined with the help of the ADAMS program. In the new dual-mass flywheel structure, the springs are distributed on both sides of the damping disc, and the damping effect is examined with the help of simulation studies [1]. Idling natural frequency and idling natural vibration model were calculated by applying generalized Jacobian and MATLAB procedures. According to the results obtained, it is seen that DMF torsion absorber and power transfer torsion absorber are kept away from the idle condition when compared. Secondly, they form low-frequency torsions in the power transmission system. In a different study, analyses were performed on the segmented linear model, and the frequency of the model under sinusoidal response was obtained. Inertia, torsional stiffness, and damping values were obtained when the studies and calculations were compiled. According to the analysis, when the inertia of both flywheels is increased, the vibration damping ability of dual mass flywheel increases [17]. The calculation of the natural frequency of the whole system is valid only if the values such as vehicle inertia, transmission gear ratios, and tire torsional stiffness are known. In order to reduce the acceleration of the secondary mass, it is provided by gear increase or reduction of the angular speed [18]. In another study, the Python program has investigated the friction between the primary flywheel of the dual mass flywheel and the arc spring [19]. The simulations were made by combining Newmark and Newton methods. The accuracy of the model has also been checked in programs as AVL Excite. Columb friction model or inverse tangent functions were used to model the friction between two surfaces. According to the results, it is observed that if the friction and viscous damping values are low, it does not cause a significant problem at low engine speeds.

In addition, the dual mass flywheels observed in the studies may also vary structurally. Therefore, in addition to addressing the dual mass flywheel, mechanisms such as DMF-CPVAs – (Centrifugal Pendulum Vibration Absorbers) [16], DMF-CS – (Circumferential Arc Spring Dual Mass Flywheel) [4, 17, 20] were also encountered.

CHAPTER 2

Mathematical Background

2.1 Introduction

This introductory chapter surveys the fundamental mathematical instruments that will show throughout this study. Many researchers will be familiar with several of the topics given here. The reader can figure out which areas need to be studied further. The main areas enclosed in this study are the design of experiments, regression analysis, neuro-regression approach, and boundedness concept. The main aim of the chapter is to show the basic and modern approaches to the modeling and design of engineering structures in a concise manner. They are given in Figure 2.1, all these steps without breaking the order of the optimal design process for a system.

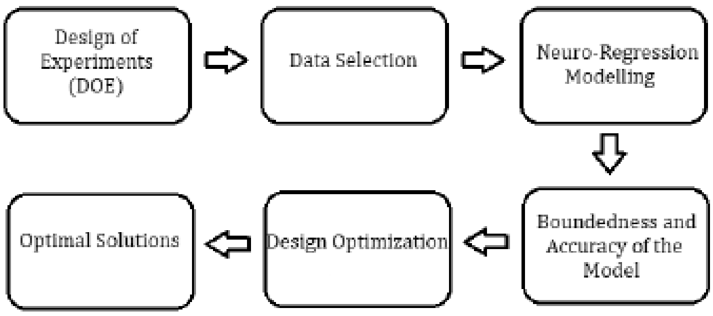


Figure 2.1: Flow diagram of an optimal design process

2.2 Design of Experiments (DOE)

Design of Experiments (DOE) is a helpful tool for finding new processes, learning more about the existing processes, and optimizing them for an excellent performance. In this section, we present and discuss some DOE techniques. As mentioned before, the list of techniques studied is far from entire as this section aims to announce the reader to the issue by presenting the basis techniques used in practice. It is crucial to choose adequate statistical tools to analyze the data available since the results can be markedly changed by noise. In DOE, replication, randomization, and blocking are the main principles of statistical approaches. Replication is the experiment's repetition to achieve a more accurate result and reduce the error of the experimental. Randomization identifies the random order in which the experiment runs. Blocking is intended to isolate an acknowledged systematic bias effect and inhibit obscuring the main effects [21, 22]. Due to the number of crunchers concerned and complicated statistical jargon, engineers generally do not like to employ DOE [22]. In manufacturing processes, experiments are being carried out to enhance our knowledge and understanding. The relationships between the core factors of the inputs and the behaviors of the output can, therefore, be examined [22, 23]. One of the comprehensive strategies in engineering companies promoted by many engineers is One-Variable-At-a-Time (OVAT). This approach changes one parameter at a time, with all other factors fixed throughout the experiment. The results, however, are unreliable, wasteful, and may offer the processes a misleading conclusion. If a particular attribute of a component is affected by several factors, then the best choice is DOE [22, 23, 24]. The engineer frequently makes systematic changes in the input parameters and specifies how well the output performance varies. It is known that all parameters do have not the same impact on the results. Therefore, the aim of a deliberate design is to realize which process parameters tend to affect the output more and then to identify the best levels for the factors [21, 22, 25]. This approach provides high process efficiency, more stable results, low manufacturing costs, and saves time for the researchers.

2.2.1 DOE Techniques

The selection of a DOE approach relies on the experiment's goals and the number of factors to be considered. In this section, the most widely used approaches have been listed and explained briefly. These approaches are Randomized Complete Block Design, Full Factorial, Fractional Factorial, Central Composite, Box-Behnken, Taguchi, Latin Hypercube, and D-Optimal Design [21].

2.2.1.1 Randomized Complete Block Design

The distribution of treatment for experimental components is not strictly limited. In practice, however, there are situations in which the experimental data vary relatively widely. In such situations, the design made is called a Randomized Complete Block Design (RCBD). The main goal of blocking is to minimize the variability between experimental units within a block and maximize the variation between blocks.

Advantages of the RCBD

1. It is possible to remove the treatments or replicates from the analysis.

2. Several treatments can be more often replicated than the others.

3. There is no strict limitation on the number of treatments or replicates.

4. Even if the experimental error is not homogeneous, there can still be valid comparisons [22, 26].

Disadvantages of the RCBD

1. There exists a smaller error on *df* for a small number of treatments.

2. If the number of treatments is enormous and there is a considerable variation between experimental units, it is possible to obtain a significant error term.

3. RCBD is not very good on the efficiency of the experiment when there is missing data.

2.2.1.2 Full Factorial Design

In the literature, the DOE method widely utilized in fabricating industries is fractional and full designs for two and three levels. Factorial designs might allow a researcher to explore a response consistent with the impact of the variables. A factorial design may be separated as fractional or full factorials. An experimental design that every factor setting occurs with every other one is a full factorial design. If the amount of factors is five or higher, a full factorial design needs a significant amount of runs and is not very useful. A fractional factorial design is a better choice [22, 23].

2.2.1.3 Fractional Factorial Design

Researchers usually do not have sufficient time, money, and funding to conduct experiments of full factorial. If some higher-order relationships are not essential, the primary effects and two-order interactive relations may be acquired by operating only a fraction of the full factorial experiment. A fractional factorial design is defined as a form of orthogonal array layout that enables researchers to explore significant impacts and the required impacts of relationships with a minimum amount of exercises or experimental runs [22, 23, 26, 27].

2.2.1.4 Central Composite Design

The design of the most famous surface is a central composite, which creates a factorial design. Five factorial levels are desired for a central composite design. One of the most critical advantages is that the corner points are checked; if it is shown that curvature is not substantial, then it is accomplished. On the other hand, if the curvature exists, the primary task is to generate the star runs [24].

2.2.1.5 Box-Behnken Design

Box-Behnken Design is based on the cube edge midpoints rather than the corner points, which lead to fewer runs; however, apart from the Central Composite Design, all runs must also be done. It should be noted that only three-factor levels are appropriate in The Box-Behnken Design. However, it has advantages if the curvature stated in the screening experiment is likely necessary [22].

2.2.1.6 Taguchi Design

Taguchi methods, or sometimes called robust design methods, are statistical methods. The primary purpose is to keep the output fluctuation minimal, even in the appearance of noise. The technique significantly increases the efficiency of engineering. The Taguchi design helps to guarantee product quality by deliberately considering the noise factors and the number of mistakes in the area. This approach centers on enhancing the primary function of the design process, thereby promoting flexible designs [22, 26].

Advantages

1. It is simple and easy to use in several other engineering circumstances, enabling it as a robust yet straightforward tool.

2. It underlines, within some qualification constraints, an average production latent value comparable to the final value instead of just a value, thus enhancing the quality of the product.

3. Without an impractically large number of testing, it enables the investigation of many distinct variables.

Disadvantages

1. Precisely that the acquired findings are only comparative and do not specify which variable has the most significant impact on the characteristic value of the product.

2. It can not be used in all interactions among the parameters since orthogonal arrays do not examine all parameter combinations.

3. It is hard to account for parameter interactions.

4. It is offline and hence improper for a procedure that changes dynamically, as in a computer simulation.

2.2.1.7 Latin Hypercube Design

It is a method for generating a near-random case of criterion values from a Multidimensional Distribution and the generalization of the Latin Square Concept to an arbitrary number of dimensions. In this approach, the first step is to identify how many sample points to address and through which column and row the case point was drawn for each sample point. Latin Hypercube method guarantees that the set of random numbers represents the real fluctuation, while standard accidental sampling is only a set of random numbers with no ensures [21–23].

2.2.1.8 Optimal Design (D-Optimal)

It is a computer-aided design that includes the finest portion amongst all feasible experiments. Software tools may also have distinct processes to generate D-optimal designs because the final design may vary depending on the tool used [22, 27]. A selection method creates the finest design based on a chosen factor and a specified amount of test runs. This approach is especially helpful if classical design methods are not being used. These situations are:

1. If supplies or factor configurations are restricted.

2. If it is needed to reduce the number of design runs.

3. When using the operation and mixing variables in the same design.

4. When the experiments already carried out must be included.

5. If the region of the experiments is unstable [22, 24, 25].

2.3 Mathematical Modeling

When examining the literature, studies performed for the intent of engineering optimization, some inadequate approaches were recognized and listed below:

i) Since it is needed for consideration the interaction between all constructional and experimental criteria, in terms of optimization, updating in the one input with preserving the other constant is not a satisfying description, and this approach leads to disregarding the nonlinear influence of input variables.

ii) On modeling and optimization, most of the works choose just one or two classic regression models as an objective function for the problem of optimization. Further computation of the R^2 value of the model for experimental works is the basis issue. However, a high R^2 value does not identify all the physical phenomena of the engineering process. The value of R^2 is the closeness of the fitted model results to the experimental output. On the other hand, the value of R^2 does not always mean an acceptable fit, even if it is ultrahigh for the real systems. Besides, the model identifies only the experimental result but not the fundamental attitude of the phenomena. Therefore, it is needed to attempt new engineering modeling works, comprising different regression forms and approaches.

iii) Apart from these, another necessary feature of the engineering model function should be bounded. Boundedness is related to the realistic modeling of engineering systems, and it is known that all the engineering parameters are finite. Therefore, before the optimization step, it should be controlled which of the selected models are also bounded under the engineering parameter intervals.

iv) Some of the available studies on the optimization of engineering systems do not take accounts of the reliability, sensitivity, and robustness of the algorithms. However, this is critical to find out subsistent behaviors of the stochastic search processes.

Due to these reasons, we introduce a different approach to the modeling-design-optimization process to optimize the engineering input parameters through the study. First of all, a detailed study on multiple nonlinear neuro-regression analyses, in conjunction with linear, quadratic, trigonometric, logarithmic, and their rational forms for the prescribed problem (output). After that, the boundednesses of the candidate models are checked to provide generating realistic values. Eventually, the novel direct search methods, including stochastic ones, are performed.

The first step of the optimization process of engineering design problems is Mathematical modeling. For this reason, the researchers try to tell the difference of mathematical models by utilizing Regression Analysis (RA), Response Surface Methodology (RSM), Finite Difference Technique (FDT), and Artificial Neural Networks (ANN). The obtained mathematical model is also suitable with the objective function of the prescribed optimization problem.

2.3.1 Neuro-regression Approach

In the modeling stage, a hybrid technique that combines the strengths of regression analysis and Artificial Neural Network (ANN) is used to improve the accuracy of the predictions. In this approach, all data is divided into two sets such that 80% and 20% of the given data and the first portion of the data is used for training; the second portion is for testing. In the training process, the goal is to minimize the error between the predicted and experimental values by adjusting the regression models and their coefficients. The test step is then conducted to achieve the outcomes of the prediction by minimizing the influences of regression model discrepancies, and this helps to bring insight into the candidate model's predictive capacity. Second, it is essential to verify the candidate model's boundaries for prescribed values to demonstrate if the model is realistic or not. In this regard, the maximum and minimum values of the models in the given interval for each design variable are calculated after obtaining the appropriate models in terms of R^2 training and testing. As a result, chosen models meet the numerous criteria required for reality.

2.3.2 Nonlinear Regression Analysis

Nonlinear regression models are those that are not linear in their parameters and can be used for three different purposes [28]:

1. To test (or compare the hypothesis) the validity of the model,

2. Characterize the model (i.e., to estimate the parameters),

3. Predict the behavior of the system (interpolation and calibration).

The nonlinear regression model can be written as specified in the following equation, Equation 2.1:

$$Y = f(X, \beta) + \varepsilon(1.1) \tag{2.1}$$

Where:

X is a vector of p predictors, β is a vector of k parameters, $f(-)$ = a known regression function,

• ε = an error term.

For nonlinear regression, mathematical modeling processes can be carried out systematically considering the essential features mentioned in the following items:

a) Nonlinear regression is more flexible than linear regression because the function does not need to be linear or linearizable. Therefore, the nonlinear regression phenomenon provides a wide selection of options to match the data.

b) Nonlinear regression may be more appropriate than transformations and linear regression, where the f function can be linearized.

c) Nonlinear regression requires a knowledge of the function f such as polynomial, trigonometric, exponential, which requires a thorough understanding of the studied process. Linear regression models, on the other hand, are suitable for process estimations that are roughly certain of the relationship between input and output but do not require precise clarity.

d) Since nonlinear regression models contain the most general mathematical expressions, it is not possible to write functionally generalized states. However, a few basic types of models used in the engineering fields can be expressed as follows:

Examples of nonlinear equations are:

$$y = a_0 + a_1 x + a_2 x^2 + \dots a_n x^n \tag{2.2}$$

$$y = a_0 + a_1 e^{x} + a_2 e^{x^2} + \dots a_n e^{x^n} \tag{2.3}$$

$$y = a_0 + a_1 \sin(x) + a_2 \sin(x^2) + \dots a_n \sin(x^n) \tag{2.4}$$

$$y = \frac{a_0 + a_1 x + a_2 x^2 + \dots a_n x^n}{b_0 + b_1 x + b_2 x^2 + \dots b_n x^n} \tag{2.5}$$

At this stage, the multivariable states of the abovementioned model types containing more than one input can also be derived with similar logic. Another critical point is that, for example, special functions such as Bessel, Laguerre, Lambert, and Gamma or different combinations of elementary functions can be selected as model structures with a broader understanding of the families of mathematical functions.

CHAPTER 3

Stochastic Optimization Methods

3.1 Introduction

Finding an approximation of an optimal solution for a function that is defined on a subset of finite-dimensional space is one of the most common problems in applied mathematics. In combinatorial optimization problems, which are crucial for most machine learning approaches, some objective functions should be optimized to find an approximation of an optimal solution. Fifty years ago, there were a lot of numerical optimization procedures for these optimization problems; most of them were deterministic (traditional optimization techniques). However, with the development of computer technology, stochastic methods (non-traditional optimization techniques) have become essential tools for engineering, science, business, and statistics. These methods are relatively the latest and popular because of the particular properties the deterministic algorithm does not have [29, 30]. For instance, stochastic methods always include probability, such as the random distribution of rainfall and water usage, in a reservoir, predicting the water level periodically, or forecasting the number of dropped connections for a communications network based on randomly variable but appropriate constant bandwidth. On the contrary, deterministic methods include probability under no circumstances; and outcomes based on exact input values [31].

Stochastic optimization is the process based on minimizing or maximizing the value of a mathematical or statistical function when one or more than one input parameter is subject to random variables. The randomness may be either as Monte Carlo randomness in the search procedure, noise in measurements, or both [29, 30].

Many industrial, economic, biological, and engineering problems can be confirmed as stochastic systems, such as banking, signal processing, geography, aerospace, communication area. In these systems, stochastic optimization is appropriate for solving decision-making problems, and many researchers have considered stochastic optimization methods in solving these problems. For instance, Yan et al. [32] suggested a qualitative and quantitative combined modeling specification depending on a hierarchical model structure framework which consists of the meta-meta model, the meta-model, and the high-level model. The results of the study showed that the proposed method could comprehensively describe the complex system. Li and Zhang [33] studied the problem of uncertain stochastic linear-quadratic optimal control under the inequality constraints for the final state. In this study, they proved the Karush-Kuhn-Tucker (KTT) theorem with hybrid constraints, and then they obtained new types of Riccati equations. This equation provides the necessary conditions for an optimal linear state feedback control existence produced by the KKT theorem. The design of a dynamic programming algorithm was achieved to solve the uncertain constrained stochastic linear quadratic subject. Aydın et al. [34] studied the design of the dimensionally stable laminated composites using the efficient global optimization method (EGO). The optimization problem of a composite plate under high stiffness and low thermal and moisture expansion coefficients was solved. The proposed optimization algorithm in this study is experimentally verified. After the design and optimization processes were completed, failure analysis of the optimized composites was performed using Tsai–Hill, Hoffman, Tsai–Wu, and Hashin–Rotem criteria. Generic steps of stochastic optimizations for renewable energy applications were extensively examined by Zakaria et al. [35]. Furthermore, the positive and negative sides of stochastic optimization were emphasized. Significant optimization methods belonging to the stochastic optimization stages are emphasized.

In their study, Niamsup and Rajchakit [36] introduced the latest improvements and significant stochastic optimization methods. It is claimed that the stochastic optimization methods are always more efficient than the deterministic optimization techniques for social, economic, technical aspects of renewable energy systems. Niamsup and Rajchakit examined polytopic discrete-time stochastic functions in the interval time-varying delays using the parameter-dependent Lyapunov-Krasovskii functional combined with linear matrix inequality techniques and new criteria for the robust stability of the stochastic system were proposed.

Maggioni et al. [37] studied the problem encountered by a bike-sharing service provider who needed to manage a fleet of bikes over a set of bike stations, each with given capacity and time-varying stochastic demand. Next, we consider one-track bike-sharing systems with transshipment, multi-stage, and two-stage stochastic optimization models are suggested to determine the optimal number of bikes to appoint to every station at the beginning of the service. Finally, managerial insights are provided comparing the solution supplied in the real system with the solutions obtained using the two-stage and the multi-stage models.

Gutierrez et al. [38] studied how to cope with the problem of the indefinite case in the optimal management of the hydrogen network of a petroleum refinery. A two-stage stochastic optimization method was used to analyze the effect of raw changes in the operation of the network. In addition, they were analyzed how to solve the hydrogen network problem to obtain feasible solutions with stochastic and deterministic solutions by using real plant data.

Khayyam et al. [39] proposed a stochastic optimization model for carbon fiber production in the carbonization process to reduce energy consumption in a proper range of fundamental mechanical properties. They developed processing operations, and fifty samples of fiber were analyzed for each set of operations, tensile strength, and modulus. During the production of the samples, the energy consumption was monitored on the processing equipment, and the five distribution functions were examined to determine distribution functions that could best describe the mechanical properties distribution of filaments. In addition, the Kolmogorov–Smirnov test was performed in order to confirm the distribution goodness of fit and correlation statistics. The study showed that the production quality could be predicted using the stochastic optimization models in the given range, and this method minimized the energy consumption of its industrial process.

Tifkitsis et al. [40] developed a stochastic multi-objective cure optimization methodology and performed it on the thick epoxy/carbon fiber laminates. The kriging method, which substitutes the Finite Element (FE) simulation, was used to construct a surrogate model for computational efficiency. The surrogate model and Monte Carlo were coupled and integrated into a stochastic multi-objective optimization framework depending on Genetic Algorithms. The results indicated a significant reduction of 40% in the temperature overshoot and cured time compared to standard cure profiles.

Simulated Annealing (SA), Differential Evolution (DE), Nelder–Mead (NM), and Random Search (RS) are examples of stochastic optimization methods [29–32]. Researchers continue to improve and add new stochastic methods or both to the literature. In the following subsections, some commonly used stochastic optimization methods are briefly overviewed.

3.2 Simulated Annealing

The simulation annealing (SA) method, one of the most effective and general optimization algorithms of stochastic algorithms, is quite useful for finding the global minimum of a function of a considerable number of independent variables. Besides, the SA method includes the analogy between finding the minimum function value in mixed-integer, discrete, or continuous minimization problems and the physical annealing process. In condensed matter physics, the physical annealing process is known as a thermal process to obtain the low energy states of a solid in a heat bath. The fundamental notion of the SA algorithm is to use random search in terms of a Markov chain, which not only accepts changes that develop the objective function but also keeps some of the non-ideal changes.

At each iteration in the SA algorithm, a new point is randomly generated, and when any stopping criteria are satisfied, the algorithm ends (Figure 3.1). The distance between the new and current point or the extent of the search is based on Boltzmann probability distribution with a scale in proportion to the temperature. Boltzmann Probability Distribution [30, 41–43] is defined in Equation 3.1 as

$$P(E) = e^{-E/kT} \tag{3.1}$$

where,

$P(E)$: The Probability of Achieving the Energy Level (E),

k: The Constant of Boltzmann,

T: Temperature.

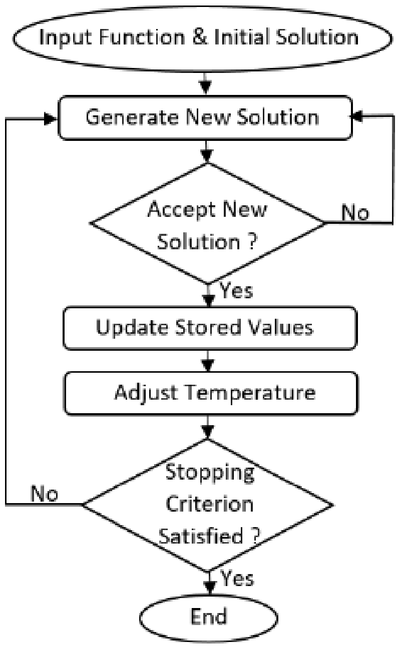


Figure 3.1: Flowchart of the simulated annealing algorithm [41]

3.3 Differential Evolution

Differential Evolution (DE) is a search technique that was first introduced by Storn and Price in 1996 for optimization problems over continuous domains. DE is one of the most powerful real-parameter optimization algorithms at present. This algorithm comprises four basic stages: selection, crossover, mutation, and initialization. In addition, there exist three real control parameters in this algorithm: (i) differentiation/mutation constant, (ii) crossover constant, and (iii) population size. The differential evolution performance relies on the manipulation of target and difference vector to obtain a trial vector. The other control parameters in the DE algorithm are (i) problem dimension that scales the difficulty of the optimization case, and (ii) the maximum number of generations known as a stopping condition, and (iii) boundary constraints [30, 43]. A flowchart summarizing the process of the DE algorithm is shown in Figure 3.2.

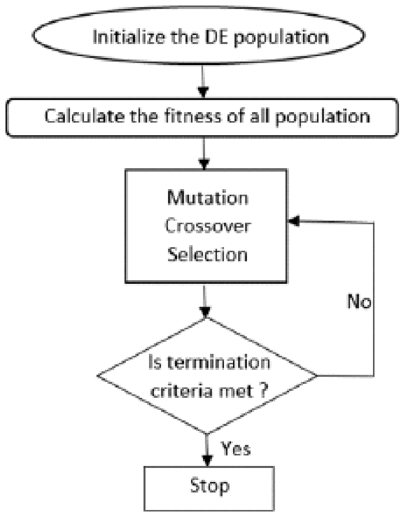


Figure 3.2: Flowchart of the differential evolution algorithm [44]

The differential evolution algorithm is a population-based algorithm like GA that uses similar operators. The primary difference between these algorithms is that GA relies on a crossover, a mechanism of useful and probabilistic exchange of information among solutions to find better solutions. However, DE relies on mutation operation as the primary search mechanism. This main operation is based on the differences among randomly sampled pairs of solutions in the population. Even though this method is numerically uneconomical, DE is strong and efficient enough to find an optimum global value and prevent the local minimum irrespective of initial points [45–47].

3.4 Nelder–Mead

In the classical optimization literature, the Nelder–Mead (NM) algorithm is also known as Simplex Search. It's a traditional local search method that has been firstly designed for unconstrained optimization problems [48]. It is important to note that, although NM is not a global optimization algorithm, it tends to work fairly well in practice for problems with many local minima. The adjustment of the NM options, like that of the DE algorithm, is controlled by four basic parameters: reflection, expansion, contraction, and shrinkage. The main feature of the NM algorithm is that the first few iterations produce satisfactory results.

Furthermore, it is required that one or two function evaluations only are notably rare in practice for each iteration. To avoid expensive or time-consuming multiple function evaluations, the Simplex can change its orientation, size, and shape to adapt to the local contour of the objective function. Moreover, NM has much flexibility in exploring complex search spaces. The algorithm's main steps are given in Figure 3.3.

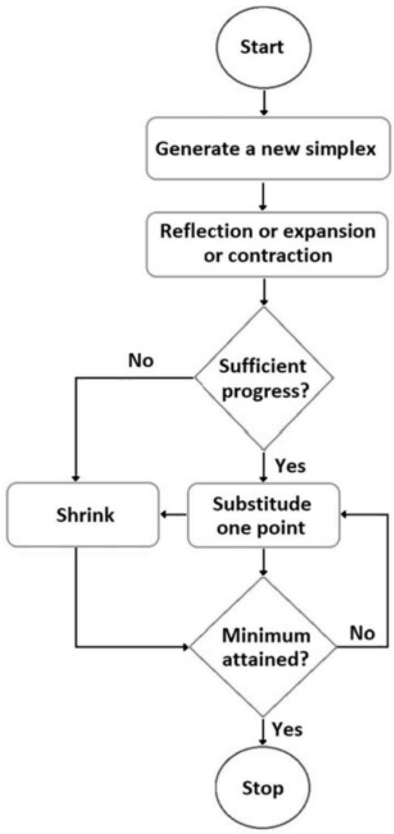


Figure 3.3: Flowchart of the nelder–mead algorithm [49].

3.5 Random Search

The Random Search (RS) algorithm, also known as the Monte Carlo Method, is based on a stochastic approach. Because of the algorithm's stochastic nature, it differs from deterministic methods like Branch-Bound and Interval Analysis. The main advantages of RS are that (i) when the absolute maximum of a multimodal function is necessary, it should be easily integrated with some form of the true search procedure, (ii) it allows non-convex, non-differentiable objective functions with continuous and/or discrete domains to approach the global optimum, (iii) it is simple to apply to even the most complex optimization tasks, (iv) for ill-structured global optimization problems, the RS algorithm is relatively stable and provides fundamental information quickly. The considered RS method utilized in this paper follows the procedure shown in Figure 3.4. In addition, a detailed discussion of the Random Search method can be found in [50, 51].

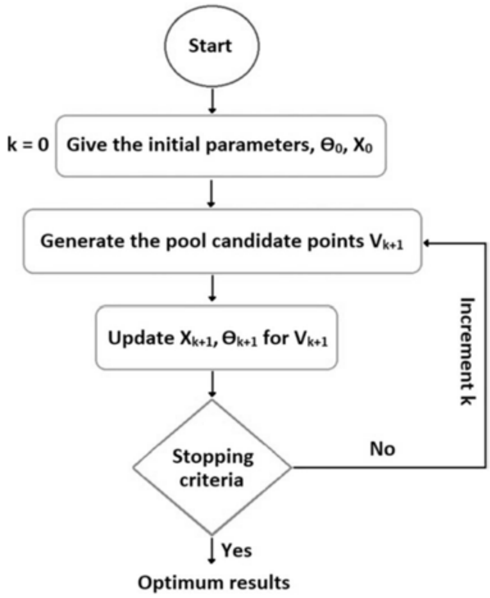


Figure 3.4: Flowchart of the random search algorithm [51].

3.6 Conclusion

This chapter shows the inspection of the seven most preferred stochastic optimization methods in different industrial areas such as engineering, biomedical, textile, construction, and automotive. These include Simulated Annealing (SA), Differential Evolution (DE), Nelder–Mead (NM), and Random Search (RS). A brief description of each method is presented, along with their flowcharts. All the stochastic methods discussed can be used for convenience to solve several problems, and comparable results with high accuracy can be obtained. This chapter gives the general mathematical foundations and algorithmic frameworks of these four methods and gives us insights into how these optimization algorithms can be effectively used in such a wide range of application areas.

CHAPTER 4

Mathematica and Optimization

4.1 Global and Local Optimization by Mathematica

The Mathematica software has a collection of commands which make exact-numeric optimization to solve linear-nonlinear and unconstrained-constrained problems. In this respect, *NMinimize* and *NMaximize* commands are used in numeric global optimization methods, while *Minimize* and *Maximize* commands are only appropriate for exact global optimization. Numeric local optimization is carried out by using the *FindMinimum* command. The abovementioned commands could be utilized for linear-nonlinear and constrained-unconstrained optimization problems [52]. Detailed explanations about commands, algorithms and which types of problems are used to solve are given in Table 4.1 and Figure 4.1.

Numerical global optimization algorithms for constrained nonlinear problems can be classified into *Gradient-Based* and *Direct Search* methods. *Gradient-Based* methods use first or second derivatives of objective function and constraints for calculation, while *Direct Search* methods have a probabilistic process and do not need derivative information.

In this chapter, the Mathematica commands (*FindMinimum, NMaximize,* and *Nminimize, RandomSearch, SimulatedAnnealing, NelderMead, DifferentialEvolution*) are explained, and the capability of the algorithms are evaluated for finding the global minimum for distinct test functions.

Table 4.1: Optimization methods and commands [52]

Optimization Types	Optimization Methods/Algorithms	Mathematica Commands
• Numerical Local Optimization	• Linear Programming Methods • Nonlinear Interior Point Algorithms	• *FindMinimum* • *FindMaximum*
• Numerical Global Optimization	• Linear Programming Methods • Differential Evolution • Nelder-Mead • Simulated Annealing • Random Search	• *NMinimize* • *NMaximize*
• Exact Global Optimization	• Linear Programming Methods, • Cylindrical Algebraic Decomposition • Lagrange Multipliers • Integer Linear Programming	• *Minimize* • *Maximize*
• Linear Optimization	• Linear Programming Methods (simplex, revised simplex, interior point)	• *LinearProgramming*

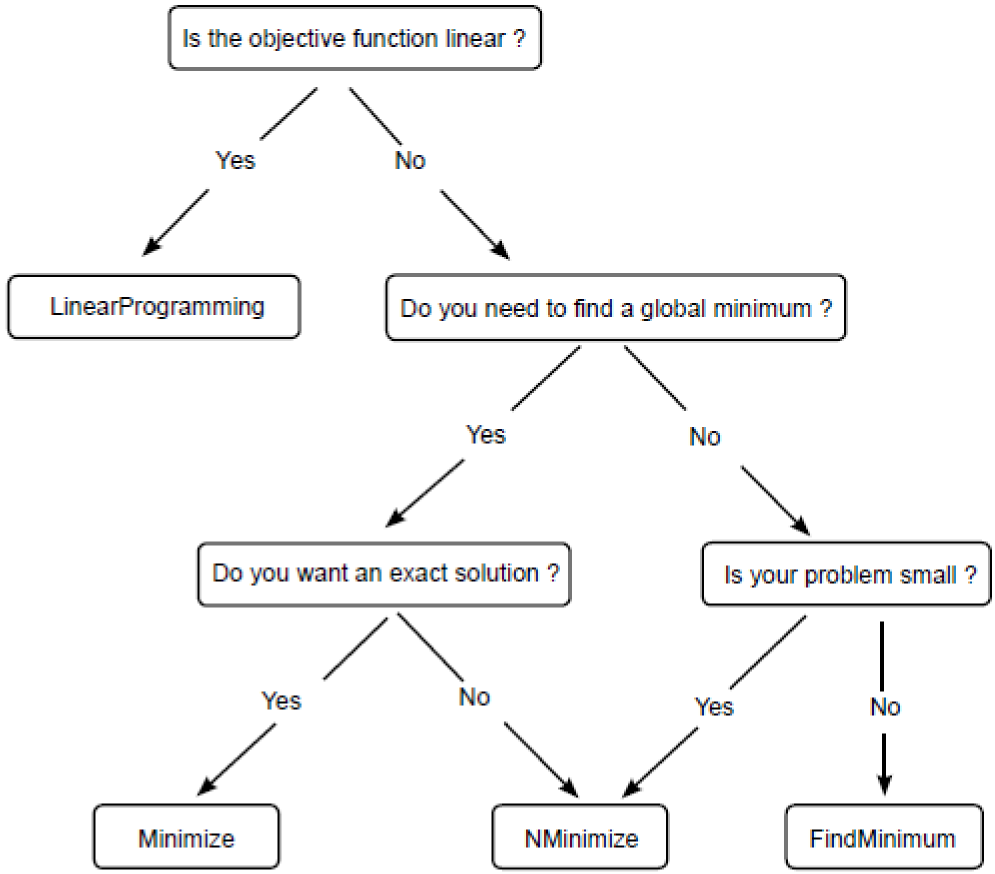


Figure 4.1: Mathematica optimization process [52]

4.1.1 *RandomSearch* Solver

The Random Search (RS) algorithm implemented by Mathematica has a stochastic approach. In the working process, the algorithm composes population, including random starting points, and then the algorithm evaluates the convergence behavior of the starting points to the local minimum utilizing the *FindMinimum* local search method. During this process, the options: (i) *SearchPoints* determines the number of starting points as per "min(10 f,100)" expression, where f is the number of variables, (ii) *RandomSeed* adjusts the starting value for random number producer, (iii) *Method* is defined by which method to use for minimizing the objective function by *FindMinimum*. Here, for unconstrained optimization problems, the *FindMinimum* command uses *Quasi-Newton* as a search method which does not need the second derivatives (Hessians matrix) to be computed; instead, the Hessian is updated by analyzing successive gradient vectors. In the case of the constrained optimization problem, the nonlinear interior point is selected as a search method by the *FindMinimum* command, (iv) *PostProcess* option can be selected as Karush–Kuhn–Tucker (KKT) conditions or *FindMinimum*. At the end of these processes, the best local minimum is selected to be the solution.

Mathematica automatically controls the options *InitialPoints*, *Method*, *PenaltyFunction*, *PostProcess*, and *SearchPoints* used in the Random Search algorithm, and appropriate values of options are selected according to optimization problems [52]. The RS algorithm follows the procedure given in Figure 4.2.

In order to evaluate the performance capacity of the Random Search algorithm in finding the global minimum, separable and non-separable multimodal test functions having more than one, few or many local minima are used. However, this kind of global optimization problem is quite hard when an algorithm is not designed appropriately, and it can be inserted into the local minima without finding the global minimums or not all global minimums. In this respect, the first selected test function, which has global minima, is located at $f(0,0) = 0$ is the Ackley [54]. The following commands give the Mathematica syntax for the definition of Ackley function and its 3D plot in an interval as seen Figure 4.3.

```
In[1]:=  f[x1_,x2_]:=-20Exp[(-0.02Sqrt[0.5(x1^2+x2^2)])]
         -Exp[(0.5(Cos[2Pix1]+Cos[2Pix2]))]+20+Exp[1];
In[2]=   Plot3D[f[x1,x2],{x1,-35,35},{x2,-35,35},
         AxesLabel->{x1,x2,y}]
```

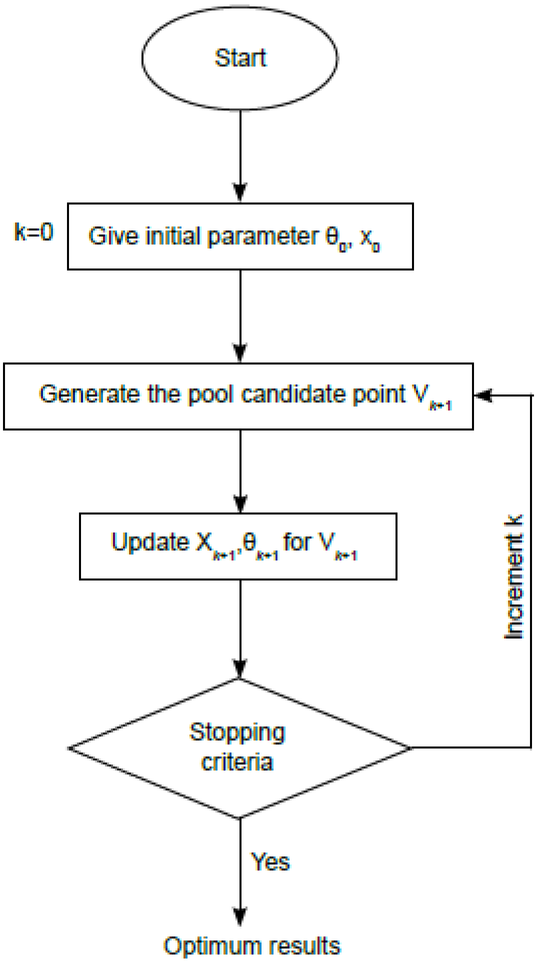


Figure 4.2: Flowchart of the random search algorithm [53]

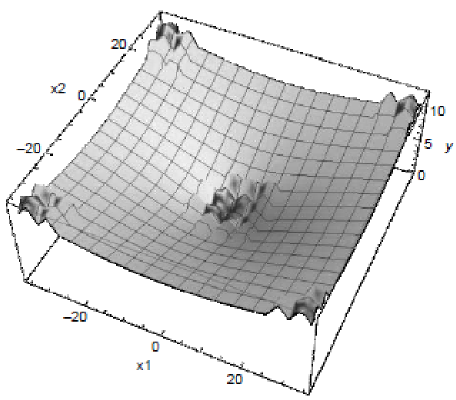


Figure 4.3: Ackley function's 3D plot in an interval.

It is noted that the *RandomSearch* command may not find a global minimum without working any alteration of its options.

```
In[3]:=   NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->"RandomSearch"]
Out[3]=   {2.83635, {x1->-5.99749,x2->8.99623}}
```

Sometimes changing the search point option that specifies the number of points to start searches can be effective in finding a global minimum.

```
In[4]:=   Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"RandomSearch","SearchPoints"->i}]],
          {i,500,3000,500}]
          {0.39531,{x1->0.996345,x2->0.996345}}
          {0.280127,{x1->-5.04225*10^-24,x2->-0.9948}}
          {0.280127,{x1->-5.04225*10^-24,x2->-0.9948}}
          {0.280127,{x1->-5.04225*10^-24,x2->-0.9948}}
          {0.280127,{x1->-5.04225*10^-24,x2->-0.9948}}
          {1.2012*10^-9,{x1->-8.42728*10^-
          10,x2->-4.16243*10^-9}}
```

The effect of the *RandomSeed* option, which constitutes starting value for the random number generator, can be investigated below. In the previous example, while the "*Searchpoints*"->500 is not sufficient to reach the global minimum, in the following example, a global minimum can be obtained by setting the values of the *SearchPoints* and the *RandomSeed* to 500 and 5, respectively.

```
In[5]:=   Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"RandomSearch","SearchPoints"->500,
          "RandomSeed"->i}]],{i,5}]
          {0.280127,{x1->-7.38323*10^-25,x2->0.9948}}
          {7.40815*10^-10,{x1->6.89861*10^-10,
          x2->-2.52669*10^-9}}
          {0.280127,{x1->5.59478*10^-24,x2->0.9948}}
          {0.39531,{x1->0.996345,x2->0.996345}}
          {1.37499*10^-9,{x1->-3.64123*10^-9,
          x2->-3.22083*10^-9}}
```

Here, points are produced on a grid to utilize as initial points. If the approximate solution range of the problem can be estimated, assigning the starting point makes it easier to get the solution.

```
In[6]:=   Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"RandomSearch","InitialPoints"-
          >Flatten[Table[{i,j},{i,-35,35,5},{j,-35,35,5}],1]}]]
Out[6]=   {-4.44089*10^-16,{x1->-1.52703*10^-15,
          x2->-1.52703*10^-15}}
```

PostProcess option is not of primary importance for this problem. *PostProcess* methods KKT and *FindMinimum* give the same results.

```
In[7]:=   Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"RandomSearch", "SearchPoints"->3000,
          "PostProcess"->KKT}]]
Out[7]=   {1.2012*10^-9,{x1->-8.42726*10^-10,
          x2->-4.16243*10^-9}}
In[8]:=   Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"RandomSearch", "SearchPoints"->3000,
          "PostProcess"->FindMinimum}]]
Out[8]=   {1.2012*10^-9,{x1->-8.42728*10^-10,
          x2->-4.16243*10^-9}}
```

Another test function, Holder Table 1, which is separable and multimodal, is used to evaluate the capability of the *RandomSearch* command in finding the global minimum [54]. This test function has global minima located at f (±9.646168, ±9.6461680) = –26.920336. The followings are Mathematica syntax for the definition of the "Holder Table 1" function and its 3D plot given in Figure 4.4.

```
Clear[f];
In[9]:= f[x1_,x2_]:=-Abs[Cos[x1]Cos[x2]Exp
[Abs[1-((x1^2+x2^2)^0.5)/Pi]]];
In[10]:= Plot3D[f[x1,x2],{x1,-10,10},{x2,-10,10}]
```

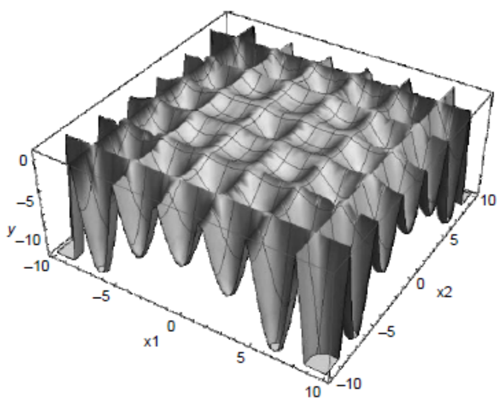


Figure 4.4: "Holder Table 1" function's 3D plot.

The RS algorithm finds one of the global minimum points without altering its options for this problem.

```
In[11]:=  NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},
          {x1,x2},Method->"RandomSearch"]
Out[11]=  {-26.9203, {x1 -> -9.64617, x2 -> -9.64617}}
In[12]:=  Do[Print[NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},
          {x1,x2},Method->{"RandomSearch", "RandomSeed"->i}]],
          {i,{1,6,7}}]
          {-26.9203,{x1->-9.64617,x2->9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->9.64617,x2->9.64617}}
```

4.1.2 *SimulatedAnnealing* Solver

The Simulated Annealing (SA) algorithm implemented by Mathematica is a stochastic approach having a working process based on the physical annealing procedure of solids. The SA is designed to find the largest or smallest values of functions having many variables and the smallest values of nonlinear functions having many local minimums. The algorithm is named Simulated Annealing because it exemplifies the perfect arrangement of atoms of solid bodies and minimizes the potential energy during the cooling process. Furthermore, the algorithm allows the structure to move away from the local minimum and investigate and find a better global minimum [55].

In the working process for each iteration, firstly, the startup solution "Z" is produced, Secondly, "Z_{New}" is generated in the neighborhood of the current point, "Z" and then Z_{Best}" is defined.

If $f(Z_{New}) \leq f(Z_{Best})$, Z_{New} replaces Z_{New} and Z. Otherwise, Z_{New} replaces with Z. In this loop, options *InitialPoints*, *SearchPoints*, and *RandomSeed* are capable of determining the initial guess and its number and starting value, respectively. The SA algorithm performs random movements in the search space based on the Boltzmann probability distribution $e^{D(k,\Delta f,f_0)}$. Here D is the function defined by option *BoltzmannExponent*, k is the current iteration, Δf is the change in the objective function. In the Mathematica, if the user does not select manually, B is defined as $\frac{-\Delta f \log(k+1)}{10}$ *BoltzmannExponent*.

For all starting points, the working process introduced above is returned by the time either the algorithm converges to a point, or the algorithm remains at the same point due to the number of iterations assigned by the option *LevelIterations* [56]. The SA algorithm follows the procedure given in Figure 4.5.

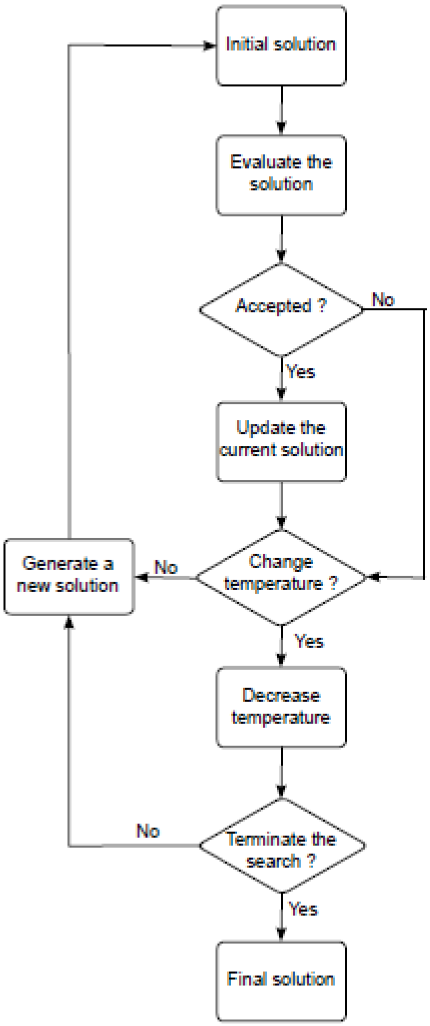


Figure 4.5: Flowchart of the simulated annealing algorithm [57]

"Ackley" and "Holder Table 1" are used to evaluate the performance capacity of the *SimulatedAnnealing* command to find the global minimum.

```
In[1]:=   f[x1_,x2_]:=-20 Exp[(-0.02 Sqrt[0.5 (x1^2+x2^2)])]-
          Exp[(0.5 (Cos[2 Pi x1]+Cos[2 Pi x2]))]+20+Exp[1];
In[2]=    Plot3D[f[x1,x2],{x1,-35,35},{x2,-35,35},
          AxesLabel->{x1,x2,y}]
```

The SA algorithm may not find a global minimum by using the default value of its options.

```
In[3]:=   NMinimize[{f[x1,x2], -35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{“SimulatedAnnealing”}]
Out[3]=   {2.37578, {x1 -> 7.99584, x2 -> 3.99792}}
```

BoltzmannExponent includes a function that determines a new point at each iteration; thus, the *BoltzmannExponent* is a significant option that shows how to achieve a global minimum. If this function is utilized without a default value, the obtained result can be changed. However, in the following problem, changing this option alone has not been enough to find the global minimum.

```
In[4]:=   NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},{x1,x2},
          Method->{“SimulatedAnnealing”, “BoltzmannExponent”
          ->Function[{i,df,f0},-df/(Exp[i/10])]}]
Out[4]=   {0.830095, {x1 -> -2.99495, x2 -> 6.41153*10^-9}}
```

For this problem, although the *PerturbationScale* alters the result, changing this option alone has not been enough to find the global minimum. The algorithm attaints to local minimum points.

```
In[5]:=   Do[Print[NMinimize[{f[x1, x2], -35 ≤ x1 ≤ 35, -35 ≤ x2
          ≤ 35}, {x1, x2}, Method -> {“SimulatedAnnealing”,
          “PerturbationScale” -> i}]], {i, 15}]
          {2.37578,{x1->7.99584,x2->3.99792}}
          {2.40345,{x1->0.999488,x2->8.99539}}
          {1.0993,{x1->-1.04986*10^-9,x2->3.99502}}
          {3.8527,{x1->-1.99944,x2->14.9958}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {4.50046,{x1->14.9966,x2->-9.99773}}
          {4.26698,{x1->11.9971,x2->-11.9971}}
          {4.27353,{x1->7.99805,x2->-14.9963}}
          {2.63697,{x1->5.99725,x2->-7.99634}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
          {6.15308,{x1->-23.9966,x2->-9.9986}}
```

Using many more *SearchPoints*, a global minimum can be obtained.

```
In[6]:=   Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method-> {“SimulatedAnnealing”,
          “SearchPoints”->i}]],{i,100,500,100}]
          {0.830095,{x1->-2.99495,x2->7.32049*10^-10}}
          {0.62186,{x1->1.99543,x2->-0.997715}}
          {0.280127,{x1->-1.64485*10^-9,x2->-0.9948}}
          {0.280127,{x1->0.9948,x2->5.25186*10^-12}}
          {1.937*10^-9,{x1->-2.31279*10^-9,x2->-6.44598*10^-9}}
```

As previously seen, while changing the search points alone is sufficient to find the global minimum, in the case of conducting a search using the *RandomSeed*, *PerturbationScale*, and *BoltzmannExponent*, the algorithm seizes the local minimums.

```
In[7]:=  Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,
         -35≤x2≤35},{x1,x2},Method->
         {"SimulatedAnnealing","RandomSeed"->i}]],{i,0,10}]
         {2.37578,{x1->7.99584,x2->3.99792}}
         {0.557056,{x1->-4.99634*10^-9,x2->1.99487}}
         {2.15456,{x1->7.99533,x2->-0.999416}}
         {0.39531,{x1->0.996345,x2->0.996345}}
         {3.46466,{x1->-8.99708,x2->9.99676}}
         {0.993567,{x1->2.99583,x2->1.99722}}
         {1.58244,{x1->-2.9975,x2->-4.99584}}
         {1.22508,{x1->-3.99557,x2->1.99779}}
         {1.46596,{x1->1.99819,x2->-4.99546}}
         {0.39531,{x1->-0.996345,x2->0.996345}}
         {2.29034,{x1->4.99729,x2->6.9962}}
```

```
In[8]:=  Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
         {x1,x2},Method-> {"SimulatedAnnealing",
         "PerturbationScale"->3,"SearchPoints"->500,
         "RandomSeed"->i}]], {i, 0, 10, 1}]
         {-4.44089*10^-16,{x1->-1.62365*10^-15,
         x2->2.19073*10^-16}}
         {0.39531,{x1->-0.996345,x2->0.996345}}
         {0.557056,{x1->1.99487,x2->-1.44602*10^-11}}
         {1.16405,{x1->-2.99649,x2->-2.99649}}
         {0.557056,{x1->-1.99487,x2->-6.2523*10^-11}}
         {0.62186,{x1->-1.99543,x2->-0.997715}}
         {0.557056,{x1->1.99487,x2->7.97744*10^-12}}
         {0.39531,{x1->0.996345,x2->0.996345}}
         {0.280127,{x1->0.9948,x2->-6.58993*10^-9}}
         {2.09443*10^-9,{x1->-7.39447*10^-9,
         x2->-3.93044*10^-10}}
         {0.993567,{x1->-1.99722,x2->-2.99583}}
```

The Simulated Annealing algorithm finds one of the global minimum points without altering its options for this problem.

```
In[11]:=  NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},{x1,x2},
          Method->"SimulatedAnnealing"]
Out[11]=  {-26.9203, {x1 -> 9.64617, x2 -> 9.64617}}
```

Unlike the Random Search algorithm, four distinct global minimum points can be found using the Simulated Annealing algorithm.

```
In[12]:=  Do[Print[NMinimize[{f[x1,x2],-10≤x1≤10,
          -10≤x2≤10},{x1,x2},Method->
          {"SimulatedAnnealing","RandomSeed"->i}]],
          {i,{1,2,3,11}}]
          {-26.9203,{x1->9.64617,x2->9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->-9.64617,x2->9.64617}}
          {-26.9203,{x1->9.64617,x2->-9.64617}}
```

4.1.3 *NelderMead* Solver

Nelder–Mead algorithm (NM) or Simplex is one of the derivative-free optimization methods among other traditional local search algorithms. It was firstly designed for unconstrained optimization problems [58]. In m-dimensional space and given a function of m variables, this method keeps a set of $m + 1$ points generating the vertices of a polytope. It should be noted that it must not be confused with the simplex method for linear programming. Iterations have been performed by forming $m + 1$ points as y_1, y_2, y_3,…, y_{m+1}. These points form the functions are ordered as $h(y_1) \leq h(y_2) \leq h(y_3) \leq \ldots h(y_{m+1})$. After the new point is produced to change with the previous worst point y_{m+1}. A polytope can be defined in its centroid $c = \sum_{i=1}^{m} y_i$, being the average position of all the points of an object. Here, a trial point should be defined (y_t). It is produced by reflecting the worst point until centroid, $y_t = c + \propto (c - y_{m+1})$, where $\propto$ is a parameter being larger than 0. In this part, the new point need not be a new worst point or a new best point. Hence, $h(y_1) \leq h(y_t) \leq h(y_m)$, y_t replace with y_{m+1}. After obtaining a new point being better than the previous best point, reflection is successfully obtained.

Further, it can be continued with $y_e = c + \beta(y_t - r)$ where β being larger than 1 is a parameter to largen polytope. If $h(y_e)$ is obtained as smaller than $h(y_t)$, the expansion process is achieved. Therefore, ye changes with y_{m+1}. Alternatively, else, y_t changes as y_{m+1}. Another certain step for the algorithm process is that if the new point yt underperforms to the second-worst point, $h(y_m) \leq h(y_m)$, the polytope is thought as very large and it is required to be contracted. Hence, a new trial point is obtained using the following expressions [59].

$$y_c = \begin{pmatrix} c + \gamma\,(y_{m+1} - c), & if\, h(y_t) \geq h(y_{m+1}) \\ c + \gamma(y_t - c), & if\, h(y_t) < h(y_{m+1}) \end{pmatrix} \tag{4.1}$$

where γ is a parameter ranging between 0 to 1. If contraction is achieved, it means that $h(y_c)$ is smaller than Min[$h(y_{m+1})$, $h(y_t)$]. Reversely, more process is required to obtain strong contraction.

Nelder–Mead has specific flexible options similar to other algorithms, which are *ContractRatio*, *ExpandRatio*, *InitialPoints*, *PenaltyFunction*, *PostProcess*, *RandomSeed*, *ReflectRatio*, *ShrinkRatio*, and *Tolerance*. Even though this algorithm does not provide complete specifications that an accurate global optimization method should require, it tends to work well for the problem of having fewer local minima. As with previous algorithms, Nelder–Mead is used to obtaining optimum global values for Ackley and Holder Table 1 test functions [52].

```
In[5]:=  NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},{x1,x2},
         Method->"NelderMead"]
Out[5]=  {0.87404, {x1 -> -0.998405, x2 -> -2.99522}}
```

It can be seen that the results of the first trial are outperformed by DE while it gives better global optima compared to *Random Search* and *Simulated Annealing* solutions for Ackley function with the default setting.

RandomSeed, which is referred to as one of the critical adjustment parameters of NM, might directly affect the performance of the NM finding global minima.

```
In[6]:=  Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
         {x1,x2},Method->{"NelderMead",
         "RandomSeed"->i}]],{i,5}]
Out[6]=  {0.557056,{x1->8.15872*10^-25,x2->-1.99487}}
         {0.280127,{x1->0.9948,x2->-6.32493*10^-9}}
         {7.12481,{x1->-20.9977,x2->-22.9975}}
         {2.32486*10^-10,{x1->4.63269*10^-10,
         x2->-6.78982*10^-10}}
         {1.3908,{x1->-4.99519,x2->-0.999038}}
```

Adjusting *RandomSeed* parameters provided a better minimum value of 2.32486*10–10, than the trial performed with the default setting.

In this algorithm, other useful adjustment parameters are *ShrinkRatio*, *ContractRatio*, and *ReflectRatio*. However, it did not obtain global minima in Ackley function, as indicated in the following.

```
In[7]:=   Do[Print[NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},
          {x1,x2},Method->{"NelderMead", "ShrinkRatio"->0.95,
          "ContractRatio"->0.95,"ReflectRatio"->2,
          "RandomSeed"->i}]],{i,5}]
Out[7]=   {0.39531,{x1->-0.996345,x2->-0.996345}}
          {0.783523,{x1->-1.99642,x2->1.99642}}
          {7.37952,{x1->-5.99939,x2->-31.9967}}
          {0.39531,{x1->-0.996345,x2->0.996345}}
          {2.40704*10^-9,{x1->-2.92841*10^-9,
          x2->-7.99045*10^-9}}
```

Another test function Holder Table 1 was minimized using the *NMinimize* command. As seen below, the global minima with default values were –26.9203.

```
In[12]:=  NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},{x1,x2},-
          Method->"NelderMead"]
Out[12]=  {-26.9203, {x1 -> 9.64617, x2 -> 9.64617}}
```

As applied to the previous test function, *RandomSeed* has been adjusted to find global minima.

```
In[13]:=  Do[Print[NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},
          {x1,x2},Method->{"NelderMead",
          "RandomSeed"->i}]],{i,5}]
Out[13]=  {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-9.13635,{x1->3.24199,x2->-9.71802}}
          {-26.9203,{x1->9.64617,x2->-9.64617}}
          {-7.76664,{x1->2.08542*10^-8,x2->9.73295}}
          {-7.76664,{x1->-7.64705*10^-9,x2->-9.73295}}
```

Outputs of this trial showed that adjustment of *RandomSeed* as is was not sufficient to reach minimum value. Lastly, other possibly useful parameters concerning literature for Nelder–Mead were adjusted to obtain global minima.

```
In[14]:=  Do[Print[NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},
          {x1,x2},Method->{"NelderMead", "ShrinkRatio"->0.95,
          "ContractRatio"->0.95,"ReflectRatio"->2,
          "RandomSeed"->i}]],{i,5}]
Out[14]=  {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
          {-26.9203,{x1->-9.64617,x2->-9.64617}}
```

In this example, it was seen that none of the parameters could assure that a global minimum is different from the value obtained using the default settings.

4.1.4 *DifferentialEvolution* Solver

Differential Evolution (DE) is one of the most common stochastic search algorithms in the optimization and solution of complicated and challenging design problems. As mentioned in the previous chapter of this study (Chapter 3), the algorithm is built on four main steps: initialization, mutation, crossover, and selection. Although DE is an efficient search algorithm thanks to covering a population of solutions in iterations rather than a single solution, it computationally requires more process time, making it an expensive method. DE is a robust and reliable algorithm to obtain global optimum. However, there is uncertainty in finding global optimum points, as also valid for other types of search methods [60].

In iterations, a new population of k points is produced. Then, the j^{th} new point is produced by taking three random points such as z_1, z_2, and $z_3$from the previously generated population. Then it builds the new formation by z_s = z_3 + s($z_3 - z_2$) that s is the real scaling factor. A new point z_{New} is generated from z_j and z_s by picking the i^{th} coordinate or another coordinate of j^{th} from z_s with the probability of p. Then, z_{New} changes with z_j, if the function of $h(z_{New})$ is smaller than the function of $h(z_j)$ [52]. The command *DifferentialEvolution* consists of specific adjustment options: *CrossProbability (P)*, *InitialPoints*, *PenaltyFunction*, *PostProcess*, *RandomSeed*, *ScalingFactor*, *SearchPoints*, and *Tolerance* whether none of them does guarantee to find global optima. The process flow chart of the algorithm is illustrated in Figure 4.6 [58]. As performed with previous search algorithms, the same test functions of Ackley and Holder Table 1 are used to evaluate the capacity of the DE algorithm to find the global minimum.

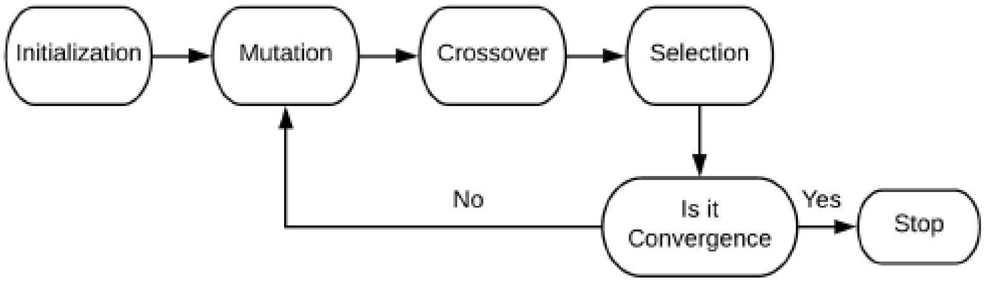


Figure 4.6: Flowchart of differential evolution algorithm [58]

```
In[3]:=   NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},{x1,x2},
          Method->"DifferentialEvolution"]
Out[3]=   {2.2587*10^-9, {x1 -> -1.63413*10^-9,
          x2 -> -7.81672*10^-9}}
```

Changing *ScalingFactor* from default value of 0.6 to 0.7 obtained better results considering global optima.

```
In[5]:=   NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35}, {x1,x2},
          Method->{"DifferentialEvolution",
          "ScalingFactor"-> 0.7}]
Out[5]=   {3.74914*10^-10, {x1 -> 8.68579*10^-10,
          x2 -> -1.00129*10^-9}}
```

Here, adjusting *ScalingFactor*, *RandomSead*, *CrossProbabilty*, or *SearchPoints* did not produce better global optima. Therefore, they were kept at default.

The global minima of other test functions Holder Table 1 were tried to find by the algorithm. Initial steps are the same with previously used algorithms.

```
In[10]:=   NMinimize[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},{x1,x2},-
           Method->"DifferentialEvolution"]
Out[10]=   {-26.9203, {[x1 -> 9.64617, x2 -> -9.64617}}
```

In this example of function, all of the parameters being different from their defualt values did not obtain different results or better global minima.

4.1.5 *FindMinimum* Solver

The *FindMinimum* command is used to find a local minimum function for unconstrained and constrained optimization problems [52].

The options of the *FindMinimum* command are *Method*, *MaxIterations*, *WorkingPrecision*, *PrecisionGoal*, and *AccuracyGoal*.

The Method option ventilates that the *FindMinimum* command selects which method solves problems. Hereof, unconstrained optimization problems; (i) Newton utilizes the exact *Hessian* or a finite difference approximation, (ii) *QuasiNewton* uses the quasi-Newton BFGS approximation, which was composed by updates based on past steps, (iii) the *LevenbergMarquardt* method, also known as the damped least-squares (DLS) method, is employed to solve nonlinear least-squares problems, (iv) the *ConjugateGradient* method is appropriate for solving linear systems, (v) the *PrincipalAxis* method does not need derivatives, and it requires two starting conditions in each variable. In constrained optimization problems, only *InteriorPoint* can be selected as a method.

The *MaxIterations* option indicates the maximum number of iterations that ought to be utilized. In constrained optimization problems, the default "*MaxIterations*"->500 is used.

WorkingPrecision, *PrecisionGoal*, and *AccuracyGoal* are options specifying the number of digits of precision. The former controls the internal computations while the latter checks the final result. By default, *WorkingPrecision->prec* is equal to *MachinePrecision*, but If *prec>MachinePrecision*, a constant *prec* value is used during the computation. When *AccuracyGoal* and *PrecisionGoal* options are selected as Automatic, the default values are set to *WorkingPrecision*/3 and Infinity, respectively [52].

Carrom table function, which is a non-separable and multimodal function and has many local minima, has been taken as a test function, and the *FindMinimum* command and the effect of its options in finding the local minima are investigated [54].

The following commands give the Mathematica syntax for the definition of $f(x_1,x_2)$ function and its 3D plot in an interval (see Figure 4.7).

```
In[1]:=  f[x1_,x2_]:=-(Cos[x1]Cos[x2]
         Exp[Abs[1-((x1^2+x2^2)^0.5)/Pi]])^2/30
In[2]:=  Plot3D[f[x1,x2],{x1,-10,10},{x2,-10,10},
         AxesLabel->{x1,x2,y}]
```

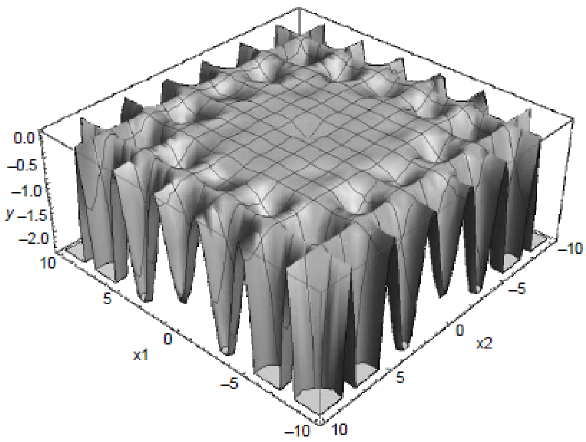


Figure 4.7: 3D plot of the function $f(x_1,x_2)$

```
In[3]:=  FindMinimum[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},{x1,x2}]
Out[3]=  {-24.1568,{x1->9.64617,x2->9.64617}}
In[4]:=  FindMinimum[{f[x1,x2],-10≤x1≤10,-10≤x2≤10},
         {x1,x2},Method->"InteriorPoint"]
Out[4]=  {-0.246302, {x1 -> -1.22418*10^-14,
         x2 -> -1.29143*10^-14}}
In[5]:=  Do[Print[FindMinimum[{f[x1,x2],-10≤x1≤10,
         -10≤x2≤10},{x1,x2},Method->"InteriorPoint",
         "MaxIterations"->i]],
         {i,{1,10,100,500,1000,2000,4000,8000}}]
         {-0.0105322,{x1->0.969586,x2->0.969586}}
         {-0.246302,{x1->-8.74067*10^-8,x2->-8.74067*10^-8}}
         {-0.246302,{x1->-8.37899*10^-15,x2->-8.38925*10^-15}}
         {-0.246302,{x1->-1.22418*10^-14,x2->-1.29143*10^-14}}
         {-0.246302,{x1->-1.22418*10^-14,x2->-1.29143*10^-14}}
         {-0.246302,{x1->-1.22418*10^-14,x2->-1.29143*10^-14}}
         {-0.246302,{x1->-1.22418*10^-14,x2->-1.29143*10^-14}}
         {-0.246302,{x1->-1.22418*10^-14,x2->-1.29143*10^-14}}

In[6]:=  Table[Print[FindMinimum[{f[x1,x2],-10≤x1≤10,
         -10≤x2≤10},{{x1,RandomReal[{-10,10}]},
         {x2,RandomReal[{-10,10}]}},Method->
         "InteriorPoint"]],{10}]
         {-0.0368271,{x1->0.000019185,x2->-3.44978}}
         {-1.42781,{x1->6.50458,x2->-6.50458}}
         {-6.7549,{x1->9.68366,x2->-6.45799}}
         {-0.272117,{x1->3.63079*10^-7,x2->-6.59135}}
         {-1.42781,{x1->6.50458,x2->-6.50458}}
         {-2.01069,{x1->-1.67999*10^-7,x2->9.73295}}
         {-1.42781,{x1->-6.50458,x2->-6.50458}}
         {-0.436543,{x1->6.56051,x2->3.28309}}
         {-0.0843916,{x1->-3.36299,x2->3.36298}}
         {-2.78243,{x1->-9.71802,x2->3.24199}}
```

4.1.6 *NMinimize* and *NMaximize* Solvers

Mathematica functions enable us to optimize complicated problems in engineering and science and their significant characteristic features using search algorithms. Although these methods are effective for locating global optima, optimum results may be difficult to find even in the absence of constraints and boundary conditions. The best way to cope with this situation might be to optimize given functions with different initial conditions. The following examples are obtained again using the initial test functions; Ackley function of $f(x_1, x_2)$ and Holder Table 1 function of $g(x_3, x_4)$, respectively.

```
In[15]:=   NMinimize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},{x1,x2}]
Out[15]=   {0.8740,{x1→-0.9984,x2→-2.9952}}
In[4]:=    NMaximize[{f[x1,x2],-35≤x1≤35,-35≤x2≤35},{x1,x2}]
Out[4]=    {12.3202,{x1→34.5137,x2→34.51377}}
In[7]:=    NMinimize[{g[x3,x4],-10≤x3≤10,-10≤x4≤10},{x3,x4}]
Out[7]=    {-26.9203,{x3→9.6461,x4→9.6461}}
In[8]:=    NMaximize[{g[x3,x4],-10≤x3≤10,-10≤x4≤10},{x3,x4}]
Out[8]=    {-2.5326×[10]^(-13),{x3→-4.7498,x4→-4.7123}}
```

According to initial results, global minima and maximal values of Ackley Function might be achieved. However, it is seen that it was not valid for Holder Table 1 function. Adjustment of the parameters or changing the restriction region might be effective towards obtaining global values.

Constraints might be either a rational combination of domain options, equalities and inequalities or list form. As an example, the unknown parameter z should be included as $z \in$ Integers in line if one needs to specify results in integer form. After that, this constraint restricts the probable solutions as being only integers. Also, the *NMinimize* command requires a rectangular starting region to begin optimization. That means each variable in the given function should have a finite upper and lower bound. Using the Method option enabling us to apply other types of search algorithms is a way to provide non-automatic set solutions, as seen in previous parts of this chapter performed using SA and RS algorithms. Here, it can be said that if the function is minimized or maximized (called an objective function) and constraints are linear, the *LinearProgramming* method is the default setting in the solving process. If the central part of the objective function is not numerical and also the variables are in integer form, DE is the algorithm as default. In other situations, NM is the search algorithm to be used. If Nelder–Mead does not provide desirable solutions, it switches with DE to obtain optimum values [52].

CHAPTER 5

Dual Mass Flywheel Optimization

5.1 Methods

5.1.1 *NDSolve* Solver

In Mathematica software, it is possible to find the solution of ODEs and PDEs numerically by the *NDSolve* command. Instead of writing a function, it gives an appropriate interpolation function, namely, *InterpolatingFunction*. Boundary values may also be specified using Dirichlet Condition (Boundary condition type for a partial differential equation which gives the prescribed value of the function on a surface) and Neumann Condition (boundary condition type for a partial differential equation which gives the first derivative on a surface). The command may solve some of the differential-algebraic equation types that either contain algebraic equations, differential equations, or both of them in one equation. The iteration process is valid for the *NDSolve* solution. In the first step of the iteration, a specific prescribed value is considered. Secondly, the output of each iteration is then the starting point of the next iteration again. Finally, this repetition process generates a sequence of output up to the endpoint. To define the maximum number of steps of the iteration process, we can use the "*MaxSteps*" option by selecting Automatic mode. In addition to this, *StartingStepSize*, *MaxStepSize*, and *NormFunction* items are used to describe (i) the size of the step at the beginning, (ii) the maximum size of the step in independent variable of the equation, and (iii) the norm of error estimation, respectively. If it is not given the maximum number of iteration for the process, Mathematica takes 10,000 as a stopping criterion. Given that tolerances are affected by error estimates. They can be scaled by combining the errors for different terms by satisfying the following condition.

$$f\left[\left\{\frac{error_1}{tolerance_r\,|\,x_1\,|+tolerance_a},\frac{error_2}{tolerance_r\,|\,x_2\,|+tolerance_a},\ldots,\frac{error_n}{tolerance_r\,|\,x_n\,|+tolerance_a}\right\}\right]\leq 1 \quad (5.1)$$

where the function f represents the norm function that computing norms of error estimate in *NDSolve* solver, $error_i$ is the i^{th} component of the $error$ and x_i is the i^{th} component of the current solution, n is the number of components. Absolute and relative tolerances are denoted by $tolerance_a$ and $tolerance_r$, respectively. In the procedure, an embedded error estimator tries to obtain an appropriate step size in an improved version of the explicit *Runge-Kutta* method, which is also adaptive embedded pairs of orders.

The options *TimeIntegration*, *BoundaryValues*, and *EquationSimplification* can be adjusted by the user for the *NDSolve* command. Depending on the DE type, these options correspond to systems of DE, Boundary Value Problems (BVPs) in ODE, and simplified equation, respectively. *Adams*, *BDF*, *ExplicitRungeKutta*, *ImplicitRungeKutta*, and *Symplectic-PartitionedRungeKutta* approaches are also hybridized with explicit *Runge-Kutta* method by time integration settings. The method starts with a trial step at the midpoint for the domain, and this leads to reducing lower-order error terms [61].

5.1.2 *FindFit* Solver

This solver is utilized to obtain the best-fitted function to the prescribed data numerically. "*ConjugateGradient*", "*Gradient*", "*LevenbergMarquardt*", "*Newton*", "*NMinimize*", and "*QuasiNewton*" are the method options that can also be selected according to the given problem. In the present problem, the *Levenberg-Marquardt* method is selected as an appropriate process to calculate the regression coefficients, which are also a sub-problem of the least-square approximation. It is a method for minimizing a sum-of-squares objective function. In this method, the following mathematical expression (Equation 5.2) is valid, and this corresponds to the variation of *Gauss-Newton* and *Gradient Descent* updates for the prescribed parameters.

$$[J^T W J + \lambda I]h = J^T W(y - \hat{y}) \tag{5.2}$$

In this equation, J^T, J, W are the transposed form of the Jacobian, traditional Jacobian, and diagonal weighting matrix, respectively. λ represents the damping parameter, which can be adjusted to be large or small. I is the identity matrix, h is the perturbation, y is a measured point set, and the fitted function is denoted by $\hat{y}$ [62].

In addition to *NDSolve* and *FindFit* solver, another important command performed in the optimization part of the present design problem is *NMinimize*. A detailed explanation of *NMiminize* is given in the previous section (see Chapter 4).

5.2 Engineering Model

Components of a typical DMF are (1) primary flywheel, (2) spring, (3) flange, and (4) secondary flywheel, as shown in Figure 5.1.

Although the flange is a component of the dual mass flywheel, it is assumed as a single piece with the component, which is the secondary flywheel. Accordingly, this model has six parameters. Firstly, the parameters whose J_1 and J_2 are the moment of inertia belong to the primary and secondary flywheel, respectively. The primary flywheel and secondary flywheel are merged using the torsional spring k and torsional damper c that determine the other parameters. The system has an engine torque, $M_e(t)$, and this torque acts upon a DMF. A torque that, contrary to engine torque, namely counter-torque, is applied to the backside of the driveline [63]. It is called Dual Mass Torsional Vibration Dynamic Absorber (DMTVDA) so that the system is more public than the traditional ones.

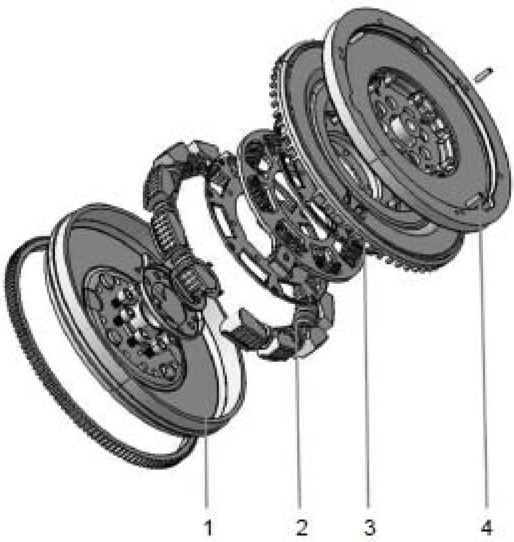


Figure 5.1: Dual mass flywheel

5.2.1 Mathematical Model

In order to obtain a mathematical model, the equations were constituted by considering the DMTVDA model and input parameters [63].

$$J_1\ddot{\varphi}_1 + c_1(\dot{\varphi}_1 - \dot{\varphi}_2) + k_1(\varphi_1 - \varphi_2) = M_e(t) \tag{5.3}$$

$$J_2\ddot{\varphi}_2 + c_1(\dot{\varphi}_2 - \dot{\varphi}_1) + k_1(\varphi_2 - \varphi_1) + c_2(\dot{\varphi}_2 - \dot{\varphi}_v) + k_2(\varphi_2 - \varphi_v) = 0 \tag{5.4}$$

where $c_2(\dot{\varphi}_2 - \dot{\varphi}_v) + k_2(\varphi_2 - \varphi_v)$ represents the gearbox input (the output side of the system) torque $M_v(t)$. All the parameters appear in Equations 5.3 and 5.4, and the definitions are listed in Table 5.1.

Some restrictions have been introduced to solve the design problem. Firstly, the six-cylinder truck engine will be taken into consideration. In this model, which includes the DMF and output shaft joined to the gearbox, the following assumption is valid for engine torque as a mathematical expression [61].

$$M_e(t) = M_0 + M_1 * \sin(w_e t + \alpha_1) \tag{1.2}$$

where α_1 is phase angle, M_0 is constant torque, M_1 represents the wave's amplitude. Lastly, it is assumed that the output side of the system does not have any vibration. Therefore, the following equations will be appropriate.

Table 5.1: Meaning of notations

$M_e(t)$	**Engine Torque**
$\varphi_1(t)$	The absolute angle of rotation of the primary flywheel
$\dot{\varphi}_1(t)$	The absolute angular speed of rotation of the primary flywheel
$\ddot{\varphi}_1(t)$	The absolute angular acceleration of the primary flywheel
$\varphi_2(t)$	The absolute angle of rotation of the secondary flywheel
$\dot{\varphi}_2(t)$	The absolute angular speed of rotation of the secondary flywheel
$\ddot{\varphi}_2(t)$	The absolute angular acceleration of the secondary flywheel
$\varphi_v(t)$	The absolute angle of rotation of the output
$\dot{\varphi}_v$	The absolute angular speed of rotation of the output
k_1	Torsional stiffness coefficient of the primary flywheel
c_1	Torsional damping coefficient of the primary flywheel
k_2	Torsional stiffness coefficient of the secondary flywheel
c_2	Torsional damping coefficient of the secondary flywheel
$M_v(t)$	Output torque

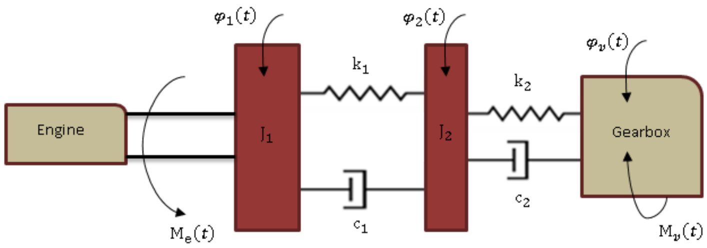


Figure 1.1: Free body diagram of DMTVDA model

$$\varphi_v = w_v t \text{ and } \dot{\varphi}_v = w_v \tag{5.5}$$

Three different cases (900, 1500, and 2000 RPM) can be considered in the speed range used for truck engines. These values correspond to low, medium, and high-speed models, respectively. Also, an engine that is a four-stroke six-cylinder will be preferred. One of the factors affecting the excitation frequency is the number of cylinders. The main excitation frequency matches the number of engines multiplied by half the number of cylinders [63].

Now, Equation 5.5 is necessary to calculate the engine torque. It should be known that the velocity from the gearbox side is applied for this calculation. 1/3 times the engine speed can be taken for angular velocity from the gearbox side, and it is assumed that (W_e/3)= W_v. 900, 1500, and 2000 RPM also correspond to 15, 25, and 33.33 Hz, respectively. In this regard, engine torque should be calculated by the process mentioned above.

Based on the experiences in this field, the best values for M_1 and M_0 are advised as 300 and 500, respectively. According to these selections, the engine torques for three cases (low, medium, and high speed) are obtained [64]. The scope of this study, it is considered only Equation 5.6.

$$M_{e1}(t) = 300 + 500 * \sin(45 * 2\pi t) \quad (5.6)$$

$$M_{e2}(t) = 300 + 500 * \sin(75 * 2\pi t) \quad (5.7)$$

$$M_{e3}(t) = 300 + 500 * \sin(100 * 2\pi t) \quad (5.8)$$

5.3 Optimization Problem Definition

In this study, in order to reduce the torsional vibration of the heavy vehicle's engine, four optimization problems have been defined (see Table 5.2). This work focuses on the optimization of the DMF for different scenarios of the truck. The primary function is to isolate the transmission input from the vibration generated by the engine. The system parameters J_1, J_2, the primary stiffness coefficient k_1, secondary stiffness coefficient k_2, and the damping coefficients c_1, c_2 are considered design variables.

A neuro-regression approach has been implemented for the present study. The resulting model is in the form of a fourth–order polynomial as in Equation 5.9:

$$
\begin{aligned}
&a(126)\,c_1^4 + 4\,a(121)\,c_1^3 + 4\,a(182)\,c_2\,c_1^3 + 4\,a(122)\,J_1\,c_1^3 + 4\,a(123)\,J_2\,c_1^3 + 4\,a(124)\,k_1\,c_1^3 + 4\,a(125)\,k_2\,c_1^3 + 6\,a(203)\,c_2^2\,c_1^2 + 6\,a(108)\,J_1^2\,c_1^2 + 6\,a(111)\,J_2^2\,c_1^2 + \\
&6\,a(115)\,k_1^2\,c_1^2 + 6\,a(120)\,k_2^2\,c_1^2 + 6\,a(106)\,c_1^2 + 12\,a(177)\,c_2\,c_1^2 + 12\,a(107)\,J_1\,c_1^2 + 12\,a(178)\,c_2\,J_1\,c_1^2 + 12\,a(109)\,J_2\,c_1^2 + 12\,a(179)\,c_2\,J_2\,c_1^2 + 12\,a(110)\,J_1\,J_2\,c_1^2 + \\
&12\,a(112)\,k_1\,c_1^2 + 12\,a(180)\,c_2\,k_1\,c_1^2 + 12\,a(113)\,J_1\,k_1\,c_1^2 + 12\,a(114)\,J_2\,k_1\,c_1^2 + 12\,a(116)\,k_2\,c_1^2 + 12\,a(181)\,c_2\,k_2\,c_1^2 + 12\,a(117)\,J_1\,k_2\,c_1^2 + 12\,a(118)\,J_2\,k_2\,c_1^2 + \\
&12\,a(119)\,k_1\,k_2\,c_1^2 + 4\,a(209)\,c_2^3\,c_1 + 4\,a(74)\,J_1^3\,c_1 + 4\,a(80)\,J_2^3\,c_1 + 4\,a(90)\,k_1^3\,c_1 + 4\,a(105)\,k_2^3\,c_1 + 12\,a(198)\,c_2^2\,c_1 + 12\,a(73)\,J_1^2\,c_1 + 12\,a(164)\,c_2\,J_1^2\,c_1 + \\
&12\,a(78)\,J_2^2\,c_1 + 12\,a(167)\,c_2\,J_2^2\,c_1 + 12\,a(79)\,J_1\,J_2^2\,c_1 + 12\,a(87)\,k_1^2\,c_1 + 12\,a(171)\,c_2\,k_1^2\,c_1 + 12\,a(88)\,J_1\,k_1^2\,c_1 + 12\,a(89)\,J_2\,k_1^2\,c_1 + 12\,a(101)\,k_2^2\,c_1 + \\
&12\,a(176)\,c_2\,k_2^2\,c_1 + 12\,a(102)\,J_1\,k_2^2\,c_1 + 12\,a(103)\,J_2\,k_2^2\,c_1 + 12\,a(104)\,k_1\,k_2^2\,c_1 + 4\,a(71)\,c_1 + 12\,a(162)\,c_2\,c_1 + 12\,a(199)\,c_2^2\,J_1\,c_1 + 12\,a(72)\,J_1\,c_1 + \\
&24\,a(163)\,c_2\,J_1\,c_1 + 12\,a(200)\,c_2^2\,J_2\,c_1 + 12\,a(77)\,J_1^2\,J_2\,c_1 + 12\,a(75)\,J_2\,c_1 + 24\,a(165)\,c_2\,J_2\,c_1 + 24\,a(76)\,J_1\,J_2\,c_1 + 24\,a(166)\,c_2\,J_1\,J_2\,c_1 + 12\,a(201)\,c_2^2\,k_1\,c_1 + \\
&12\,a(83)\,J_1^2\,k_1\,c_1 + 12\,a(86)\,J_2^2\,k_1\,c_1 + 12\,a(81)\,k_1\,c_1 + 24\,a(168)\,c_2\,k_1\,c_1 + 24\,a(82)\,J_1\,k_1\,c_1 + 24\,a(169)\,c_2\,J_1\,k_1\,c_1 + 24\,a(84)\,J_2\,k_1\,c_1 + 24\,a(170)\,c_2\,J_2\,k_1\,c_1 + \\
&24\,a(85)\,J_1\,J_2\,k_1\,c_1 + 12\,a(202)\,c_2^2\,k_2\,c_1 + 12\,a(93)\,J_1^2\,k_2\,c_1 + 12\,a(96)\,J_2^2\,k_2\,c_1 + 12\,a(100)\,k_1^2\,k_2\,c_1 + 12\,a(91)\,k_2\,c_1 + 24\,a(172)\,c_2\,k_2\,c_1 + 24\,a(92)\,J_1\,k_2\,c_1 + \\
&24\,a(173)\,c_2\,J_1\,k_2\,c_1 + 24\,a(94)\,J_2\,k_2\,c_1 + 24\,a(174)\,c_2\,J_2\,k_2\,c_1 + 24\,a(95)\,J_1\,J_2\,k_2\,c_1 + 24\,a(97)\,k_1\,k_2\,c_1 + 24\,a(175)\,c_2\,k_1\,k_2\,c_1 + 24\,a(98)\,J_1\,k_1\,k_2\,c_1 + \\
&24\,a(99)\,J_2\,k_1\,k_2\,c_1 + a(210)\,c_2^4 + a(5)\,J_1^4 + a(15)\,J_2^4 + a(35)\,k_1^4 + a(70)\,k_2^4 + 4\,a(204)\,c_2^3 + 4\,a(4)\,J_1^3 + 4\,a(130)\,c_2\,J_1^3 + 4\,a(13)\,J_2^3 + 4\,a(136)\,c_2\,J_2^3 + \\
&4\,a(14)\,J_1\,J_2^3 + 4\,a(32)\,k_1^3 + 4\,a(146)\,c_2\,k_1^3 + 4\,a(33)\,J_1\,k_1^3 + 4\,a(34)\,J_2\,k_1^3 + 4\,a(66)\,k_2^3 + 4\,a(161)\,c_2\,k_2^3 + 4\,a(67)\,J_1\,k_2^3 + 4\,a(68)\,J_2\,k_2^3 + 4\,a(69)\,k_1\,k_2^3 + \\
&6\,a(183)\,c_2^2 + 6\,a(185)\,c_2^2\,J_1^2 + 6\,a(3)\,J_1^2 + 12\,a(129)\,c_2\,J_1^2 + 6\,a(188)\,c_2^2\,J_2^2 + 6\,a(12)\,J_1^2\,J_2^2 + 6\,a(10)\,J_2^2 + 12\,a(134)\,c_2\,J_2^2 + 12\,a(11)\,J_1\,J_2^2 + 12\,a(135)\,c_2\,J_1\,J_2^2 + \\
&6\,a(192)\,c_2^2\,k_1^2 + 6\,a(28)\,J_1^2\,k_1^2 + 6\,a(31)\,J_2^2\,k_1^2 + 6\,a(26)\,k_1^2 + 12\,a(143)\,c_2\,k_1^2 + 12\,a(27)\,J_1\,k_1^2 + 12\,a(144)\,c_2\,J_1\,k_1^2 + 12\,a(29)\,J_2\,k_1^2 + 12\,a(145)\,c_2\,J_2\,k_1^2 + \\
&12\,a(30)\,J_1\,J_2\,k_1^2 + 6\,a(197)\,c_2^2\,k_2^2 + 6\,a(58)\,J_1^2\,k_2^2 + 6\,a(61)\,J_2^2\,k_2^2 + 6\,a(65)\,k_1^2\,k_2^2 + 6\,a(56)\,k_2^2 + 12\,a(157)\,c_2\,k_2^2 + 12\,a(57)\,J_1\,k_2^2 + 12\,a(158)\,c_2\,J_1\,k_2^2 + \\
&12\,a(59)\,J_2\,k_2^2 + 12\,a(159)\,c_2\,J_2\,k_2^2 + 12\,a(60)\,J_1\,J_2\,k_2^2 + 12\,a(62)\,k_1\,k_2^2 + 12\,a(160)\,c_2\,k_1\,k_2^2 + 12\,a(63)\,J_1\,k_1\,k_2^2 + 12\,a(64)\,J_2\,k_1\,k_2^2 + a(1) + 4\,a(127)\,c_2 + \\
&4\,a(205)\,c_2^3\,J_1 + 12\,a(184)\,c_2^2\,J_1 + 4\,a(2)\,J_1 + 12\,a(128)\,c_2\,J_1 + 4\,a(206)\,c_2^3\,J_2 + 4\,a(9)\,J_1^3\,J_2 + 12\,a(186)\,c_2^2\,J_2 + 12\,a(8)\,J_1^2\,J_2 + 12\,a(133)\,c_2\,J_1^2\,J_2 + \\
&4\,a(6)\,J_2 + 12\,a(131)\,c_2\,J_2 + 12\,a(187)\,c_2^2\,J_1\,J_2 + 12\,a(7)\,J_1\,J_2 + 24\,a(132)\,c_2\,J_1\,J_2 + 4\,a(207)\,c_2^3\,k_1 + 4\,a(19)\,J_1^3\,k_1 + 4\,a(25)\,J_2^3\,k_1 + 12\,a(189)\,c_2^2\,k_1 + \\
&12\,a(18)\,J_1^2\,k_1 + 12\,a(139)\,c_2\,J_1^2\,k_1 + 12\,a(23)\,J_2^2\,k_1 + 12\,a(142)\,c_2\,J_2^2\,k_1 + 12\,a(24)\,J_1\,J_2^2\,k_1 + 4\,a(16)\,k_1 + 12\,a(137)\,c_2\,k_1 + 12\,a(190)\,c_2^2\,J_1\,k_1 + \\
&12\,a(17)\,J_1\,k_1 + 24\,a(138)\,c_2\,J_1\,k_1 + 12\,a(191)\,c_2^2\,J_2\,k_1 + 12\,a(22)\,J_1^2\,J_2\,k_1 + 12\,a(20)\,J_2\,k_1 + 24\,a(140)\,c_2\,J_2\,k_1 + 24\,a(21)\,J_1\,J_2\,k_1 + 24\,a(141)\,c_2\,J_1\,J_2\,k_1 + \\
&4\,a(208)\,c_2^3\,k_2 + 4\,a(39)\,J_1^3\,k_2 + 4\,a(45)\,J_2^3\,k_2 + 4\,a(55)\,k_1^3\,k_2 + 12\,a(193)\,c_2^2\,k_2 + 12\,a(38)\,J_1^2\,k_2 + 12\,a(149)\,c_2\,J_1^2\,k_2 + 12\,a(43)\,J_2^2\,k_2 + 12\,a(152)\,c_2\,J_2^2\,k_2 + \\
&12\,a(44)\,J_1\,J_2^2\,k_2 + 12\,a(52)\,k_1^2\,k_2 + 12\,a(156)\,c_2\,k_1^2\,k_2 + 12\,a(53)\,J_1\,k_1^2\,k_2 + 12\,a(54)\,J_2\,k_1^2\,k_2 + 4\,a(36)\,k_2 + 12\,a(147)\,c_2\,k_2 + 12\,a(194)\,c_2^2\,J_1\,k_2 + \\
&12\,a(37)\,J_1\,k_2 + 24\,a(148)\,c_2\,J_1\,k_2 + 12\,a(195)\,c_2^2\,J_2\,k_2 + 12\,a(42)\,J_1^2\,J_2\,k_2 + 12\,a(40)\,J_2\,k_2 + 24\,a(150)\,c_2\,J_2\,k_2 + 24\,a(41)\,J_1\,J_2\,k_2 + 24\,a(151)\,c_2\,J_1\,J_2\,k_2 + \\
&12\,a(196)\,c_2^2\,k_1\,k_2 + 12\,a(48)\,J_1^2\,k_1\,k_2 + 12\,a(51)\,J_2^2\,k_1\,k_2 + 12\,a(46)\,k_1\,k_2 + 24\,a(153)\,c_2\,k_1\,k_2 + 24\,a(47)\,J_1\,k_1\,k_2 + 24\,a(154)\,c_2\,J_1\,k_1\,k_2 + 24\,a(49)\,J_2\,k_1\,k_2 + \\
&24\,a(155)\,c_2\,J_2\,k_1\,k_2 + 24\,a(50)\,J_1\,J_2\,k_1\,k_2
\end{aligned}
\tag{5.9}
$$

5.4 Results and Discussion

To verify the accuracy of the predictions during the modeling phase, a hybrid method is utilized, which combines the benefits of regression analysis and artificial neural networks (ANN). In this methodology, all of the data is separated into two sets, each containing 80% and 20% of the total data, with the first portion being used for training and the second for testing. The goal of the training process is to reduce the error between experimental and expected values by changing the regression models and coefficients listed in Table 5.3.

Following that, the testing phase is used to achieve the prediction results by reducing the effects of regression model inconsistencies. This procedure yields information about the candidate models' ability to anticipate. Next, checking the boundedness of candidate models for prescribed values is critical in determining whether or not the model is realistic. In this case, the maximum and minimum values of the models in the given interval for each design variable are calculated after acquiring the right models in terms of R^2 training and testing. This method helps to identify whether the chosen models meet the several requirements that are required for realism.

In the present study, 16 different regression models (see Table 5.3) with six parameters have been tested, and the results are listed in Table 5.4.

Table 5.2: Four different scenarios

Problem No	Objectives	Constraints
Problem 1	**Minimize** $\varphi_1(J_1, J_2, c_1, c_2, k_1, k_2) - \varphi_2(J_1, J_2, c_1, c_2, k_1, k_2)$	$1.1 \leq J_1 \leq 1.8$ $0.5 \leq J_2 \leq 0.6$ $25 \leq c_1 \leq 30$ $15 \leq c_2 \leq 20$ $18000 \leq k_1 \leq 20000$ $5500 \leq k_2 \leq 16500$
Problem 2	**Minimize** $\varphi_1(J_1, J_2, c_1, c_2, k_1, k_2) - \varphi_2(J_1, J_2, c_1, c_2, k_1, k_2)$	$1.1 \leq J_1 \leq 1.8$ $0.5 \leq J_2 \leq 0.6$ $25 \leq c_1 \leq 30$ & $c_1 \in$ Integers $15 \leq c_2 \leq 20$ & $c_2 \in$ Integers $18000 \leq k_1 \leq 20000$ & $k_1 \in$ Integers $5500 \leq k_2 \leq 16500$ & $k_2 \in$ Integers
Problem 3	**Minimize** $\varphi_1(J_1, J_2, c_1, c_2, k_1, k_2) - \varphi_2(J_1, J_2, c_1, c_2, k_1, k_2)$	$1.1 \leq J_1 \leq 1.8$ $0.5 \leq J_2 \leq 0.6$ $c_1 \in \{25, 30\}$ $c_2 \in \{15, 20\}$ $k_1 \in \{18000, 18500, 19000, 19500, 20000\}$ $k_2 \in \{5500, 6500, \ldots, 15500, 16500\}$
Problem 4	**Minimize** $\varphi_1(J_1, J_2, c_1, c_2, k_1, k_2) - \varphi_2(J_1, J_2, c_1, c_2, k_1, k_2)$	$1.1 \leq J_1 \leq 1.8$ $0.5 \leq J_2 \leq 0.6$ $c_1 \in \{30, 25\}$ $c_2 \in \{15, 20\}$ $k_1 \in \{18000, 16000, 12000, 20000\}$ $k_2 \in \{16500, 11000, 5500\}$

Table 5.3: Multiple regression model types including linear, quadratic, trigonometric, logarithmic, and their rational forms.

Model Name	Nomenclature	Formula
Multiple linear	L	$Y = \sum_{i=1}^{6}(a_i x_i) + c$
Multiple linear rational	LR	$Y = \frac{\sum_{i=1}^{6}(a_i x_i) + c_1}{\sum_{j=1}^{6}(\beta_j x_j)} + c_2$
Second order multiple nonlinear	SON	$Y = \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c$
Second order multiple nonlinear rational	SONR	$Y = \frac{\sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c_1}{\sum_{l=1}^{6}\sum_{m=1}^{6}(\beta_m x_m x_l) + \sum_{n=1}^{6}(\beta_n x_n)} + c_2$
Third order multiple nonlinear	TON	$Y = \sum_{l=1}^{6}\sum_{m=1}^{6}\sum_{p=1}^{6}(\beta_l x_l x_m x_p) + \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c$
Third order multiple nonlinear rational	TONR	$Y = \frac{\sum_{l=1}^{6}\sum_{m=1}^{6}\sum_{p=1}^{6}(\beta_l x_l x_m x_p) + \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c_1}{\sum_{f=1}^{6}\sum_{r=1}^{6}\sum_{v=1}^{6}(\theta_f x_f x_r x_v) + \sum_{t=1}^{6}\sum_{s=1}^{6}(\gamma_s x_s x_t) + \sum_{n=1}^{6}(\gamma_n x_n)} + c_2$
Fourth order multiple nonlinear	FON	$Y = \sum_{r=1}^{6}\sum_{s=1}^{6}\sum_{t=1}^{6}\sum_{v=1}^{6}(\gamma_r x_r x_s x_t x_v) + \sum_{l=1}^{6}\sum_{m=1}^{6}\sum_{p=1}^{6}(\beta_l x_l x_m x_p) + \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c$
Fourth order multiple nonlinear rational	FONR	$Y = \frac{\sum_{r=1}^{6}\sum_{s=1}^{6}\sum_{t=1}^{6}\sum_{v=1}^{6}(\gamma_r x_r x_s x_t x_v) + \sum_{l=1}^{6}\sum_{m=1}^{6}\sum_{p=1}^{6}(\beta_l x_l x_m x_p) + \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j x_j x_k) + \sum_{i=1}^{6}(a_i x_i) + c_1}{\sum_{f=1}^{6}\sum_{e=1}^{6}\sum_{u=1}^{6}\sum_{c=1}^{6}(\nu_f x_f x_e x_u x_c) + \sum_{w=1}^{6}\sum_{z=1}^{6}\sum_{y=1}^{6}(\delta_w x_w x_z x_y) + \sum_{g=1}^{6}\sum_{h=1}^{6}(\theta_h x_h x_g) + \sum_{d=1}^{6}(\theta_d x_d)} + c_2$

Table 5.3: Multiple regression model types including linear, quadratic, trigonometric, logarithmic, and their rational forms.

First order trigonometric multiple nonlinear	**FOTN**	$\boldsymbol{Y = \sum_{i=1}^{6}(a_i Sin[x_i] + a_i Cos[x_i]) + c}$
First order trigonometric multiple nonlinear rational	FOTNR	$Y = \frac{\sum_{i=1}^{6}(a_i Sin[x_i] + a_i Cos[x_i]) + c_1}{\sum_{j=1}^{6}(\beta_j Sin[x_j] + \gamma_j Cos[x_j])} + c_2$
Second order trigonometric multiple nonlinear	SOTN	$Y = \sum_{i=1}^{6}(a_i Sin[x_i] + a_i Cos[x_i]) + \sum_{j=1}^{6}(\beta_j Sin^2[x_j] + \gamma_j Cos^2[x_j]) + c$
Second order trigonometric multiple nonlinear rational	SOTNR	$Y = \frac{\sum_{i=1}^{6}(a_i Sin[x_i] + a_i Cos[x_i]) + \sum_{j=1}^{6}(\beta_j Sin^2[x_j] + \gamma_j Cos^2[x_j]) + c_1}{\sum_{k=1}^{6}(\theta_i Sin[x_k] + \theta_i Cos[x_k]) + \sum_{l=1}^{6}(\delta_j Sin^2[x_l] + \delta_j Cos^2[x_l])} + c_2$
First order logarithmic multiple nonlinear	FOLN	$Y = \sum_{i=1}^{6}(a_i Log[x_i]) + c$
First order logarithmic multiple nonlinear rational	FOLNR	$Y = \frac{\sum_{i=1}^{6}(a_i Log[x_i]) + c_1}{\sum_{j=1}^{6}(\beta_j Log[x_j])} + c_2$
Second order logarithmic multiple nonlinear	SOLN	$Y = \sum_{k=1}^{6}\sum_{j=1}^{6}(a_j Log[x_j x_k]) + \sum_{i=1}^{6}(a_i Log[x_i]) + c$
Second order logarithmic multiple nonlinear rational	SOLNR	$Y = \frac{\sum_{k=1}^{6}\sum_{j=1}^{6}(a_j Log[x_j x_k]) + \sum_{i=1}^{6}(a_i Log[x_i]) + c_1}{\sum_{m=1}^{6}\sum_{l=1}^{6}(a_l Log[x_l x_m]) + \sum_{n=1}^{6}(a_n Log[x_n])} + c_2$

In light of the above-described methods, the optimal design of a dual mass flywheel was organized as follows.

(i) The input dataset was created, and they have been modeled.

(ii) Sixteen candidate functional structures are proposed for modeling the data of the dual mass flywheel design, and the "boundedness of the functions" are evaluated for the appropriateness in terms of $R^2_{training}$ and $R^2_{testing}$ values.

(iii) Four different optimization problems are introduced using the obtained appropriate models, and these problems are solved by four different algorithms.

Considering only the low speed model (900 RPM), the system of equation is solved for 768 different matches (combination of the cases for $J_1 \in \{1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1\}$, $J_2 \in \{0.6, 0.5\}$, $c_1 \in \{30, 25\}$, $c_1 \in \{15, 20\}$, $k_1 \in \{18000, 16000, 12000, 20000\}$, $k_2 \in \{16500, 11000, 5500\}$). The results have been obtained as in an interpolation function representation.

Table 5.4: Results of the Neuro-regression models in terms of fitting performance and boundedness.

Models	$R^2_{training}$	$R^2_{\text{trainingAdj}}$	$R^2_{testing}$	Maximum	Minimum
L	0.997554	0.997496	0.981565	0.013955	0.0036876
LR	0.999274	0.999257	0.993045	0.0136434	0.00283717
SON	0.999572	0.999529	0.995834	0.0137186	0.00267966
SONR	0.999968	0.999965	0.99976	0.243562	-0.125976
TON	0.999915	0.999883	0.999304	0.0138862	0.00275177
TONR	1	1	0.999999	0.741632	-0.512173
FON	0.999993	0.999975	0.999917	0.0139884	0.00254153
FONR	1	1	1	3.75876	-2.37302
FOTN	0.902868	0.898507	0.260734	0.0241933	0.00380621
FOTNR	0.980851	0.979991	0.8442	1.39325*10^12	-3426.18
SOTN	0.999295	0.998992	0.994627	0.0284267	-0.0160677
SOTNR	0.999827	0.999752	0.998114	791082.	-4.97778*10^7
FOLN	0.998687	0.998655	0.990099	0.0136888	0.0040109
FOLNR	0.999391	0.999376	0.994025	0.013655	0.00266919
SOLN	0.999573	0.999529	0.995747	0.0156984	0.00364282
SOLNR	0.999977	0.999974	0.999815	2.04287	-0.105388

Table 5.5: Randomly selected differential equation system results for different input parameters

Case Number	J_1 $(kg.m^2)$	J_2 $(kg.m^2)$	k_1 (Nm/rad)	k_2 (Nm/rad)	c_1 (Nms/rad)	c_2 (Nms/rad)	t=1 (s)	t=2 (s)	t=3 (s)	t=4 (s)	t=5 (s)	t=6 (s)	t=7 (s)	t=8 (s)	t=9 (s)	t=10 (s)
							$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)
1	1.8	0.6	18000	16500	30	20	0.0124297	0.0119706	0.0119829	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828
50	1.8	0.5	18000	16500	30	15	0.014873	0.0106042	0.011032	0.0110011	0.0110028	0.0110028	0.0110028	0.0110028	0.0110028	0.0110028
100	1.7	0.6	18000	16500	25	15	0.0184136	0.0114281	0.0111446	0.0111844	0.0111872	0.011187	0.0111869	0.0111869	0.0111869	0.0111869
150	1.7	0.5	18000	11000	30	15	0.0126564	0.0107916	0.0109811	0.0109635	0.0109651	0.0109649	0.0109649	0.0109649	0.0109649	0.0109649
200	1.6	0.6	18000	11000	25	15	0.017757	0.0115795	0.0119826	0.0119873	0.0119843	0.0119845	0.0119845	0.0119845	0.0119845	0.0119845
300	1.5	0.6	18000	5500	25	15	0.0141981	0.0124353	0.0125747	0.0125648	0.0125655	0.0125655	0.0125655	0.0125655	0.0125655	0.0125655
500	1.3	0.6	16000	11000	25	15	0.0114029	0.0138163	0.0136271	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338
700	1.1	0.6	12000	16500	25	15	0.0196211	0.0199049	0.0199096	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097
768	1.1	0.5	20000	5500	25	15	0.00447687	0.0035706	0.00356775	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818

Table 5.6: Graphical representation of all differential equation system results for different input parameters

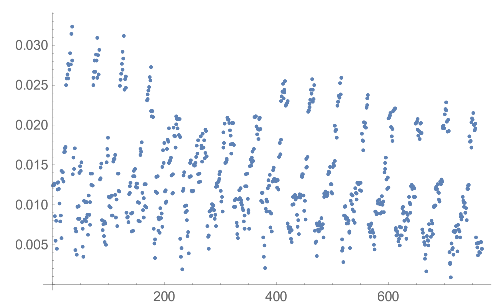
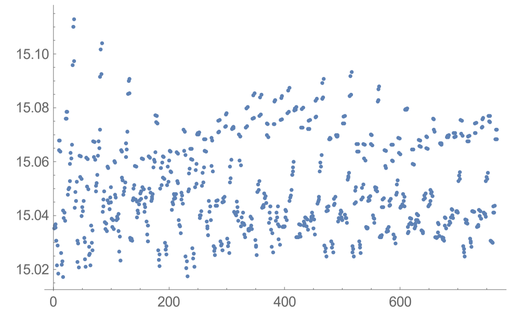
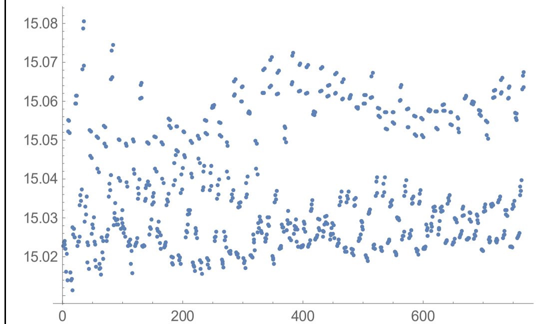
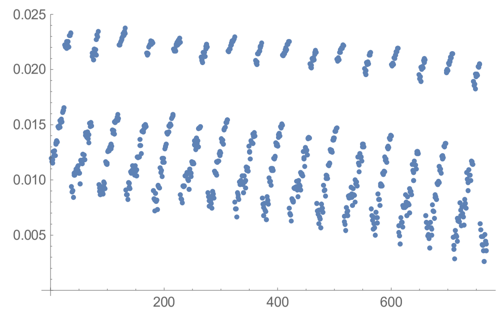
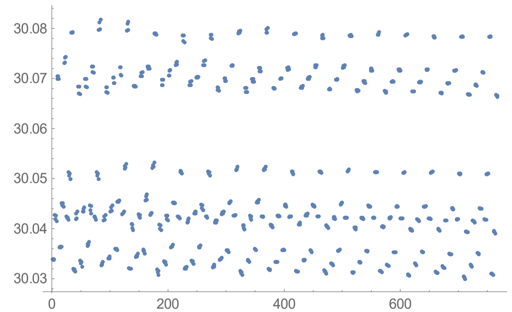
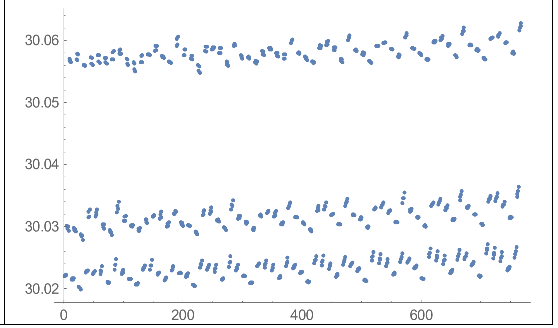
t (s)	$\varphi_1 - \varphi_2$ (rad) / Case Number	φ_1 (rad) / Case Number	φ_2 (rad) / Case Number
1			
2			

Table 5.6: Graphical representation of all differential equation system results for different input parameters

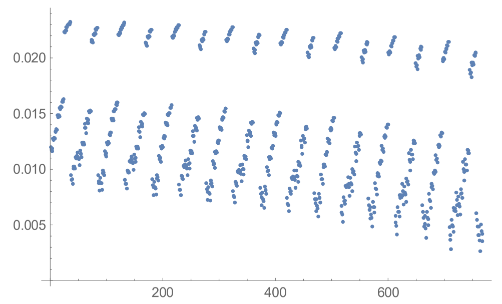
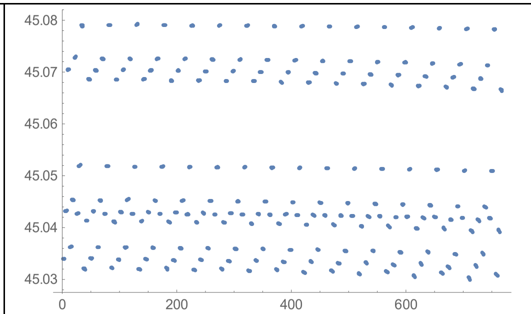
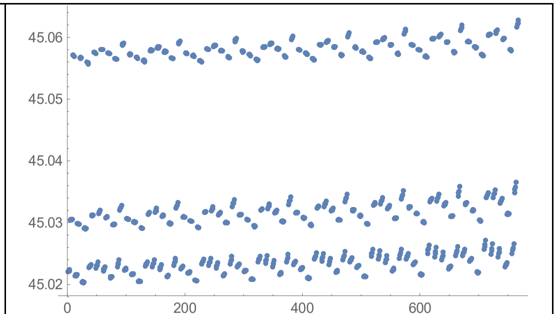
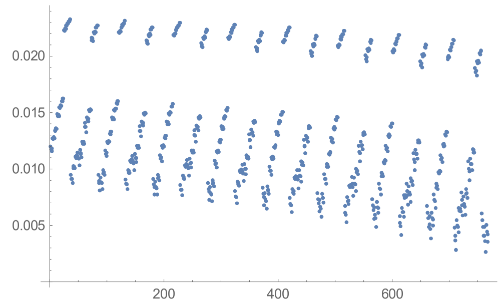
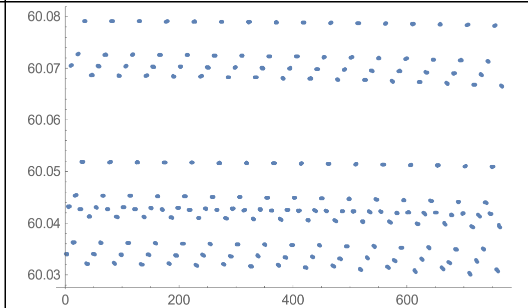
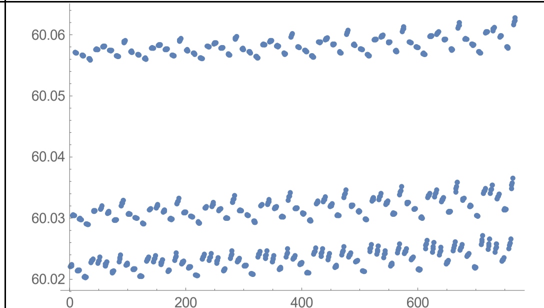
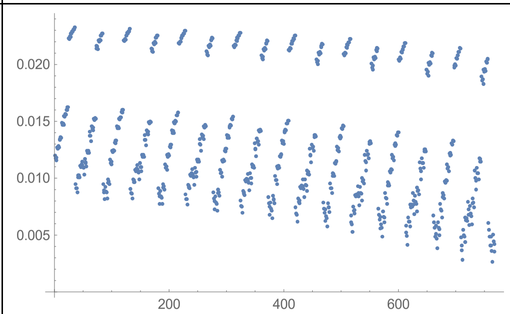
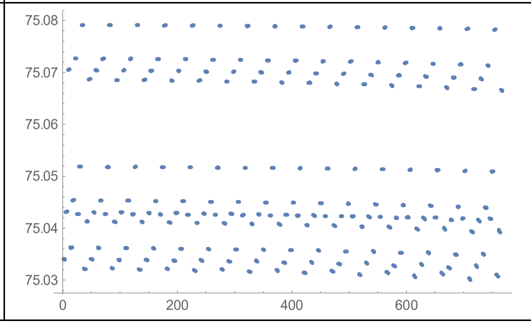
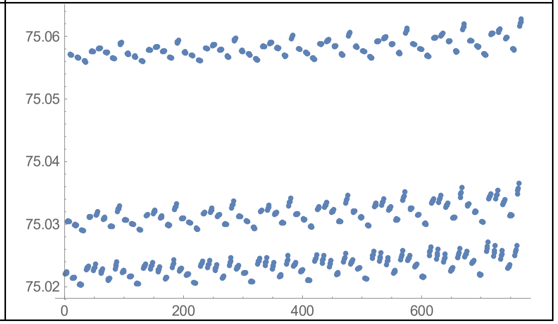
3			
4			
5			

Table 5.6: Graphical representation of all differential equation system results for different input parameters

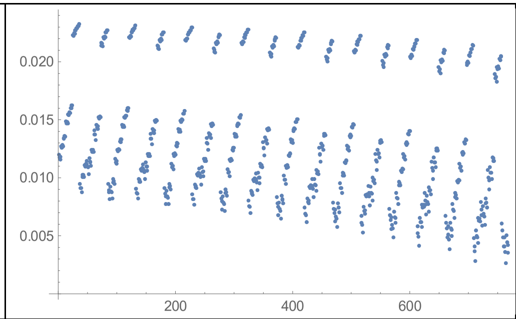
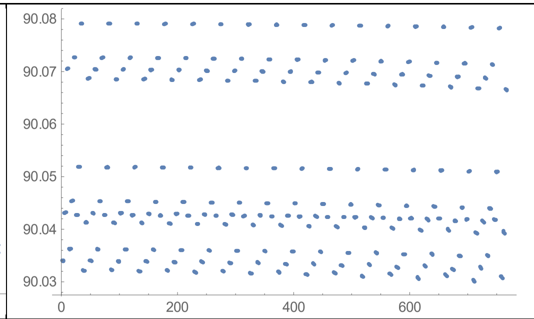
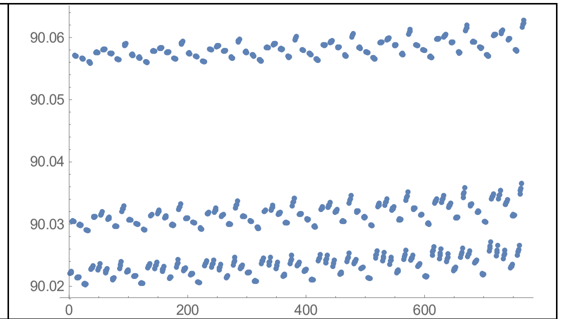
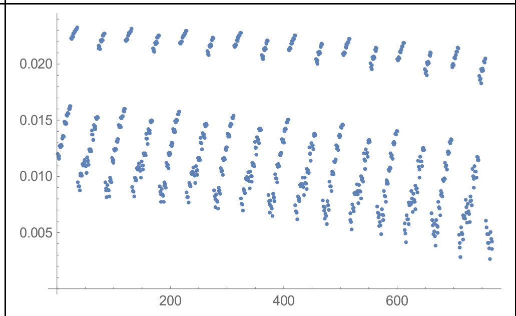
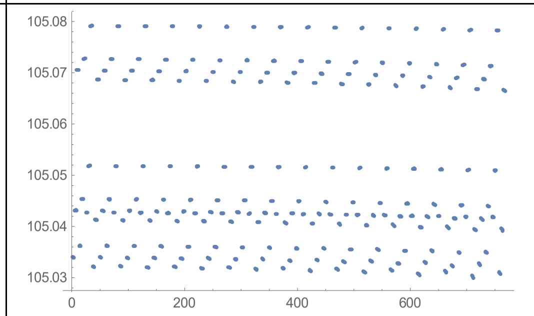
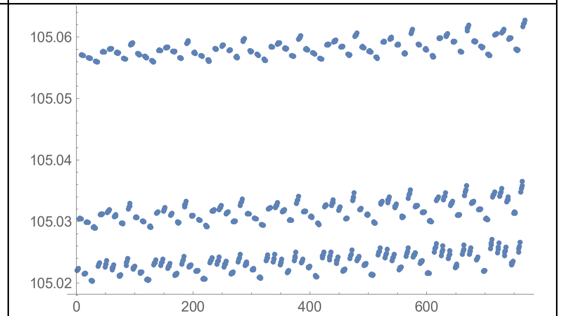
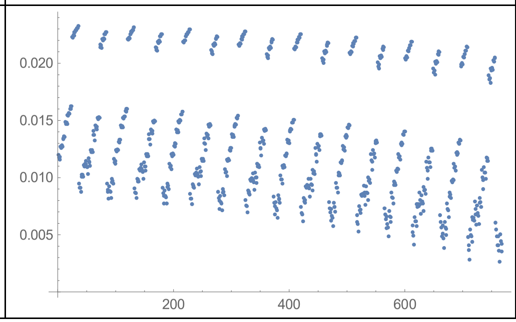
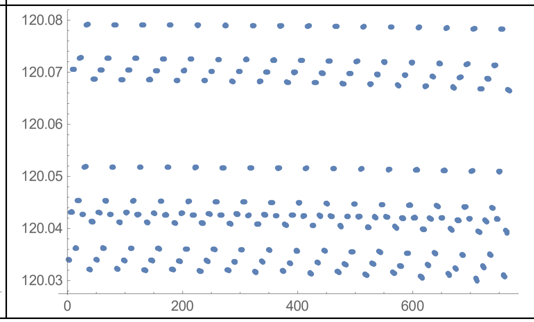
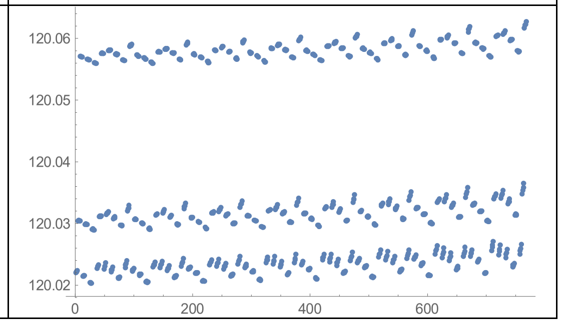
6			
7			
8			

Table 5.6: Graphical representation of all differential equation system results for different input parameters

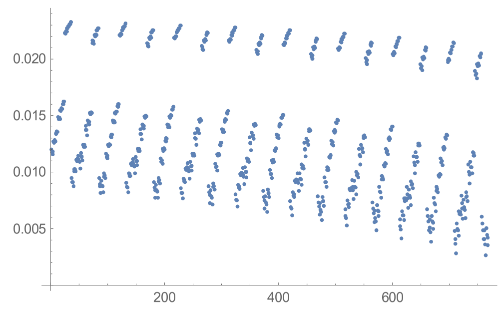
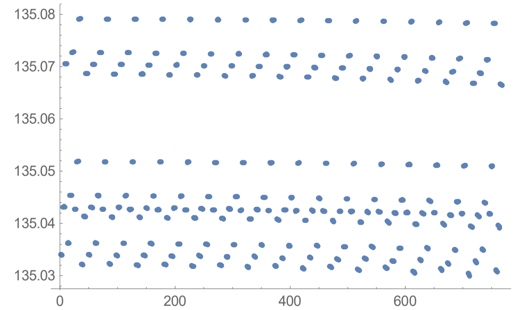
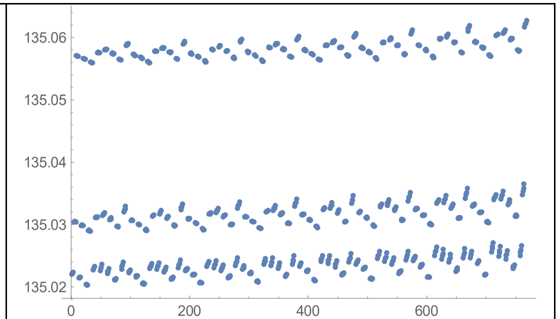
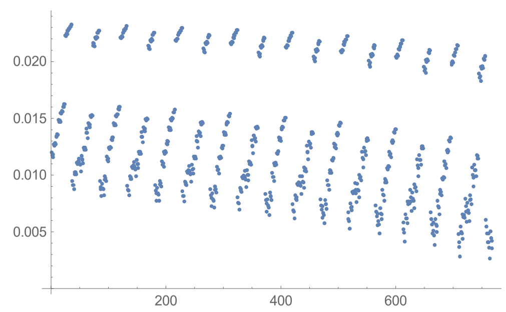
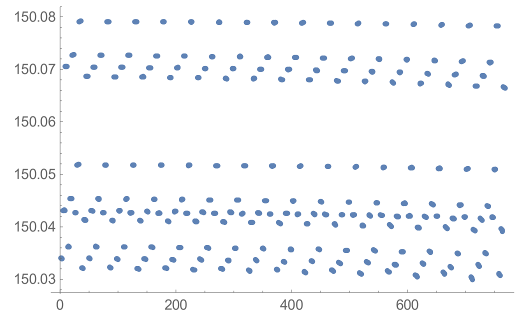
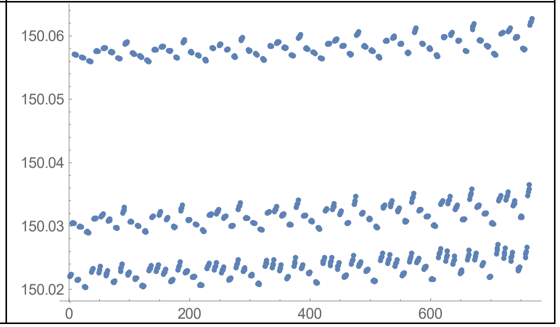
9			
10			

Table 5.5 represents the values of the numerical solutions of the randomly selected nine solutions. The table containing the results in all cases is given in Appendix A. When the table is examined, it is seen that different $\varphi_1 - \varphi_2$ values are obtained at different times.

Considering all these cases, after t_4, the results obtained in the 1., 500., 700. and 768. cases have respectively remained constant as 0.119828 and 0.0136338, 0.0199097 and 0.00356818 rad. This situation remains stable after t=5s in the case numbers 50 and 300, t=6s in the case number 150 and 200, and after t=7s in the case number 100. Especially at t=1s, it is observed that the standard deviation is higher than other time values. The standard deviation at t=1s is 0.045%. For other time values result is 0.041%. Even if these differences are observed, the results at all t values are not far from the average results.

The graph of the results, including all cases, is given in Table 5.6. In the table, the x-axis represents the result values, and the y-axis represents the case numbers. The result values are in radians, in the first column $(\varphi_1 - \varphi_2)$, φ_1 in the second column and φ_2 in the third column. Each graph shows the value of φ_1, φ_2 and $(\varphi_1 - \varphi_2)$ with blue dots. This table gives us the general behavior of $(\varphi_1 - \varphi_2)$, φ_1 and φ_2 at t=1, 2, 3, …, 10s. When we analyze the graphs, the results of $(\varphi_1 - \varphi_2)$ shown in the first column appear to parallel vertically, while a generally parallel horizontal behavior is observed in columns φ_1 and φ_2. When the behavior of $(\varphi_1 - \varphi_2)$ is examined, it is seen that the result data is in the range of 0 – 0.030 rad at time t_1, 0 – 0.025 rad at time t_2, and 0 – 0.022 rad from time t_3 to t_{10}. It has been determined that φ_1 and φ_2 values vary between 15 and 150 rad from t_1 to t_{10}. This change can also be expressed as 859.437 - 85943.7 degrees. Looking at the results for φ_1, the results at time t_1 are in the range of 15.02 – 15.10 rad. For other t values, results were observed in the range of 30.03 – 30.08, 45.03 – 45.08, 60.03 – 60.08, 75.03 – 75.08, 90.03 – 90.08, 105.03 – 105.08, 120.03 – 120.08 and 135.03 – 135.08, 150.03 – 150.08 rads respectively. Similarly, the results for φ_2 are in the range of 15.02 – 15.08, 30.02 – 30.06, 45.02 – 45.06, 60.02 – 60.06, 70.02 – 70.06, 90.02 – 90.06, 105.02 – 105.06, 120.02 – 120.06 and 135.02 – 135.06, 150.02 – 150.06 rads, respectively. When the graphs given for

φ_1 and φ_2 are examined, it is observed that although the numerical values of the results are increasing over time, these increase amounts are the same. Therefore, the results for $\varphi_1 - \varphi_2$ showed little change and remained unchanged after t_3.

Before constituting the mathematical model, 768 lines of data are divided into two parts as 20% and 80% of it. The multiple nonlinear regression model is constituted using 80% part of the data, and the values in the 20% ones are used for testing.

Sixteen different regression models have been tested (see Appendix B), and according to results in Table 5.4, results with R^2 values close to 1 have been taken into account. Thus, it is seen that the first order trigonometric multiple nonlinear (FOTN) model does not give a good value. The boundedness of the function (created model) considering the limitation of the engineering parameters have been checked. Considering the minimum and maximum limits, it is thought that it is not possible for the masses to rotate in the opposite direction in practice. Thus, it is seen that models with a negative minimum value are also not suitable. In the remaining models, the fourth–order multiple nonlinear (FON) model with the lowest minimum limit has been chosen as the best model to be used in this study.

In Table 5.7, the statistical accuracy of the model is shown, and the results are given for $R^2_{Training}$ and $R^2_{Testing}$ as 0.999993 and 0.999917, respectively. In the given model in Table 5.7 boundedness of the function (created model) considering the limitation of the engineering parameters have been checked. The maximum angular displacement is obtained as 0.0139884 radians (0.8015degrees), while the minimum angular displacement is obtained as 0.00254153 radians (0.1456 degrees). As a result, it is confirmed that the model works correctly between these limits. Therefore, the model is to be realistic.

Four different optimization problems are introduced. In these four different problems, the constraint ranges have been limited from problem 1 to problem 4. While the first problem shows us the optimum result that can be obtained theoretically, in other problems, the constraints are rearranged in line with market aspirations. For each problem, the optimization studies are based on four different methods (DE, SA, RS, and NM). Sixteen results are obtained for four problems, and these results are compared.

The optimum results are obtained by minimizing the differences between angular displacements of the first and second flywheels using values of the parameters in the aforementioned section. The optimization results are given in Table 5.8. As it is seen in the first optimization problem, the relative displacement $\varphi_1 - \varphi_2$ value is observed to be considerably less than 0,00356775 rad that is the smallest value obtained with the system of differential equation solution. This solution gives approximate results with DE, SA, RS, and NM optimization algorithms as 0.00254144, 0.00254152, 0.00254147, and 0.00254153 rad, respectively. It is observed that 28.77% improvement as a result of the first problem. When problem 1 is examined, it can be seen that there are some values of parameters that cannot be used in the real application. However, it should be noted that the aim of problem one is to show the theoretical limits. For the second optimization problem, it is requested from the system that all c and k values are integers. It is seen that the best result is obtained by using the DE optimization algorithm for constraints of problem 2. This result is 16% higher than the first problem. The third and fourth optimization problems are studied that limited parameters value ranges in order that it can be used in the real application. In optimization problem 3, the results for all optimization algorithms are the same and 0.00254145 radians. In optimization problem 4, the results are also the same in all optimization algorithms and 0.00273573 rad. When these results are examined, it is seen that results are close to other optimization problems. It is seen that the minimum value of $\varphi_1 - \varphi_2$ is 28.7% better than the smallest value obtained with the system of differential equation solution.

According to all results of optimizations, regardless of the constraints, optimum results are obtained when J_1 value is 1.1 kg.m^2 and k_1 value is 20000 Nm/rad.

Table 5.7: Introduced model and its statistical accuracy

Model (φ1 – φ2)	$R^2_{training}$	$R^2_{testing}$
$0.0164528 + 0.000353783 c_1 + 4.08468\times10^{-6} c_1^2 - 1.40915\times10^{-7} c_1^3 - 1.39352\times10^{-8} c_1^4 + 0.00054463 c_2 + 0.0000119847 c_1 c_2 +$ $1.54242\times10^{-7} c_1^2 c_2 - 3.69772\times10^{-9} c_1^3 c_2 + 9.2765\times10^{-6} c_2^2 + 2.27897\times10^{-7} c_1 c_2^2 + 4.29286\times10^{-9} c_1^2 c_2^2 - 4.63219\times10^{-7} c_2^3 -$ $7.86278\times10^{-9} c_1 c_2^3 - 6.46622\times10^{-8} c_2^4 + 0.00958861 J_1 + 0.000159591 c_1 J_1 - 8.69316\times10^{-7} c_1^2 J_1 - 2.47714\times10^{-7} c_1^3 J_1 +$ $0.00028949 c_2 J_1 + 2.66922\times10^{-6} c_1 c_2 J_1 - 1.76848\times10^{-7} c_1^2 c_2 J_1 + 2.53374\times10^{-6} c_2^2 J_1 - 1.82735\times10^{-7} c_1 c_2^2 J_1 -$ $4.80243\times10^{-7} c_2^3 J_1 - 0.0207402 J_1^2 - 0.000232781 c_1 J_1^2 + 9.72449\times10^{-6} c_1^2 J_1^2 - 0.000489108 c_2 J_1^2 + 4.17107\times10^{-6} c_1 c_2 J_1^2 +$ $8.79303\times10^{-6} c_2^2 J_1^2 + 0.0130297 J_1^3 - 0.000126583 c_1 J_1^3 - 7.5823\times10^{-6} c_2 J_1^3 - 0.0021551 J_1^4 + 0.0180561 J_2 + 0.000468797 c_1 J_2 +$ $0.0000101277 c_1^2 J_2 + 1.34261\times10^{-7} c_1^3 J_2 + 0.000668528 c_2 J_2 + 7.20328\times10^{-6} c_1 c_2 J_2 - 3.3783\times10^{-7} c_1^2 c_2 J_2 + 0.0000175815 c_2^2 J_2 -$ $6.08772\times10^{-7} c_1 c_2^2 J_2 + 4.04716\times10^{-8} c_2^3 J_2 + 0.00765823 J_1 J_2 + 0.000336248 c_1 J_1 J_2 + 0.0000139331 c_1^2 J_1 J_2 +$ $0.000304816 c_2 J_1 J_2 - 2.6099\times10^{-6} c_1 c_2 J_1 J_2 + 9.70887\times10^{-6} c_2^2 J_1 J_2 - 0.0106706 J_1^2 J_2 - 0.000238223 c_1 J_1^2 J_2 +$ $0.000250288 c_2 J_1^2 J_2 + 0.00745442 J_1^3 J_2 + 0.0115966 J_2^2 + 0.000530991 c_1 J_2^2 + 0.0000226418 c_1^2 J_2^2 + 0.000631066 c_2 J_2^2 -$ $0.0000124174 c_1 c_2 J_2^2 + 0.0000324425 c_2^2 J_2^2 - 0.00327085 J_1 J_2^2 + 0.000676346 c_1 J_1 J_2^2 + 0.000157986 c_2 J_1 J_2^2 + 0.0277019 J_1^2 J_2^2 -$ $0.0138367 J_2^3 + 0.000421768 c_1 J_2^3 + 0.000172168 c_2 J_2^3 - 0.0337066 J_1 J_2^3 - 0.07824 J_2^4 - 2.46884\times10^{-6} k_1 - 4.26753\times10^{-8} c_1 k_1 +$ $1.16027\times10^{-10} c_1^2 k_1 + 5.84152\times10^{-11} c_1^3 k_1 - 6.90245\times10^{-8} c_2 k_1 - 1.09358\times10^{-9} c_1 c_2 k_1 + 1.02513\times10^{-11} c_1^2 c_2 k_1 -$ $4.94836\times10^{-11} c_2^2 k_1 + 9.63045\times10^{-12} c_1 c_2^2 k_1 + 1.70013\times10^{-10} c_2^3 k_1 + 2.82778\times10^{-6} J_1 k_1 + 2.96211\times10^{-8} c_1 J_1 k_1 -$ $1.47726\times10^{-9} c_1^2 J_1 k_1 + 6.0633\times10^{-8} c_2 J_1 k_1 + 2.43668\times10^{-10} c_1 c_2 J_1 k_1 - 1.85511\times10^{-9} c_2^2 J_1 k_1 - 1.42751\times10^{-6} J_1^2 k_1 +$ $2.26465\times10^{-8} c_1 J_1^2 k_1 + 2.27155\times10^{-9} c_2 J_1^2 k_1 + 2.26678\times10^{-7} J_1^3 k_1 - 2.22392\times10^{-6} J_2 k_1 - 9.08145\times10^{-8} c_1 J_2 k_1 -$ $3.56836\times10^{-9} c_1^2 J_2 k_1 - 9.88098\times10^{-8} c_2 J_2 k_1 + 2.96215\times10^{-9} c_1 c_2 J_2 k_1 - 3.88893\times10^{-9} c_2^2 J_2 k_1 + 6.3274\times10^{-7} J_1 J_2 k_1 +$ $3.48752\times10^{-9} c_1 J_1 J_2 k_1 - 4.44132\times10^{-8} c_2 J_1 J_2 k_1 - 2.2636\times10^{-6} J_1^2 J_2 k_1 - 3.87335\times10^{-8} J_2^2 k_1 - 1.84034\times10^{-7} c_1 J_2^2 k_1 -$ $1.29301\times10^{-7} c_2 J_2^2 k_1 - 6.65686\times10^{-6} J_1 J_2^2 k_1 + 6.43036\times10^{-6} J_2^3 k_1 - 4.72415\times10^{-11} k_1^2 - 1.07989\times10^{-13} c_1 k_1^2 + 5.18875\times10^{-14} c_1^2 k_1^2 -$ $1.00011\times10^{-12} c_2 k_1^2 - 4.94057\times10^{-14} c_1 c_2 k_1^2 + 3.30099\times10^{-14} c_2^2 k_1^2 - 1.7543\times10^{-11} J_1 k_1^2 - 1.19612\times10^{-12} c_1 J_1 k_1^2 +$ $1.87888\times10^{-13} c_2 J_1 k_1^2 + 1.24189\times10^{-11} J_1^2 k_1^2 + 8.24677\times10^{-11} J_2 k_1^2 + 7.89921\times10^{-12} c_1 J_2 k_1^2 + 7.75362\times10^{-12} c_2 J_2 k_1^2 +$ $3.58616\times10^{-10} J_1 J_2 k_1^2 + 4.36744\times10^{-10} J_2^2 k_1^2 + 4.07916\times10^{-15} k_1^3 - 4.36288\times10^{-17} c_1 k_1^3 - 3.97047\times10^{-17} c_2 k_1^3 -$ $2.82614\times10^{-15} J_1 k_1^3 - 2.38498\times10^{-14} J_2 k_1^3 + 1.35141\times10^{-19} k_1^4 + 4.08672\times10^{-7} k_2 + 1.21471\times10^{-8} c_1 k_2 + 3.36791\times10^{-10} c_1^2 k_2 +$ $8.53703\times10^{-12} c_1^3 k_2 + 1.56429\times10^{-8} c_2 k_2 + 1.935\times10^{-10} c_1 c_2 k_2 - 6.16064\times10^{-12} c_1^2 c_2 k_2 + 4.50945\times10^{-10} c_2^2 k_2 -$ $1.50416\times10^{-11} c_1 c_2^2 k_2 + 4.81741\times10^{-12} c_2^3 k_2 + 4.39919\times10^{-8} J_1 k_2 + 8.82288\times10^{-10} c_1 J_1 k_2 + 6.43371\times10^{-12} c_1^2 J_1 k_2 +$ $4.80371\times10^{-9} c_2 J_1 k_2 + 6.17662\times10^{-11} c_1 c_2 J_1 k_2 + 3.7548\times10^{-10} c_2^2 J_1 k_2 + 6.65104\times10^{-7} J_1^2 k_2 + 4.36648\times10^{-9} c_1 J_1^2 k_2 -$ $2.68083\times10^{-9} c_2 J_1^2 k_2 - 1.06527\times10^{-7} J_1^3 k_2 + 4.19525\times10^{-7} J_2 k_2 - 2.31867\times10^{-8} c_1 J_2 k_2 - 2.15909\times10^{-9} c_1^2 J_2 k_2 -$ $1.61209\times10^{-8} c_2 J_2 k_2 + 2.23108\times10^{-9} c_1 c_2 J_2 k_2 - 2.91637\times10^{-9} c_2^2 J_2 k_2 + 3.51232\times10^{-8} J_1 J_2 k_2 - 4.90387\times10^{-8} c_1 J_1 J_2 k_2 -$ $4.91017\times10^{-8} c_2 J_1 J_2 k_2 - 1.5311\times10^{-6} J_1^2 J_2 k_2 + 1.86186\times10^{-7} J_2^2 k_2 - 1.19258\times10^{-7} c_1 J_2^2 k_2 - 1.05617\times10^{-7} c_2 J_2^2 k_2 -$ $1.53129\times10^{-8} J_1 J_2^2 k_2 - 6.05155\times10^{-7} J_2^3 k_2 - 1.70102\times10^{-10} k_1 k_2 - 1.92955\times10^{-12} c_1 k_1 k_2 + 7.92392\times10^{-14} c_1^2 k_1 k_2 -$ $3.40772\times10^{-12} c_2 k_1 k_2 - 4.85708\times10^{-14} c_1 c_2 k_1 k_2 + 1.39866\times10^{-13} c_2^2 k_1 k_2 - 9.17639\times10^{-11} J_1 k_1 k_2 - 3.92621\times10^{-13} c_1 J_1 k_1 k_2 +$ $2.1864\times10^{-13} c_2 J_1 k_1 k_2 + 2.59102\times10^{-11} J_1^2 k_1 k_2 + 5.91049\times10^{-11} J_2 k_1 k_2 + 8.37042\times10^{-12} c_1 J_2 k_1 k_2 + 7.12569\times10^{-12} c_2 J_2 k_1 k_2 +$ $3.48426\times10^{-10} J_1 J_2 k_1 k_2 + 7.41845\times10^{-10} J_2^2 k_1 k_2 + 1.78157\times10^{-14} k_1^2 k_2 - 9.44226\times10^{-17} c_1 k_1^2 k_2 - 8.62487\times10^{-17} c_2 k_1^2 k_2 -$ $4.3673\times10^{-15} J_1 k_1^2 k_2 - 4.50244\times10^{-14} J_2 k_1^2 k_2 + 2.96135\times10^{-19} k_1^3 k_2 + 2.8644\times10^{-12} k_2^2 - 1.65041\times10^{-13} c_1 k_2^2 -$ $1.52319\times10^{-14} c_1^2 k_2^2 - 1.22673\times10^{-13} c_2 k_2^2 - 6.57989\times10^{-15} c_1 c_2 k_2^2 - 2.12492\times10^{-14} c_2^2 k_2^2 - 1.97979\times10^{-11} J_1 k_2^2 +$ $4.71592\times10^{-13} c_1 J_1 k_2^2 + 3.93406\times10^{-13} c_2 J_1 k_2^2 + 8.53716\times10^{-12} J_1^2 k_2^2 - 8.25771\times10^{-12} J_2 k_2^2 + 6.15196\times10^{-12} c_1 J_2 k_2^2 +$ $5.11776\times10^{-12} c_2 J_2 k_2^2 + 9.95975\times10^{-11} J_1 J_2 k_2^2 - 3.79252\times10^{-11} J_2^2 k_2^2 - 2.11088\times10^{-15} k_1 k_2^2 - 6.76962\times10^{-17} c_1 k_1 k_2^2 -$ $5.46337\times10^{-17} c_2 k_1 k_2^2 - 2.06962\times10^{-15} J_1 k_1 k_2^2 - 2.58467\times10^{-14} J_2 k_1 k_2^2 + 2.58368\times10^{-19} k_1^2 k_2^2 + 1.81945\times10^{-17} k_2^3 -$ $5.19149\times10^{-17} c_1 k_2^3 - 4.24107\times10^{-17} c_2 k_2^3 - 1.62884\times10^{-15} J_1 k_2^3 - 1.33248\times10^{-15} J_2 k_2^3 + 4.73705\times10^{-19} k_1 k_2^3 + 1.78641\times10^{-20} k_2^4$	0.999993	0.999917

Table 5.8: Optimization results for different scenarios

Problem No	Optimization Algoritm	$\varphi_1-\varphi_2$ (rad)	J_1 ($kg.m^2$)	J_2 ($kg.m^2$)	c_1 (Nms/rad)	c_2 (Nms/rad)	k_1 (Nm/rad)	k_2 (Nm/rad)
Optimization Problem 1	DE	0.00254144	1.1	0.519683	25	15	20000	13483.9
	SA	0.00254152	1.1	0.519732	25.0001	15.0001	20000	13484.9
	RS	0.00254147	1.1	0.519712	25	15.0001	20000	13484.2
	NM	0.00254153	1.1	0.51973	25.0001	15.0001	20000	13484.8
Optimization Problem 2	DE	0.00295987	1.1	0.527402	26	16	19999	13658
	SA	0.00325731	1.10649	0.50088	26	17	19999	12870
	RS	0.00311364	1.1	0.541344	26	16	19913	14552
	NM	0.0029705	1.1	0.5134	26	16	19999	13254
Optimization Problem 3	DE	0.00254145	1.1	0.519981	25	15	20000	13500
	SA	0.00254145	1.1	0.519976	25	15	20000	13500
	RS	0.00254145	1.1	0.519981	25	15	20000	13500
	NM	0.00254145	1.1	0.519981	25	15	20000	13500
Optimization Problem 4	DE	0.00273573	1.1	0.5	25	15	20000	11000
	SA	0.00273573	1.1	0.5	25	15	20000	11000
	RS	0.00273573	1.1	0.5	25	15	20000	11000
	NM	0.00273573	1.1	0.5	25	15	20000	11000

5.5 Conclusion

In this study, dual mass flywheel parameter optimizations have been performed for transferring vibrations that are generated in the engine of a heavy vehicle to the driveline of the vehicle with the lowest amplitude. A differential equation system is solved by explicit Runge-Kutta for different input parameters values, and then a data set is obtained using full factorial design. The model is constituted using these data values by the Levenberg-Marquardt method in the Mathematica program. The improvements resulting from optimizations have been observed to be 28%. The results show that optimization plays a significant role in DMF designs, and the results obtained in the Mathematica program provide an appropriate methodological way for these to be developed. From this point of view, the relevant parameters must have a higher number of different values in order to reach the efficiency increase obtained with theoretical limits.

References

[1] Chen DM, Shi XF, Liu GQ. Design and simulation study of a new type of dual mass flywheel. Applied Mechanics and Materials 2012; 184-185: 130–133.

[2] Theodossiades S, Gnanakumarr M, Rahnejat H, Kelly P. Effect of a dual-mass flywheel on the impact-induced noise in vehicular powertrain systems. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering 2006; 220: 747–761.

[3] Zeng LP, Song LQ, Zhou JP. Design and elastic contact analysis of a friction bearing with shape constraint for promoting the torque characteristics of a dual mass flywheel. Mechanism and Machine Theory 2015; 97: 356–374.

[4] Wu C, Peng YF, Xiang C, Mao XY, Hu JH, Liu YO. Static torsional stiffness computation of circumferential arc spring dual mass flywheel. Advanced Materials Research 2015; 1065-1069: 2080–2085.

[5] Wang Y, Qin X, Huang S, Deng S. Design and analysis of a multi-stage torsional stiffness dual mass flywheel based on vibration control. Applied Acoustics 2016; 104: 172–181.

[6] Song LQ, Zeng LP, Zhang SP, Zhou JD, Niu HE. Design and analysis of a dual mass flywheel with continuously variable stiffness based on compensation principle. Mechanism and Machine Theory 2014; 79: 124–140.

[7] Kim TH, Song HL, Hwang SH, Kim HS. Analysis of dual mass flywheel using discrete arcspring model. Key Engineering Materials 2006; 326-328: 1607–1610.

[8] Schaper U, Sawodny O, MaIl T, Blessing V. Modeling and torque estimation of an automotive dual mass flywheel. The Proceedings of the American Control Conference; 2009, St. Louis, MO, USA.

[9] Meng X, Li J. Idling natural characteristic analysis of the torsion absorber with dual mass flywheel. Applied Mechanics and Materials 2011; 80-81: 860–864.

[10] Chen L, Zeng R, Jiang Z. Nonlinear dynamical model of an automotive dual mass flywheel. Advances in Mechanical Engineering 2015; 7(6): 1–11.

[11] Mahl T, Sawodny O. Modelling of an automotive dual mass flywheel. 5th IFAC: Symposium on Mechatronic Systems; 2010; Marriott Boston Cambridge, MA, USA.

[12] Ozansoy O, Tevruz T, Mugan A. Multiobjective pareto optimal design of a clutch system. International Journal of Engineering & Technology 2015; 1(1): 26–43.

[13] Yang X, Zhang T-S, Lei N-L. Experimental study on torsional vibration of transmission system under engine excitation. The Proceedings of the 3rd International Conference, Applied Mechanics and Mechanical Automation; 2017.

[14] Chen D, Xu JH, Shi XF, Ma YY. Experimental study on torsional vibration of dual mass flywheel. Advanced Materials Research 2012; 490-495: 2318–2322.

[15] Kang TS, Kauh SK, Hua KP. Development of the displacement measuring system for a dual mass flywheel in a vehicle. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering 2009; 223: 1273.

[16] Zhao G, Chen L, Jiang Z. Linear analysis for performance of dual mass flywheel with centrifugal pendulum vibration absorbers system. Telkomnika 2013; 11(5): 2371–2376.

[17] Zeng R, Chen L, Jiang Z. Frequency response analysis of damped dual mass flywheel. Applied Mechanics and Materials 2015; 724: 271–274.

[18] Pavlov N. Numerical simulation on the vibration of a test bed with engine with dual mass flywheel. International Scientific Journal, Machines Technologies, Materials 2018; 49–52.

[19] Johansson D, Karlsson K. Simulation Models of Dual Mass Flywheels (M.S. Thesis). Gothenburg: Chalmers University of Technology; 2017.

[20] Jiang ZF, Chen L. Research on the method of circumferential spring dual mass flywheel damper matching with diesel engine. Key Engineering Materials 2010; 419-420: 65–68.

[21] Cavazzuti M. Optimization Methods From Theory to Design: Scientific and Technological Aspects in Mechanics. Springer; 2013.

[22] Antony J. Design of Experiments for Engineers and Scientists. 2nd Edition. Elsevier; 2014.

[23] Vecchio R.J. Understanding Design of Experiments. Gardner Publications; 1997.

[24] Montgomery DC, Runger GC and Hubele NF, Engineering Statistics. 5th Edition. John Wiley and Sons; 2010.

[25] Antony J. Some key things industrial engineers should know about experimental design. Logistics Information Management 11 1998; (6): 386–392.

[26] Gunst RF, Mason RL. How to Construct Fractional Factorial Experiments. ASQC Quality Press; 1991.

[27] Montgomery DC. Design and Analysis of Experiments. John Wiley and Sons; 2001.

[28] Turhan, F. 1100 serisi alüminyum malzemelerde tig kaynağı ile oluşan kaynak dikiş geometrisinin optimizasyonu (M.S. Thesis). İzmir: Katip Çelebi University; 2017.

[29] Ombach J. A short introduction to stochastic optimization. Schedae Informaticae 2014; 23: 9–20.

[30] Rao RV, Savsani VJ. Mechanical Design Optimization using Advanced Optimization Techniques, Springer; 2012.

[31] Tech Target. https://whatis.techtarget.com/definition/stochastic-optimization.

[32] Yan X, Zhou Y, Wen Y, Chai X. Qualitative and quantitative integrated modeling for stochastic simulation and optimization, Journal of Applied Mathematics 2013; 1–12. doi.org/10.1155/2013/831273

[33] Li G, Zhang W. Study on Indefinite stochastic linear quadratic optimal control with inequality constraint. Journal of Applied Mathematics 2013; 1–9 doi.org/10.1155/2013/805829

[34] Aydin L, Aydin O, Artem HS, Mert A. Design of dimensionally stable composites using efficient global optimization method. Journal of materials: Design and applications 2016; 1–13.

[35] Zakaria A, Firas BI, Hossain-Lipu MS, Hannan MA. Uncertainty models for stochastic optimization in renewable energy applications. Renewable Energy 2019; 1543–1571.

[36] Niamsup P, Rajchakit G. New results on robust stability and stabilization of linear discrete-time stochastic systems with convex polytopic uncertainties. Journal of Applied Mathematics 2013; 1–10. doi.org/10.1155/2013/368259

[37] Maggioni F, Cagnolari M, Bertazzi L, Wallace SW. Stochastic optimization models for a bike-sharing problem with transshipment. European Journal of Operational Research 2019; 276(1): 272–283.

[38] Maggioni F, Cagnolari M, Bertazzi L, Wallace SW. Stochastic optimization models for a bike-sharing problem with transshipment. European Journal of Operational Research 2019; 276(1): 272–283.

[39] Khayyam H, Naebe M, Bab-Hadiashar A, Jamshidi F, Li Q, Atkiss S, and oth. Stochastic optimization models for energy management in carbonization process of carbon fiber production. Applied Energy 2015; 158: 643–655.

[40] Tifkitsis KI, Mesogitis TS, Struzziero G, Skordos AA. Stochastic multi-objective optimisation of the cure process of thick laminates. Composites Part A 112 2018; 383–394.

[41] Ozturk S, Aydin L, Celik E. A comprehensive study on slicing processes optimization of silicon ingot for photovoltaic applications. The Journal of Solar Energy Engineering 2018; 161; 109–124.

[42] Grabusts P, Musatovs J, Golenkov V. The application of simulated annealing method for optimal route detection between objects. Procedia Computer Science 2018; 149; 95–101.

[43] Aydin L, Artem HS, Design and optimization of fiber composites. Ed.: Seydibeyoglu MÖ, Mohanty AK, Misra M. Woodhead Publishing Series in Composites Science and Engineering; 2017. 299–315.

[44] Huang Z, Chen Y, An improved differential evolution algorithm based on adaptive parameter. Journal of Control Science and Engineering Volume 2013; 1–5. doi.org/10.1155/2013/462706

[45] Das S, Suganthan PN, Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 2011; 15(1): 4–35.

[46] Jujjavarapu S, Chimmiri V. Dynamic modeling and metabolic flux analysis for optimized production of rhamnolipids. Chemical Engineering Communications 2016; 203; 326–338.

[47] Arunachalam V. Optimization using Differential Evolution (M.S. Thesis). London: The University of Western Ontario; 2008.

[48] Nelder JA, Mead R. A simplex method for function minimization. Comput. J. 1965; 7(4): 308–313.

[49] Barati R. Parameter estimation of nonlinear Muskingum models using Nelder–Mead simplex algorithm. J. Hydrol. Eng. 2011; 16 (11): 946–954.

[50] Karnopp DC, Random search techniques for optimization problems. Automatica 1963; 1 (2–3): 111–121.

[51] Zabinsky ZB, Random search algorithms. Wiley Encyclopedia of Operations Research and Management Science; 2009.

[52] Wolfram Mathematica software 11.3.

[53] Ozturk S, Aydin L, Kucukdogan N, Celik E. Optimization of lapping processes of silicon wafer for photovoltaic applications. Sol. Energy 2018; 164: 1–11.

[54] Momin J, Xin-She Y, Hans-Jürgen Z. Test functions for global optimization: A comprehensive survey.Ed.: Xin-She Y, Zhihua C, Renbin X, Amir Hossein G, Mehmet K. Swarm Intelligence and Bio-Inspired Computation, Elsevier 2013; 193–222.

[55] Karaboğa D. Yapay zeka optimizasyon algoritmaları. Nobel akademik yayıncılık; 2018.

[56] Ingber L. Simulated annealing: Practice versus theory. Mathematical and Computer Modelling 1993; 18(11): 29–57.

[57] Savran M, Aydin L. Stochastic optimization of graphite-flax/epoxy hybrid laminated composite for maximum fundamental frequency and minimum cost. Engineering Structures 2018; 174: 675–687.

[58] Vo-Duy T, Ho-Huu V, Do-Thi TD, Dang-Trung H, Nguyen-Thoi T. A global numerical approach for lightweight design optimization of laminated composite plates subjected to frequency constraints. Composite Structures 2017; 159: 646–655.

[59] Nelder JA, Mead R. A simplex-method for function minimization. Computer Journal 1965; 7(4): 308–313. doi.org/10.1093/comjnl/7.4.308.

[60] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 1997; 11(4): 341–359. doi.org/10.1023/A:1008202821328.

[61] Sofroniou M, Knapp R. Advanced Numerical Differential Equation Solving In Mathematica. Wolfram Mathematica Tutorial Collection 2008; Retrieved from https:\\ library.wolfram.com.

[62] Wellin PR, Gaylord RJ, Kamin SN. An Introduction to Programming with Mathematica. Cambridge University Press; 2005.

[63] Rahnejat H, Tribology and Dynamics of Engine and Powertrain: Fundamentals, Applications and Future Trends. Woodhead Publishing Limited; 2010.

[64] Bourgois G. Dual Mass Flywheel for Torsional Vibrations Damping Parametric Study for Application in Heavy Vehicle (M.S. Thesis). Gothenburg: Chalmers University of Technology; 2016.

Appendices

Appendix A

All the results of the differential equation system for different input parameters

Case Number	J_1 $(kg.m^2)$	J_2 $(kg.m^2)$	k_1 (Nm/rad)	k_2 (Nm/rad)	c_1 (Nms/rad)	c_2 (Nms/rad)	t=1 (s)	t=2 (s)	t=3 (s)	t=4 (s)	t=5 (s)	t=6 (s)	t=7 (s)	t=8 (s)	t=9 (s)	t=10 (s)
							$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)	$\varphi_1-\varphi_2$ (rad)
1	1.8	0.6	18000	16500	30	20	0.0124297	0.0119706	0.0119829	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828	0.0119828
2	1.8	0.6	18000	16500	30	15	0.0125238	0.0116793	0.011734	0.0117305	0.0117308	0.0117307	0.0117307	0.0117307	0.0117307	0.0117307
3	1.8	0.6	18000	16500	25	20	0.0124972	0.0118133	0.0118406	0.0118398	0.0118398	0.0118398	0.0118398	0.0118398	0.0118398	0.0118398
4	1.8	0.6	18000	16500	25	15	0.0126732	0.0115129	0.0116058	0.0115985	0.011599	0.011599	0.011599	0.011599	0.011599	0.011599
5	1.8	0.6	18000	11000	30	20	0.00855831	0.0125827	0.012739	0.0127353	0.0127347	0.0127347	0.0127347	0.0127347	0.0127347	0.0127347
6	1.8	0.6	18000	11000	30	15	0.00547568	0.0121953	0.0126475	0.0126281	0.012623	0.0126228	0.0126228	0.0126228	0.0126228	0.0126228
7	1.8	0.6	18000	11000	25	20	0.00800572	0.0126194	0.0128182	0.0128122	0.0128112	0.0128112	0.0128112	0.0128112	0.0128112	0.0128112
8	1.8	0.6	18000	11000	25	15	0.00452634	0.0121873	0.0127726	0.0127429	0.0127343	0.0127339	0.0127339	0.0127339	0.0127339	0.0127339
9	1.8	0.6	18000	5500	30	20	0.0127264	0.0133834	0.0134077	0.0134086	0.0134086	0.0134086	0.0134086	0.0134086	0.0134086	0.0134086
10	1.8	0.6	18000	5500	30	15	0.0119565	0.0132474	0.0133597	0.0133695	0.0133703	0.0133704	0.0133704	0.0133704	0.0133704	0.0133704
11	1.8	0.6	18000	5500	25	20	0.012822	0.013556	0.0135855	0.0135867	0.0135867	0.0135867	0.0135867	0.0135867	0.0135867	0.0135867
12	1.8	0.6	18000	5500	25	15	0.0120128	0.0134345	0.013566	0.0135782	0.0135793	0.0135794	0.0135794	0.0135794	0.0135794	0.0135794

13	1.8	0.6	16000	16500	30	20	0.0101314	0.0148536	0.0148522	0.0148427	0.0148426	0.0148426	0.0148426	0.0148426	0.0148426	0.0148426
14	1.8	0.6	16000	16500	30	15	0.00793998	0.0147244	0.0147054	0.0146771	0.0146771	0.0146772	0.0146772	0.0146772	0.0146772	0.0146772
15	1.8	0.6	16000	16500	25	20	0.00864296	0.0148651	0.0148623	0.0148404	0.0148403	0.0148404	0.0148404	0.0148404	0.0148404	0.0148404
16	1.8	0.6	16000	16500	25	15	0.00583522	0.0147824	0.0147659	0.0147011	0.0147011	0.0147015	0.0147015	0.0147015	0.0147015	0.0147015
17	1.8	0.6	16000	11000	30	20	0.0142473	0.0154084	0.0154882	0.0154934	0.0154937	0.0154937	0.0154937	0.0154937	0.0154937	0.0154937
18	1.8	0.6	16000	11000	30	15	0.0132914	0.0151695	0.015398	0.0154231	0.0154257	0.015426	0.015426	0.015426	0.015426	0.015426
19	1.8	0.6	16000	11000	25	20	0.0140477	0.0155028	0.0156209	0.0156298	0.0156304	0.0156305	0.0156305	0.0156305	0.0156305	0.0156305
20	1.8	0.6	16000	11000	25	15	0.012959	0.0152265	0.0155472	0.0155883	0.0155933	0.0155939	0.015594	0.015594	0.015594	0.015594
21	1.8	0.6	16000	5500	30	20	0.0165461	0.0161082	0.0160381	0.0160336	0.0160333	0.0160333	0.0160333	0.0160333	0.0160333	0.0160333
22	1.8	0.6	16000	5500	30	15	0.0170628	0.0163097	0.0160518	0.0160168	0.0160132	0.0160129	0.0160129	0.0160129	0.0160129	0.0160129
23	1.8	0.6	16000	5500	25	20	0.0166935	0.0163051	0.0162314	0.0162263	0.016226	0.016226	0.016226	0.016226	0.016226	0.016226
24	1.8	0.6	16000	5500	25	15	0.0172181	0.0165473	0.0162746	0.0162348	0.0162305	0.0162301	0.01623	0.01623	0.01623	0.01623
25	1.8	0.6	12000	16500	30	20	0.0249622	0.0221907	0.0223318	0.0223266	0.0223267	0.0223267	0.0223267	0.0223267	0.0223267	0.0223267
26	1.8	0.6	12000	16500	30	15	0.0258542	0.0220152	0.0222671	0.0222554	0.0222558	0.0222558	0.0222558	0.0222558	0.0222558	0.0222558
27	1.8	0.6	12000	16500	25	20	0.0263534	0.0221615	0.0224705	0.0224537	0.0224544	0.0224544	0.0224544	0.0224544	0.0224544	0.0224544
28	1.8	0.6	12000	16500	25	15	0.0276669	0.0218947	0.0224452	0.0224074	0.0224093	0.0224093	0.0224093	0.0224093	0.0224093	0.0224093
29	1.8	0.6	12000	11000	30	20	0.025831	0.0225191	0.0227604	0.0227467	0.0227473	0.0227473	0.0227473	0.0227473	0.0227473	0.0227473
30	1.8	0.6	12000	11000	30	15	0.0274768	0.0221921	0.0227638	0.0227166	0.0227198	0.0227196	0.0227196	0.0227196	0.0227196	0.0227196
31	1.8	0.6	12000	11000	25	20	0.0269461	0.0225473	0.0229472	0.0229187	0.0229204	0.0229203	0.0229203	0.0229203	0.0229203	0.0229203
32	1.8	0.6	12000	11000	25	15	0.0290291	0.022062	0.0230029	0.022905	0.0229133	0.0229128	0.0229128	0.0229128	0.0229128	0.0229128
33	1.8	0.6	12000	5500	30	20	0.0275529	0.0230739	0.0230658	0.0230763	0.0230762	0.0230762	0.0230762	0.0230762	0.0230762	0.0230762
34	1.8	0.6	12000	5500	30	15	0.0314201	0.0230864	0.0229998	0.0230698	0.0230692	0.0230686	0.0230686	0.0230686	0.0230686	0.0230686
35	1.8	0.6	12000	5500	25	20	0.0281161	0.0232695	0.0232478	0.0232614	0.0232613	0.0232613	0.0232613	0.0232613	0.0232613	0.0232613
36	1.8	0.6	12000	5500	25	15	0.0323365	0.0233154	0.023179	0.0232695	0.023269	0.0232681	0.0232682	0.0232682	0.0232682	0.0232682
37	1.8	0.6	20000	16500	30	20	0.0138535	0.00938405	0.00946722	0.00947629	0.00947582	0.00947581	0.00947581	0.00947581	0.00947581	0.00947582
38	1.8	0.6	20000	16500	30	15	0.015885	0.00891096	0.00907548	0.00911159	0.00910917	0.00910904	0.00910906	0.00910906	0.00910906	0.00910906
39	1.8	0.6	20000	16500	25	20	0.0145221	0.00899611	0.00911149	0.00912891	0.00912787	0.00912784	0.00912785	0.00912786	0.00912786	0.00912785
40	1.8	0.6	20000	16500	25	15	0.0170731	0.00844138	0.00867171	0.00874023	0.00873477	0.0087344	0.00873447	0.00873447	0.00873447	0.00873447
41	1.8	0.6	20000	11000	30	20	0.00701096	0.0104258	0.0102638	0.0102612	0.0102618	0.0102617	0.0102617	0.0102617	0.0102617	0.0102617
42	1.8	0.6	20000	11000	30	15	0.00435187	0.0106031	0.0100874	0.0100742	0.0100794	0.0100791	0.0100791	0.0100791	0.0100791	0.0100791

43	1.8	0.6	20000	11000	25	20	0.00663518	0.0104377	0.0102294	0.0102263	0.0102272	0.0102271	0.0102271	0.0102271	0.0102271	0.0102271
44	1.8	0.6	20000	11000	25	15	0.00372216	0.0107419	0.0100848	0.0100674	0.0100756	0.0100751	0.0100751	0.0100751	0.0100751	0.0100751
45	1.8	0.6	20000	5500	30	20	0.00979314	0.0110081	0.0110722	0.011075	0.0110752	0.0110752	0.0110752	0.0110752	0.0110752	0.0110752
46	1.8	0.6	20000	5500	30	15	0.00826539	0.0106886	0.0109748	0.0110021	0.0110044	0.0110045	0.0110046	0.0110046	0.0110046	0.0110046
47	1.8	0.6	20000	5500	25	20	0.00985134	0.0111397	0.0112096	0.0112128	0.0112129	0.0112129	0.0112129	0.0112129	0.0112129	0.0112129
48	1.8	0.6	20000	5500	25	15	0.00828083	0.0108356	0.0111461	0.0111766	0.0111792	0.0111794	0.0111794	0.0111794	0.0111794	0.0111794
49	1.8	0.5	18000	16500	30	20	0.0140095	0.0112971	0.011482	0.0114727	0.0114731	0.0114731	0.0114731	0.0114731	0.0114731	0.0114731
50	1.8	0.5	18000	16500	30	15	0.014873	0.0106042	0.011032	0.0110011	0.0110028	0.0110028	0.0110028	0.0110028	0.0110028	0.0110028
51	1.8	0.5	18000	16500	25	20	0.0142257	0.010606	0.0109179	0.0108981	0.010899	0.010899	0.010899	0.010899	0.010899	0.010899
52	1.8	0.5	18000	16500	25	15	0.0153335	0.00965607	0.0103761	0.0103098	0.0103144	0.0103143	0.0103142	0.0103142	0.0103143	0.0103142
53	1.8	0.5	18000	11000	30	20	0.00784271	0.011785	0.0116949	0.0116869	0.0116872	0.0116872	0.0116872	0.0116872	0.0116872	0.0116872
54	1.8	0.5	18000	11000	30	15	0.0047995	0.0116561	0.0113882	0.0113485	0.0113516	0.0113518	0.0113518	0.0113518	0.0113518	0.0113518
55	1.8	0.5	18000	11000	25	20	0.00700147	0.0115402	0.0114136	0.0114016	0.0114022	0.0114022	0.0114022	0.0114022	0.0114022	0.0114022
56	1.8	0.5	18000	11000	25	15	0.00352869	0.0114657	0.011102	0.0110412	0.0110469	0.0110473	0.0110472	0.0110472	0.0110472	0.0110472
57	1.8	0.5	18000	5500	30	20	0.0104765	0.0123053	0.01239	0.0123925	0.0123926	0.0123926	0.0123926	0.0123926	0.0123926	0.0123926
58	1.8	0.5	18000	5500	30	15	0.0081797	0.0118335	0.0122031	0.0122259	0.0122265	0.0122264	0.0122264	0.0122264	0.0122264	0.0122264
59	1.8	0.5	18000	5500	25	20	0.0103483	0.0122912	0.012384	0.0123868	0.0123869	0.0123869	0.0123869	0.0123869	0.0123869	0.0123869
60	1.8	0.5	18000	5500	25	15	0.00795499	0.0118195	0.0122244	0.01225	0.0122506	0.0122505	0.0122505	0.0122505	0.0122505	0.0122505
61	1.8	0.5	16000	16500	30	20	0.0103641	0.0142592	0.0141057	0.014104	0.0141043	0.0141043	0.0141043	0.0141043	0.0141043	0.0141043
62	1.8	0.5	16000	16500	30	15	0.00844637	0.0140388	0.0137127	0.0137096	0.0137109	0.0137109	0.0137108	0.0137108	0.0137108	0.0137108
63	1.8	0.5	16000	16500	25	20	0.00876414	0.0139892	0.0137133	0.0137094	0.0137104	0.0137104	0.0137104	0.0137104	0.0137104	0.0137104
64	1.8	0.5	16000	16500	25	15	0.00625742	0.0138372	0.0132582	0.0132493	0.0132537	0.0132534	0.0132534	0.0132534	0.0132534	0.0132534
65	1.8	0.5	16000	11000	30	20	0.0111876	0.0143218	0.014535	0.0145441	0.0145443	0.0145443	0.0145443	0.0145443	0.0145443	0.0145443
66	1.8	0.5	16000	11000	30	15	0.00874109	0.0137099	0.0142731	0.0143115	0.0143126	0.0143125	0.0143125	0.0143125	0.0143125	0.0143125
67	1.8	0.5	16000	11000	25	20	0.0103788	0.0141252	0.0144236	0.0144384	0.0144387	0.0144387	0.0144387	0.0144387	0.0144387	0.0144387
68	1.8	0.5	16000	11000	25	15	0.00752987	0.0133718	0.0141545	0.0142174	0.0142195	0.0142193	0.0142192	0.0142192	0.0142192	0.0142192
69	1.8	0.5	16000	5500	30	20	0.0139421	0.0151418	0.0152048	0.0152078	0.0152079	0.015208	0.015208	0.015208	0.015208	0.0152079
70	1.8	0.5	16000	5500	30	15	0.0125215	0.0148138	0.0150747	0.0151011	0.0151036	0.0151038	0.0151038	0.0151038	0.0151038	0.0151038
71	1.8	0.5	16000	5500	25	20	0.0138908	0.0152174	0.0152913	0.015295	0.0152952	0.0152952	0.0152952	0.0152952	0.0152952	0.0152952
72	1.8	0.5	16000	5500	25	15	0.0123827	0.0148866	0.0151875	0.0152195	0.0152227	0.015223	0.015223	0.015223	0.015223	0.015223

73	1.8	0.5	12000	16500	30	20	0.0250237	0.0215023	0.0216333	0.0216311	0.0216311	0.0216311	0.0216311	0.0216311	0.0216311	0.0216311
74	1.8	0.5	12000	16500	30	15	0.0259408	0.0212019	0.0214214	0.0214176	0.0214174	0.0214174	0.0214174	0.0214174	0.0214174	0.0214174
75	1.8	0.5	12000	16500	25	20	0.0266257	0.0212746	0.0215668	0.0215596	0.0215594	0.0215595	0.0215595	0.0215595	0.0215595	0.0215595
76	1.8	0.5	12000	16500	25	15	0.0280645	0.0208749	0.0213677	0.0213544	0.0213536	0.0213537	0.0213537	0.0213537	0.0213537	0.0213537
77	1.8	0.5	12000	11000	30	20	0.0266234	0.0218737	0.0221139	0.022109	0.0221088	0.0221088	0.0221088	0.0221088	0.0221088	0.0221088
78	1.8	0.5	12000	11000	30	15	0.0288133	0.0214949	0.0220189	0.0220065	0.0220052	0.0220054	0.0220054	0.0220054	0.0220054	0.0220054
79	1.8	0.5	12000	11000	25	20	0.0280603	0.0218109	0.022206	0.0221963	0.0221957	0.0221958	0.0221958	0.0221958	0.0221958	0.0221958
80	1.8	0.5	12000	11000	25	15	0.0309036	0.0212792	0.022147	0.0221216	0.0221179	0.0221185	0.0221185	0.0221185	0.0221185	0.0221185
81	1.8	0.5	12000	5500	30	20	0.0258989	0.0227366	0.0225672	0.0225677	0.0225681	0.0225681	0.0225681	0.0225681	0.0225681	0.0225681
82	1.8	0.5	12000	5500	30	15	0.0287625	0.0231325	0.0225225	0.0225202	0.0225245	0.0225248	0.0225248	0.0225248	0.0225248	0.0225248
83	1.8	0.5	12000	5500	25	20	0.0262925	0.0229338	0.0227259	0.0227259	0.0227264	0.0227264	0.0227264	0.0227264	0.0227264	0.0227264
84	1.8	0.5	12000	5500	25	15	0.0294254	0.023448	0.022706	0.0226996	0.0227056	0.0227061	0.0227061	0.0227061	0.0227061	0.0227061
85	1.8	0.5	20000	16500	30	20	0.0131875	0.00958881	0.0094496	0.00945378	0.0094542	0.0094542	0.0094542	0.00945421	0.00945421	0.0094542
86	1.8	0.5	20000	16500	30	15	0.0146744	0.00929844	0.00894872	0.00896088	0.00896335	0.0089634	0.00896339	0.00896339	0.00896339	0.00896339
87	1.8	0.5	20000	16500	25	20	0.0133664	0.00897553	0.00875372	0.00876106	0.00876209	0.00876212	0.0087621	0.0087621	0.0087621	0.0087621
88	1.8	0.5	20000	16500	25	15	0.0151972	0.00865393	0.0081034	0.00812517	0.00813105	0.00813121	0.00813118	0.00813117	0.00813117	0.00813117
89	1.8	0.5	20000	11000	30	20	0.00812593	0.00938151	0.00924551	0.00925406	0.00925365	0.00925366	0.00925366	0.00925367	0.00925367	0.00925367
90	1.8	0.5	20000	11000	30	15	0.00683631	0.00919253	0.00875664	0.00880555	0.00880131	0.00880159	0.00880159	0.00880158	0.00880159	0.00880158
91	1.8	0.5	20000	11000	25	20	0.00755513	0.00890096	0.00873111	0.0087434	0.00874273	0.00874275	0.00874275	0.00874275	0.00874275	0.00874275
92	1.8	0.5	20000	11000	25	15	0.00610541	0.00870501	0.00815137	0.00822228	0.00821525	0.0082158	0.00821577	0.00821577	0.00821577	0.00821577
93	1.8	0.5	20000	5500	30	20	0.00791903	0.00983066	0.00988076	0.00988028	0.00988019	0.0098802	0.00988019	0.0098802	0.00988019	0.0098802
94	1.8	0.5	20000	5500	30	15	0.00535937	0.0093927	0.00962855	0.00962298	0.00962122	0.00962112	0.00962112	0.00962112	0.00962112	0.00962112
95	1.8	0.5	20000	5500	25	20	0.00767217	0.00966789	0.00972052	0.00971987	0.00971979	0.00971979	0.00971979	0.00971979	0.00971978	0.00971978
96	1.8	0.5	20000	5500	25	15	0.00502585	0.0092334	0.00948349	0.00947655	0.00947447	0.00947436	0.00947436	0.00947436	0.00947436	0.00947436
97	1.7	0.6	18000	16500	30	20	0.0153514	0.0116813	0.0116209	0.0116267	0.0116269	0.0116269	0.0116269	0.0116269	0.0116269	0.0116269
98	1.7	0.6	18000	16500	30	15	0.0170106	0.0114974	0.0113314	0.0113507	0.0113517	0.0113516	0.0113516	0.0113516	0.0113516	0.0113516
99	1.7	0.6	18000	16500	25	20	0.0162023	0.0115482	0.0114417	0.0114536	0.011454	0.011454	0.011454	0.011454	0.011454	0.011454
100	1.7	0.6	18000	16500	25	15	0.0184136	0.0114281	0.0111446	0.0111844	0.0111872	0.011187	0.0111869	0.0111869	0.0111869	0.0111869
101	1.7	0.6	18000	11000	30	20	0.0109929	0.012546	0.0124036	0.0124111	0.0124108	0.0124108	0.0124108	0.0124108	0.0124108	0.0124108
102	1.7	0.6	18000	11000	30	15	0.00984224	0.0126947	0.0122455	0.0122879	0.012285	0.0122851	0.0122851	0.0122851	0.0122851	0.0122851

103	1.7	0.6	18000	11000	25	20	0.0109387	0.0126536	0.0124683	0.0124797	0.0124792	0.0124792	0.0124792	0.0124792	0.0124792	0.0124792
104	1.7	0.6	18000	11000	25	15	0.00969174	0.0129224	0.0123295	0.0123943	0.0123892	0.0123894	0.0123894	0.0123894	0.0123894	0.0123894
105	1.7	0.6	18000	5500	30	20	0.0110994	0.0130743	0.0131264	0.013126	0.013126	0.013126	0.013126	0.013126	0.013126	0.013126
106	1.7	0.6	18000	5500	30	15	0.00868371	0.0128439	0.0130872	0.0130824	0.0130808	0.0130807	0.0130807	0.0130807	0.0130807	0.0130807
107	1.7	0.6	18000	5500	25	20	0.0111821	0.0132553	0.0133105	0.01331	0.0133099	0.0133099	0.0133099	0.0133099	0.0133099	0.0133099
108	1.7	0.6	18000	5500	25	15	0.00868295	0.0130457	0.013306	0.0132998	0.0132978	0.0132977	0.0132977	0.0132977	0.0132977	0.0132977
109	1.7	0.6	16000	16500	30	20	0.0154051	0.0144993	0.0145462	0.0145439	0.014544	0.014544	0.014544	0.014544	0.014544	0.014544
110	1.7	0.6	16000	16500	30	15	0.0157645	0.0142477	0.0143711	0.0143621	0.0143627	0.0143627	0.0143627	0.0143627	0.0143627	0.0143627
111	1.7	0.6	16000	16500	25	20	0.0157469	0.0144454	0.0145341	0.0145284	0.0145288	0.0145287	0.0145287	0.0145287	0.0145287	0.0145287
112	1.7	0.6	16000	16500	25	15	0.016279	0.014171	0.0143935	0.0143722	0.0143741	0.0143739	0.0143739	0.014374	0.0143739	0.0143739
113	1.7	0.6	16000	11000	30	20	0.0112291	0.0152425	0.0152373	0.0152292	0.0152291	0.0152291	0.0152291	0.0152291	0.0152291	0.0152291
114	1.7	0.6	16000	11000	30	15	0.00840302	0.0152141	0.015192	0.0151525	0.0151525	0.0151528	0.0151528	0.0151528	0.0151528	0.0151528
115	1.7	0.6	16000	11000	25	20	0.010663	0.0153949	0.0153813	0.015368	0.015368	0.015368	0.015368	0.015368	0.015368	0.015368
116	1.7	0.6	16000	11000	25	15	0.00737924	0.0154225	0.015389	0.0153241	0.0153242	0.0153247	0.0153246	0.0153246	0.0153246	0.0153246
117	1.7	0.6	16000	5500	30	20	0.0138937	0.0157138	0.0158003	0.0158031	0.0158032	0.0158032	0.0158032	0.0158032	0.0158032	0.0158032
118	1.7	0.6	16000	5500	30	15	0.0117632	0.0153789	0.015752	0.0157776	0.0157787	0.0157786	0.0157786	0.0157786	0.0157786	0.0157786
119	1.7	0.6	16000	5500	25	20	0.0139514	0.0159062	0.0160028	0.0160061	0.0160062	0.0160062	0.0160062	0.0160062	0.0160062	0.0160062
120	1.7	0.6	16000	5500	25	15	0.0117014	0.0155617	0.0159773	0.0160068	0.0160079	0.0160079	0.0160079	0.0160079	0.0160079	0.0160079
121	1.7	0.6	12000	16500	30	20	0.0249331	0.0222237	0.0221269	0.0221262	0.0221263	0.0221263	0.0221263	0.0221263	0.0221263	0.0221263
122	1.7	0.6	12000	16500	30	15	0.0256791	0.0222258	0.0220508	0.0220484	0.0220486	0.0220486	0.0220486	0.0220486	0.0220486	0.0220486
123	1.7	0.6	12000	16500	25	20	0.0264254	0.0224758	0.0222604	0.0222581	0.0222584	0.0222584	0.0222584	0.0222584	0.0222584	0.0222584
124	1.7	0.6	12000	16500	25	15	0.0276423	0.022601	0.0222146	0.022207	0.0222078	0.0222079	0.0222079	0.0222079	0.0222079	0.0222079
125	1.7	0.6	12000	11000	30	20	0.0269521	0.0226877	0.0225661	0.0225711	0.0225714	0.0225714	0.0225714	0.0225714	0.0225714	0.0225714
126	1.7	0.6	12000	11000	30	15	0.0291556	0.0228357	0.0225237	0.0225386	0.0225403	0.0225403	0.0225403	0.0225403	0.0225403	0.0225403
127	1.7	0.6	12000	11000	25	20	0.0283125	0.0229514	0.0227436	0.0227536	0.0227544	0.0227544	0.0227544	0.0227544	0.0227544	0.0227544
128	1.7	0.6	12000	11000	25	15	0.0311692	0.0232316	0.0227101	0.0227402	0.0227449	0.0227449	0.0227449	0.0227449	0.0227449	0.0227449
129	1.7	0.6	12000	5500	30	20	0.0244104	0.0230587	0.0229275	0.0229214	0.0229213	0.0229213	0.0229213	0.0229213	0.0229213	0.0229213
130	1.7	0.6	12000	5500	30	15	0.0257633	0.0234175	0.0229584	0.0229147	0.0229121	0.022912	0.022912	0.022912	0.022912	0.022912
131	1.7	0.6	12000	5500	25	20	0.0246328	0.0232787	0.0231271	0.0231191	0.0231188	0.0231188	0.0231188	0.0231188	0.0231188	0.0231188
132	1.7	0.6	12000	5500	25	15	0.0260744	0.0237172	0.0231858	0.0231291	0.0231252	0.0231251	0.0231251	0.0231251	0.0231252	0.0231251

133	1.7	0.6	20000	16500	30	20	0.00897333	0.00908121	0.00906773	0.00906637	0.00906627	0.00906627	0.00906627	0.00906627	0.00906627	0.00906627
134	1.7	0.6	20000	16500	30	15	0.00842319	0.00868747	0.00867251	0.0086692	0.00866882	0.00866878	0.00866878	0.00866878	0.00866878	0.00866878
135	1.7	0.6	20000	16500	25	20	0.00847161	0.00868408	0.00866931	0.00866713	0.00866693	0.00866692	0.00866692	0.00866691	0.00866692	0.00866693
136	1.7	0.6	20000	16500	25	15	0.00785823	0.00825812	0.00824136	0.0082356	0.00823474	0.00823464	0.00823463	0.00823462	0.00823462	0.00823462
137	1.7	0.6	20000	11000	30	20	0.0113142	0.00977478	0.0098753	0.00986978	0.00987005	0.00987004	0.00987004	0.00987004	0.00987003	0.00987003
138	1.7	0.6	20000	11000	30	15	0.012335	0.00933615	0.00969921	0.00966286	0.00966609	0.00966583	0.00966585	0.00966585	0.00966585	0.00966585
139	1.7	0.6	20000	11000	25	20	0.0115082	0.00968543	0.0098181	0.00980998	0.00981043	0.00981041	0.00981041	0.00981041	0.00981041	0.00981041
140	1.7	0.6	20000	11000	25	15	0.0127353	0.00920957	0.00968415	0.00963115	0.00963641	0.00963594	0.00963598	0.00963597	0.00963597	0.00963597
141	1.7	0.6	20000	5500	30	20	0.00894994	0.010722	0.0107295	0.0107278	0.0107278	0.0107278	0.0107278	0.0107278	0.0107278	0.0107278
142	1.7	0.6	20000	5500	30	15	0.00668582	0.0106266	0.0106651	0.0106459	0.0106452	0.0106453	0.0106453	0.0106453	0.0106453	0.0106453
143	1.7	0.6	20000	5500	25	20	0.00902388	0.0108578	0.0108643	0.0108623	0.0108623	0.0108623	0.0108623	0.0108623	0.0108623	0.0108623
144	1.7	0.6	20000	5500	25	15	0.00672078	0.0108057	0.0108412	0.0108196	0.0108189	0.010819	0.010819	0.010819	0.010819	0.010819
145	1.7	0.5	18000	16500	30	20	0.0135678	0.0112809	0.0111466	0.0111435	0.0111435	0.0111435	0.0111435	0.0111435	0.0111435	0.0111435
146	1.7	0.5	18000	16500	30	15	0.0142547	0.0109684	0.0106606	0.0106484	0.0106485	0.0106486	0.0106486	0.0106486	0.0106486	0.0106486
147	1.7	0.5	18000	16500	25	20	0.0136102	0.0107525	0.0105275	0.0105205	0.0105206	0.0105207	0.0105207	0.0105207	0.0105207	0.0105207
148	1.7	0.5	18000	16500	25	15	0.0145375	0.0104406	0.00992826	0.00990134	0.00990166	0.00990185	0.00990187	0.00990187	0.00990186	0.00990187
149	1.7	0.5	18000	11000	30	20	0.0122532	0.0112766	0.0113296	0.0113269	0.011327	0.011327	0.011327	0.011327	0.011327	0.011327
150	1.7	0.5	18000	11000	30	15	0.0126564	0.0107916	0.0109811	0.0109635	0.0109651	0.0109649	0.0109649	0.0109649	0.0109649	0.0109649
151	1.7	0.5	18000	11000	25	20	0.012166	0.0109276	0.0110062	0.0110015	0.0110018	0.0110018	0.0110018	0.0110018	0.0110018	0.0110018
152	1.7	0.5	18000	11000	25	15	0.0126848	0.0103693	0.0106404	0.0106113	0.0106142	0.010614	0.010614	0.010614	0.010614	0.010614
153	1.7	0.5	18000	5500	30	20	0.0103135	0.0120865	0.012053	0.0120516	0.0120517	0.0120517	0.0120517	0.0120517	0.0120517	0.0120517
154	1.7	0.5	18000	5500	30	15	0.00803979	0.0120457	0.0118809	0.0118663	0.0118674	0.0118675	0.0118675	0.0118675	0.0118675	0.0118675
155	1.7	0.5	18000	5500	25	20	0.0102267	0.012069	0.0120302	0.0120287	0.0120288	0.0120288	0.0120288	0.0120288	0.0120288	0.0120288
156	1.7	0.5	18000	5500	25	15	0.00789452	0.0120771	0.0118892	0.0118727	0.0118742	0.0118742	0.0118742	0.0118742	0.0118742	0.0118742
157	1.7	0.5	16000	16500	30	20	0.0162161	0.0136705	0.0138004	0.0137959	0.013796	0.013796	0.013796	0.013796	0.013796	0.013796
158	1.7	0.5	16000	16500	30	15	0.0170134	0.013106	0.0133911	0.0133775	0.0133779	0.0133779	0.0133779	0.0133779	0.0133779	0.0133779
159	1.7	0.5	16000	16500	25	20	0.0166858	0.0131325	0.0133738	0.0133626	0.0133629	0.0133629	0.013363	0.0133629	0.0133629	0.0133629
160	1.7	0.5	16000	16500	25	15	0.017813	0.0123771	0.0129066	0.0128726	0.0128739	0.0128739	0.0128739	0.0128739	0.0128739	0.0128739
161	1.7	0.5	16000	11000	30	20	0.0114677	0.014387	0.0142326	0.0142344	0.0142346	0.0142345	0.0142345	0.0142345	0.0142345	0.0142345
162	1.7	0.5	16000	11000	30	15	0.00936841	0.0144245	0.0139712	0.0139817	0.0139831	0.013983	0.013983	0.013983	0.013983	0.013983

163	1.7	0.5	16000	11000	25	20	0.0108777	0.014326	0.0141032	0.0141066	0.014107	0.0141069	0.0141069	0.0141069	0.0141069	0.0141069
164	1.7	0.5	16000	11000	25	15	0.00844022	0.014497	0.0138447	0.0138632	0.0138661	0.0138657	0.0138657	0.0138657	0.0138657	0.0138657
165	1.7	0.5	16000	5500	30	20	0.0126044	0.0149019	0.0149292	0.0149272	0.0149272	0.0149272	0.0149272	0.0149272	0.0149272	0.0149272
166	1.7	0.5	16000	5500	30	15	0.00986678	0.0147071	0.0148324	0.0148128	0.0148114	0.0148114	0.0148114	0.0148114	0.0148114	0.0148114
167	1.7	0.5	16000	5500	25	20	0.012547	0.0149853	0.0150131	0.0150107	0.0150106	0.0150106	0.0150106	0.0150106	0.0150106	0.0150106
168	1.7	0.5	16000	5500	25	15	0.00968074	0.0148214	0.0149538	0.0149298	0.0149281	0.0149281	0.0149281	0.0149281	0.0149281	0.0149281
169	1.7	0.5	12000	16500	30	20	0.0230858	0.0214819	0.0214015	0.0213993	0.0213993	0.0213993	0.0213993	0.0213993	0.0213993	0.0213993
170	1.7	0.5	12000	16500	30	15	0.0232715	0.0213108	0.0211757	0.0211704	0.0211703	0.0211703	0.0211703	0.0211703	0.0211703	0.0211703
171	1.7	0.5	12000	16500	25	20	0.0238387	0.0215049	0.0213229	0.0213153	0.0213151	0.0213151	0.0213151	0.0213151	0.0213151	0.0213151
172	1.7	0.5	12000	16500	25	15	0.0242929	0.0214188	0.0211116	0.0210936	0.021093	0.021093	0.021093	0.021093	0.021093	0.021093
173	1.7	0.5	12000	11000	30	20	0.0247359	0.0220677	0.0219011	0.0218968	0.0218969	0.0218969	0.0218969	0.0218969	0.0218969	0.0218969
174	1.7	0.5	12000	11000	30	15	0.0259694	0.0221801	0.0218012	0.0217844	0.0217844	0.0217844	0.0217844	0.0217844	0.0217844	0.0217844
175	1.7	0.5	12000	11000	25	20	0.0255771	0.0222719	0.0219938	0.0219842	0.0219842	0.0219843	0.0219843	0.0219843	0.0219843	0.0219843
176	1.7	0.5	12000	11000	25	15	0.0272571	0.0225648	0.0219358	0.0218989	0.0218988	0.021899	0.021899	0.021899	0.021899	0.021899
177	1.7	0.5	12000	5500	30	20	0.0217618	0.0223712	0.0223816	0.0223815	0.0223814	0.0223814	0.0223814	0.0223814	0.0223814	0.0223814
178	1.7	0.5	12000	5500	30	15	0.0210998	0.0222746	0.0223315	0.0223334	0.0223334	0.0223334	0.0223334	0.0223333	0.0223333	0.0223333
179	1.7	0.5	12000	5500	25	20	0.0217501	0.0225278	0.0225475	0.0225477	0.0225477	0.0225477	0.0225477	0.0225477	0.0225477	0.0225477
180	1.7	0.5	12000	5500	25	15	0.0209971	0.0224335	0.0225196	0.022524	0.0225241	0.0225241	0.0225241	0.0225241	0.0225241	0.0225241
181	1.7	0.5	20000	16500	30	20	0.00700519	0.0089959	0.00911887	0.00912425	0.00912442	0.00912442	0.00912443	0.00912442	0.00912442	0.00912442
182	1.7	0.5	20000	16500	30	15	0.00518652	0.00828665	0.00859306	0.00861382	0.00861483	0.00861486	0.00861486	0.00861486	0.00861486	0.00861486
183	1.7	0.5	20000	16500	25	20	0.0056634	0.00817908	0.00837266	0.00838318	0.00838361	0.00838362	0.00838362	0.00838362	0.00838362	0.00838362
184	1.7	0.5	20000	16500	25	15	0.00337211	0.00720834	0.007684	0.0077245	0.00772698	0.00772706	0.00772705	0.00772705	0.00772705	0.00772705
185	1.7	0.5	20000	11000	30	20	0.01167	0.00875447	0.00885406	0.00885457	0.00885438	0.00885439	0.00885439	0.00885439	0.00885439	0.00885439
186	1.7	0.5	20000	11000	30	15	0.013541	0.00804348	0.00836717	0.00837323	0.00837093	0.00837108	0.00837108	0.00837108	0.00837108	0.00837108
187	1.7	0.5	20000	11000	25	20	0.0115106	0.00815946	0.00828414	0.00828527	0.00828495	0.00828496	0.00828496	0.00828496	0.00828496	0.00828496
188	1.7	0.5	20000	11000	25	15	0.0136234	0.00730044	0.00770875	0.00771933	0.00771543	0.00771569	0.0077157	0.0077157	0.0077157	0.0077157
189	1.7	0.5	20000	5500	30	20	0.00838947	0.00952215	0.00947449	0.00947496	0.00947498	0.00947498	0.00947498	0.00947498	0.00947498	0.00947498
190	1.7	0.5	20000	5500	30	15	0.00673122	0.00943467	0.00918259	0.00918894	0.00918959	0.00918952	0.00918953	0.00918952	0.00918952	0.00918952
191	1.7	0.5	20000	5500	25	20	0.00817474	0.00932843	0.0092765	0.0092771	0.00927713	0.00927713	0.00927713	0.00927713	0.00927713	0.00927713
192	1.7	0.5	20000	5500	25	15	0.00649257	0.00926802	0.00899323	0.00900091	0.00900164	0.00900155	0.00900154	0.00900155	0.00900155	0.00900155

193	1.6	0.6	18000	16500	30	20	0.00962019	0.0111298	0.0112133	0.0112168	0.011217	0.011217	0.011217	0.011217	0.011217	0.011217
194	1.6	0.6	18000	16500	30	15	0.0083103	0.0106902	0.0109001	0.0109139	0.0109147	0.0109147	0.0109147	0.0109147	0.0109147	0.0109147
195	1.6	0.6	18000	16500	25	20	0.00888304	0.0108583	0.0109985	0.0110061	0.0110065	0.0110065	0.0110065	0.0110065	0.0110065	0.0110065
196	1.6	0.6	18000	16500	25	15	0.00729847	0.010332	0.0106768	0.0107062	0.0107082	0.0107084	0.0107084	0.0107084	0.0107084	0.0107084
197	1.6	0.6	18000	11000	30	20	0.0147444	0.0119386	0.0120339	0.0120338	0.0120336	0.0120337	0.0120337	0.0120337	0.0120337	0.0120337
198	1.6	0.6	18000	11000	30	15	0.0168573	0.0115829	0.0118906	0.0118927	0.0118911	0.0118912	0.0118912	0.0118912	0.0118912	0.0118912
199	1.6	0.6	18000	11000	25	20	0.0152517	0.0119661	0.01209	0.0120902	0.01209	0.01209	0.01209	0.01209	0.01209	0.01209
200	1.6	0.6	18000	11000	25	15	0.017757	0.0115795	0.0119826	0.0119873	0.0119843	0.0119845	0.0119845	0.0119845	0.0119845	0.0119845
201	1.6	0.6	18000	5500	30	20	0.0116852	0.0128416	0.0127938	0.0127943	0.0127943	0.0127943	0.0127943	0.0127943	0.0127943	0.0127943
202	1.6	0.6	18000	5500	30	15	0.0102353	0.012986	0.012733	0.0127397	0.0127402	0.0127402	0.0127402	0.0127402	0.0127402	0.0127402
203	1.6	0.6	18000	5500	25	20	0.0118543	0.013035	0.0129824	0.012983	0.012983	0.012983	0.012983	0.012983	0.012983	0.012983
204	1.6	0.6	18000	5500	25	15	0.0104011	0.0132342	0.0129557	0.0129639	0.0129646	0.0129645	0.0129645	0.0129645	0.0129645	0.0129645
205	1.6	0.6	16000	16500	30	20	0.0170131	0.0142749	0.0141979	0.0141991	0.0141992	0.0141992	0.0141992	0.0141992	0.0141992	0.0141992
206	1.6	0.6	16000	16500	30	15	0.0181656	0.0141806	0.0139959	0.0139986	0.0139991	0.0139991	0.0139991	0.0139991	0.0139991	0.0139991
207	1.6	0.6	16000	16500	25	20	0.017934	0.0143067	0.0141637	0.0141663	0.0141667	0.0141667	0.0141667	0.0141667	0.0141667	0.0141667
208	1.6	0.6	16000	16500	25	15	0.0195969	0.0143224	0.0139846	0.013991	0.0139927	0.0139928	0.0139928	0.0139928	0.0139928	0.0139928
209	1.6	0.6	16000	11000	30	20	0.0152734	0.0149175	0.0149209	0.0149211	0.0149211	0.0149211	0.0149211	0.0149211	0.0149211	0.0149211
210	1.6	0.6	16000	11000	30	15	0.0155378	0.0148041	0.0148352	0.0148343	0.0148343	0.0148343	0.0148343	0.0148343	0.0148343	0.0148343
211	1.6	0.6	16000	11000	25	20	0.0155731	0.0150494	0.0150607	0.0150607	0.0150607	0.0150607	0.0150607	0.0150607	0.0150607	0.0150607
212	1.6	0.6	16000	11000	25	15	0.0159504	0.0149568	0.0150112	0.0150088	0.0150088	0.0150088	0.0150088	0.0150088	0.0150088	0.0150088
213	1.6	0.6	16000	5500	30	20	0.0136792	0.0155648	0.0155353	0.0155338	0.0155338	0.0155338	0.0155338	0.0155338	0.0155338	0.0155338
214	1.6	0.6	16000	5500	30	15	0.0114422	0.0156637	0.0155189	0.015503	0.015504	0.015504	0.015504	0.015504	0.015504	0.015504
215	1.6	0.6	16000	5500	25	20	0.0138108	0.0157844	0.0157493	0.0157475	0.0157476	0.0157476	0.0157476	0.0157476	0.0157476	0.0157476
216	1.6	0.6	16000	5500	25	15	0.0114923	0.0159324	0.0157632	0.0157449	0.0157461	0.0157462	0.0157462	0.0157462	0.0157462	0.0157462
217	1.6	0.6	12000	16500	30	20	0.0196462	0.021827	0.0218938	0.0218949	0.0218949	0.0218949	0.0218949	0.0218949	0.0218949	0.0218949
218	1.6	0.6	12000	16500	30	15	0.0186596	0.021685	0.0218067	0.0218092	0.0218091	0.0218091	0.0218091	0.0218091	0.0218091	0.0218091
219	1.6	0.6	12000	16500	25	20	0.0185946	0.021874	0.0220273	0.0220313	0.0220313	0.0220313	0.0220313	0.0220313	0.0220313	0.0220313
220	1.6	0.6	12000	16500	25	15	0.0172032	0.0216855	0.0219653	0.0219743	0.0219743	0.0219743	0.0219743	0.0219743	0.0219743	0.0219743
221	1.6	0.6	12000	11000	30	20	0.0210416	0.0223001	0.0223644	0.0223673	0.0223674	0.0223674	0.0223674	0.0223674	0.0223674	0.0223674
222	1.6	0.6	12000	11000	30	15	0.0200897	0.0221453	0.0223188	0.0223313	0.0223321	0.0223321	0.0223321	0.0223321	0.0223321	0.0223321

223	1.6	0.6	12000	11000	25	20	0.0207496	0.0224436	0.0225546	0.022561	0.0225613	0.0225613	0.0225613	0.0225613	0.0225613	0.0225613
224	1.6	0.6	12000	11000	25	15	0.0195793	0.0222349	0.0225205	0.0225468	0.022549	0.0225492	0.0225492	0.0225492	0.0225492	0.0225492
225	1.6	0.6	12000	5500	30	20	0.0207638	0.0226447	0.0227372	0.0227408	0.0227409	0.0227409	0.0227409	0.0227409	0.0227409	0.0227409
226	1.6	0.6	12000	5500	30	15	0.0187332	0.0223318	0.0226992	0.0227277	0.0227295	0.0227296	0.0227296	0.0227296	0.0227296	0.0227296
227	1.6	0.6	12000	5500	25	20	0.0207107	0.0228348	0.0229476	0.0229523	0.0229525	0.0229525	0.0229525	0.0229525	0.0229525	0.0229525
228	1.6	0.6	12000	5500	25	15	0.0184642	0.022475	0.0229183	0.0229554	0.0229579	0.022958	0.022958	0.022958	0.022958	0.022958
229	1.6	0.6	20000	16500	30	20	0.0057525	0.00865169	0.00860053	0.00859733	0.00859744	0.00859744	0.00859744	0.00859744	0.00859744	0.00859744
230	1.6	0.6	20000	16500	30	15	0.00354938	0.00832291	0.00817735	0.008164	0.00816487	0.00816488	0.00816488	0.00816489	0.00816489	0.00816488
231	1.6	0.6	20000	16500	25	20	0.00460294	0.00822393	0.00814134	0.00813518	0.00813546	0.00813546	0.00813546	0.00813546	0.00813546	0.00813547
232	1.6	0.6	20000	16500	25	15	0.00191278	0.00790794	0.00768206	0.00765568	0.00765779	0.00765787	0.00765786	0.00765786	0.00765786	0.00765786
233	1.6	0.6	20000	11000	30	20	0.0117827	0.00947945	0.0094127	0.00941457	0.0094147	0.0094147	0.0094147	0.0094147	0.0094147	0.0094147
234	1.6	0.6	20000	11000	30	15	0.013649	0.00942701	0.00917256	0.00918307	0.00918476	0.00918478	0.00918478	0.00918477	0.00918477	0.00918478
235	1.6	0.6	20000	11000	25	20	0.0119645	0.00940774	0.00931921	0.00932167	0.00932188	0.00932189	0.00932188	0.00932188	0.00932188	0.00932189
236	1.6	0.6	20000	11000	25	15	0.0141187	0.00943618	0.00910326	0.00911724	0.00912	0.00912004	0.00912002	0.00912002	0.00912002	0.00912002
237	1.6	0.6	20000	5500	30	20	0.00982046	0.0103523	0.0103186	0.0103197	0.0103197	0.0103197	0.0103197	0.0103197	0.0103197	0.0103197
238	1.6	0.6	20000	5500	30	15	0.00907012	0.0103959	0.0102094	0.0102231	0.0102225	0.0102225	0.0102225	0.0102225	0.0102225	0.0102225
239	1.6	0.6	20000	5500	25	20	0.00995827	0.0104817	0.0104461	0.0104473	0.0104472	0.0104472	0.0104472	0.0104472	0.0104472	0.0104472
240	1.6	0.6	20000	5500	25	15	0.00925094	0.0105752	0.0103769	0.0103922	0.0103915	0.0103915	0.0103915	0.0103915	0.0103915	0.0103915
241	1.6	0.5	18000	16500	30	20	0.00812018	0.0106959	0.0107723	0.0107722	0.0107721	0.0107721	0.0107721	0.0107721	0.0107721	0.0107721
242	1.6	0.5	18000	16500	30	15	0.00605778	0.010072	0.0102522	0.010251	0.0102505	0.0102505	0.0102505	0.0102505	0.0102505	0.0102505
243	1.6	0.5	18000	16500	25	20	0.006586	0.00996366	0.0100935	0.0100932	0.010093	0.010093	0.010093	0.010093	0.010093	0.010093
244	1.6	0.5	18000	16500	25	15	0.00391658	0.00913017	0.00944101	0.0094386	0.00943707	0.00943698	0.00943698	0.00943698	0.00943698	0.00943698
245	1.6	0.5	18000	11000	30	20	0.0135153	0.0109575	0.010912	0.0109147	0.0109148	0.0109148	0.0109148	0.0109148	0.0109148	0.0109148
246	1.6	0.5	18000	11000	30	15	0.0152531	0.0106777	0.0105061	0.0105214	0.0105224	0.0105223	0.0105223	0.0105223	0.0105223	0.0105223
247	1.6	0.5	18000	11000	25	20	0.0135502	0.0106032	0.0105364	0.0105405	0.0105407	0.0105407	0.0105407	0.0105407	0.0105407	0.0105407
248	1.6	0.5	18000	11000	25	15	0.0156038	0.0103347	0.0100895	0.0101129	0.0101148	0.0101148	0.0101147	0.0101147	0.0101147	0.0101147
249	1.6	0.5	18000	5500	30	20	0.0117575	0.0116637	0.011656	0.0116565	0.0116565	0.0116565	0.0116565	0.0116565	0.0116565	0.0116565
250	1.6	0.5	18000	5500	30	15	0.0116911	0.0114811	0.0114466	0.0114513	0.0114509	0.0114509	0.0114509	0.0114509	0.0114509	0.0114509
251	1.6	0.5	18000	5500	25	20	0.0117556	0.011617	0.0116102	0.0116107	0.0116107	0.0116107	0.0116107	0.0116107	0.0116107	0.0116107
252	1.6	0.5	18000	5500	25	15	0.0117587	0.0114611	0.0114295	0.0114346	0.0114341	0.0114341	0.0114341	0.0114341	0.0114341	0.0114341

253	1.6	0.5	16000	16500	30	20	0.0149159	0.0135272	0.0134485	0.0134461	0.013446	0.013446	0.013446	0.013446	0.013446	0.013446
254	1.6	0.5	16000	16500	30	15	0.0150921	0.0131773	0.0130091	0.0130008	0.0130005	0.0130005	0.0130005	0.0130005	0.0130005	0.0130005
255	1.6	0.5	16000	16500	25	20	0.0149304	0.0131185	0.0129733	0.0129671	0.0129669	0.0129669	0.0129669	0.0129669	0.0129669	0.0129669
256	1.6	0.5	16000	16500	25	15	0.0152817	0.0127744	0.0124636	0.0124426	0.0124417	0.0124417	0.0124417	0.0124417	0.0124417	0.0124417
257	1.6	0.5	16000	11000	30	20	0.016033	0.0137731	0.0138821	0.0138785	0.0138786	0.0138786	0.0138786	0.0138786	0.0138786	0.0138786
258	1.6	0.5	16000	11000	30	15	0.0174294	0.0132784	0.0136221	0.0136035	0.0136041	0.0136041	0.0136041	0.0136041	0.0136041	0.0136041
259	1.6	0.5	16000	11000	25	20	0.0163771	0.0135699	0.0137294	0.0137233	0.0137234	0.0137234	0.0137234	0.0137234	0.0137234	0.0137234
260	1.6	0.5	16000	11000	25	15	0.0181244	0.0129847	0.0134874	0.0134554	0.0134565	0.0134565	0.0134565	0.0134565	0.0134565	0.0134565
261	1.6	0.5	16000	5500	30	20	0.0138461	0.0146481	0.0146	0.0146014	0.0146014	0.0146014	0.0146014	0.0146014	0.0146014	0.0146014
262	1.6	0.5	16000	5500	30	15	0.0128015	0.0147002	0.0144563	0.0144724	0.0144718	0.0144718	0.0144718	0.0144718	0.0144718	0.0144718
263	1.6	0.5	16000	5500	25	20	0.0139243	0.0147302	0.0146768	0.0146785	0.0146784	0.0146784	0.0146784	0.0146784	0.0146784	0.0146784
264	1.6	0.5	16000	5500	25	15	0.0128928	0.0148376	0.0145647	0.0145842	0.0145834	0.0145834	0.0145834	0.0145834	0.0145834	0.0145834
265	1.6	0.5	12000	16500	30	20	0.0185926	0.0210956	0.0211349	0.0211344	0.0211344	0.0211344	0.0211344	0.0211344	0.0211344	0.0211344
266	1.6	0.5	12000	16500	30	15	0.017412	0.020824	0.0208894	0.020888	0.0208879	0.0208879	0.0208879	0.0208879	0.0208879	0.0208879
267	1.6	0.5	12000	16500	25	20	0.0170957	0.0209419	0.0210365	0.021035	0.0210348	0.0210348	0.0210348	0.0210348	0.0210348	0.0210348
268	1.6	0.5	12000	16500	25	15	0.015425	0.020636	0.0207988	0.0207942	0.0207937	0.0207937	0.0207937	0.0207937	0.0207937	0.0207937
269	1.6	0.5	12000	11000	30	20	0.0189294	0.0215375	0.02165	0.0216529	0.0216529	0.0216529	0.0216529	0.0216529	0.0216529	0.0216529
270	1.6	0.5	12000	11000	30	15	0.0171631	0.0212471	0.0215197	0.0215298	0.0215299	0.0215299	0.0215299	0.0215299	0.0215299	0.0215299
271	1.6	0.5	12000	11000	25	20	0.0181012	0.0215406	0.0217333	0.0217397	0.0217398	0.0217397	0.0217397	0.0217397	0.0217397	0.0217397
272	1.6	0.5	12000	11000	25	15	0.0158709	0.0211579	0.0216221	0.021645	0.0216452	0.0216451	0.0216451	0.0216451	0.0216451	0.0216451
273	1.6	0.5	12000	5500	30	20	0.0194043	0.0221033	0.0221662	0.0221654	0.0221653	0.0221653	0.0221653	0.0221653	0.0221653	0.0221653
274	1.6	0.5	12000	5500	30	15	0.0165195	0.0218685	0.0221209	0.0221132	0.0221116	0.0221116	0.0221116	0.0221116	0.0221116	0.0221116
275	1.6	0.5	12000	5500	25	20	0.0192715	0.0222686	0.0223415	0.0223402	0.0223401	0.0223401	0.0223401	0.0223401	0.0223401	0.0223401
276	1.6	0.5	12000	5500	25	15	0.0161066	0.0220279	0.0223263	0.0223149	0.0223126	0.0223125	0.0223125	0.0223125	0.0223125	0.0223125
277	1.6	0.5	20000	16500	30	20	0.00733349	0.00884478	0.00875313	0.00875618	0.00875616	0.00875616	0.00875616	0.00875616	0.00875616	0.00875616
278	1.6	0.5	20000	16500	30	15	0.00598691	0.00845836	0.00821404	0.00822789	0.00822743	0.00822743	0.00822742	0.00822742	0.00822743	0.00822743
279	1.6	0.5	20000	16500	25	20	0.00621431	0.00810184	0.00795449	0.00796081	0.00796067	0.00796067	0.00796067	0.00796068	0.00796067	0.00796068
280	1.6	0.5	20000	16500	25	15	0.00448963	0.00764765	0.00725057	0.00727911	0.0072779	0.00727789	0.0072779	0.0072779	0.0072779	0.0072779
281	1.6	0.5	20000	11000	30	20	0.0091551	0.00846273	0.00840384	0.00840114	0.00840104	0.00840104	0.00840104	0.00840104	0.00840104	0.00840104
282	1.6	0.5	20000	11000	30	15	0.00927738	0.00809839	0.00790204	0.00788433	0.00788311	0.00788305	0.00788305	0.00788305	0.00788305	0.00788305

283	1.6	0.5	20000	11000	25	20	0.00855738	0.00783978	0.00776639	0.00776245	0.00776228	0.00776228	0.00776228	0.00776228	0.00776228	0.00776228
284	1.6	0.5	20000	11000	25	15	0.00864821	0.00741921	0.00717319	0.00714742	0.00714537	0.00714524	0.00714523	0.00714523	0.00714524	0.00714523
285	1.6	0.5	20000	5500	30	20	0.00958798	0.00898913	0.00900743	0.00900691	0.00900692	0.00900692	0.00900692	0.00900692	0.00900692	0.00900692
286	1.6	0.5	20000	5500	30	15	0.0100757	0.00858369	0.00869803	0.00869003	0.00869055	0.00869053	0.00869052	0.00869052	0.00869052	0.00869052
287	1.6	0.5	20000	5500	25	20	0.00939318	0.00874142	0.00876223	0.00876161	0.00876163	0.00876163	0.00876163	0.00876163	0.00876163	0.00876163
288	1.6	0.5	20000	5500	25	15	0.00994458	0.00833	0.00845863	0.00844928	0.00844992	0.00844988	0.00844989	0.00844988	0.00844988	0.00844989
289	1.5	0.6	18000	16500	30	20	0.00935494	0.0108107	0.010739	0.0107408	0.0107408	0.0107408	0.0107408	0.0107408	0.0107408	0.0107408
290	1.5	0.6	18000	16500	30	15	0.00824979	0.0105872	0.0103987	0.0104071	0.0104069	0.0104069	0.0104069	0.0104069	0.0104069	0.0104069
291	1.5	0.6	18000	16500	25	20	0.0087185	0.0106015	0.0104784	0.0104827	0.0104826	0.0104826	0.0104826	0.0104826	0.0104826	0.0104826
292	1.5	0.6	18000	16500	25	15	0.00736298	0.010455	0.0101288	0.0101477	0.0101472	0.0101471	0.0101471	0.0101471	0.0101471	0.0101471
293	1.5	0.6	18000	11000	30	20	0.0122831	0.0116425	0.011592	0.0115899	0.0115898	0.0115898	0.0115898	0.0115898	0.0115898	0.0115898
294	1.5	0.6	18000	11000	30	15	0.0126841	0.0116077	0.0114419	0.011428	0.0114271	0.011427	0.011427	0.011427	0.011427	0.011427
295	1.5	0.6	18000	11000	25	20	0.0123654	0.0116965	0.0116317	0.0116284	0.0116283	0.0116283	0.0116283	0.0116283	0.0116283	0.0116283
296	1.5	0.6	18000	11000	25	15	0.0128744	0.0117407	0.0115262	0.011505	0.0115034	0.0115033	0.0115033	0.0115033	0.0115033	0.0115033
297	1.5	0.6	18000	5500	30	20	0.013029	0.0123814	0.0124011	0.0124006	0.0124006	0.0124006	0.0124006	0.0124006	0.0124006	0.0124006
298	1.5	0.6	18000	5500	30	15	0.0138411	0.0122202	0.0123432	0.0123349	0.0123354	0.0123354	0.0123354	0.0123354	0.0123354	0.0123354
299	1.5	0.6	18000	5500	25	20	0.013279	0.0125705	0.012593	0.0125924	0.0125924	0.0125924	0.0125924	0.0125924	0.0125924	0.0125924
300	1.5	0.6	18000	5500	25	15	0.0141981	0.0124353	0.0125747	0.0125648	0.0125655	0.0125655	0.0125655	0.0125655	0.0125655	0.0125655
301	1.5	0.6	16000	16500	30	20	0.0117911	0.0137305	0.013796	0.0137972	0.0137972	0.0137972	0.0137972	0.0137972	0.0137972	0.0137972
302	1.5	0.6	16000	16500	30	15	0.0103981	0.013417	0.0135713	0.0135749	0.0135749	0.0135749	0.0135749	0.0135749	0.0135749	0.0135749
303	1.5	0.6	16000	16500	25	20	0.0109677	0.0136172	0.0137389	0.0137417	0.0137417	0.0137417	0.0137417	0.0137417	0.0137417	0.0137417
304	1.5	0.6	16000	16500	25	15	0.0091892	0.013249	0.0135352	0.0135448	0.0135447	0.0135446	0.0135446	0.0135446	0.0135446	0.0135446
305	1.5	0.6	16000	11000	30	20	0.0171428	0.0145785	0.0145561	0.0145586	0.0145586	0.0145586	0.0145586	0.0145586	0.0145586	0.0145586
306	1.5	0.6	16000	11000	30	15	0.0191234	0.0145391	0.0144443	0.0144586	0.014459	0.014459	0.014459	0.014459	0.014459	0.014459
307	1.5	0.6	16000	11000	25	20	0.017766	0.0147294	0.0146925	0.0146967	0.0146968	0.0146968	0.0146968	0.0146968	0.0146968	0.0146968
308	1.5	0.6	16000	11000	25	15	0.0201834	0.0147555	0.0146093	0.0146334	0.0146343	0.0146342	0.0146342	0.0146342	0.0146342	0.0146342
309	1.5	0.6	16000	5500	30	20	0.0152875	0.0152233	0.0152143	0.0152148	0.0152148	0.0152148	0.0152148	0.0152148	0.0152148	0.0152148
310	1.5	0.6	16000	5500	30	15	0.0153632	0.0152143	0.0151736	0.0151787	0.0151782	0.0151783	0.0151783	0.0151783	0.0151783	0.0151783
311	1.5	0.6	16000	5500	25	20	0.0155613	0.0154475	0.0154392	0.0154397	0.0154397	0.0154397	0.0154397	0.0154397	0.0154397	0.0154397
312	1.5	0.6	16000	5500	25	15	0.0157117	0.0154676	0.015429	0.0154345	0.015434	0.0154341	0.0154341	0.0154341	0.0154341	0.0154341

313	1.5	0.6	12000	16500	30	20	0.0208847	0.0216561	0.0216243	0.0216251	0.0216251	0.0216251	0.0216251	0.0216251	0.0216251	0.0216251
314	1.5	0.6	12000	16500	30	15	0.0206297	0.021584	0.0215278	0.0215298	0.0215298	0.0215298	0.0215298	0.0215298	0.0215298	0.0215298
315	1.5	0.6	12000	16500	25	20	0.0206321	0.0218407	0.0217624	0.0217654	0.0217653	0.0217653	0.0217653	0.0217653	0.0217653	0.0217653
316	1.5	0.6	12000	16500	25	15	0.0202704	0.0218358	0.0216929	0.0217007	0.0217004	0.0217004	0.0217004	0.0217004	0.0217004	0.0217004
317	1.5	0.6	12000	11000	30	20	0.0193003	0.0221838	0.0221298	0.0221282	0.0221283	0.0221283	0.0221283	0.0221283	0.0221283	0.0221283
318	1.5	0.6	12000	11000	30	15	0.0175761	0.022248	0.0220934	0.0220874	0.0220879	0.0220879	0.0220879	0.0220879	0.0220879	0.0220879
319	1.5	0.6	12000	11000	25	20	0.0186747	0.0224323	0.0223375	0.0223339	0.0223341	0.0223341	0.0223341	0.0223341	0.0223341	0.0223341
320	1.5	0.6	12000	11000	25	15	0.0164534	0.0225934	0.0223312	0.0223173	0.0223187	0.0223187	0.0223187	0.0223187	0.0223187	0.0223187
321	1.5	0.6	12000	5500	30	20	0.0202221	0.0225544	0.0225304	0.0225286	0.0225286	0.0225286	0.0225286	0.0225286	0.0225286	0.0225286
322	1.5	0.6	12000	5500	30	15	0.0176954	0.0226433	0.0225312	0.022514	0.0225146	0.0225147	0.0225147	0.0225147	0.0225147	0.0225147
323	1.5	0.6	12000	5500	25	20	0.0202547	0.022791	0.0227584	0.022756	0.022756	0.022756	0.022756	0.022756	0.022756	0.022756
324	1.5	0.6	12000	5500	25	15	0.0175173	0.0229267	0.0227818	0.0227595	0.0227604	0.0227605	0.0227605	0.0227605	0.0227605	0.0227605
325	1.5	0.6	20000	16500	30	20	0.00977776	0.00798443	0.00805876	0.0080568	0.00805683	0.00805682	0.00805682	0.00805682	0.00805682	0.00805683
326	1.5	0.6	20000	16500	30	15	0.010551	0.00737619	0.00759294	0.00758402	0.00758416	0.00758419	0.00758419	0.00758419	0.00758419	0.00758419
327	1.5	0.6	20000	16500	25	20	0.00972857	0.00740224	0.00752156	0.0075177	0.00751776	0.00751777	0.00751777	0.00751777	0.00751777	0.00751777
328	1.5	0.6	20000	16500	25	15	0.0107636	0.0066562	0.00700466	0.00698676	0.00698711	0.00698715	0.00698714	0.00698714	0.00698714	0.00698714
329	1.5	0.6	20000	11000	30	20	0.00772161	0.00882709	0.00887838	0.00888025	0.00888031	0.00888031	0.00888031	0.00888031	0.00888031	0.00888031
330	1.5	0.6	20000	11000	30	15	0.00630095	0.00840608	0.00860424	0.00861852	0.00861938	0.00861942	0.00861941	0.00861942	0.00861942	0.00861942
331	1.5	0.6	20000	11000	25	20	0.00737968	0.00867292	0.00874026	0.00874301	0.0087431	0.00874311	0.00874312	0.0087431	0.0087431	0.00874311
332	1.5	0.6	20000	11000	25	15	0.00580878	0.00822887	0.008485	0.00850575	0.00850715	0.00850722	0.00850723	0.00850722	0.00850723	0.00850723
333	1.5	0.6	20000	5500	30	20	0.0106871	0.00981047	0.00983532	0.0098348	0.00983481	0.00983481	0.00983481	0.00983481	0.00983481	0.00983481
334	1.5	0.6	20000	5500	30	15	0.01181	0.00957178	0.00972649	0.00971896	0.00971919	0.00971919	0.0097192	0.0097192	0.00971919	0.0097192
335	1.5	0.6	20000	5500	25	20	0.0108508	0.00992337	0.00995032	0.00994974	0.00994975	0.00994975	0.00994975	0.00994975	0.00994975	0.00994975
336	1.5	0.6	20000	5500	25	15	0.0120822	0.00971807	0.00988569	0.00987739	0.00987764	0.00987764	0.00987764	0.00987764	0.00987764	0.00987764
337	1.5	0.5	18000	16500	30	20	0.0105963	0.0103472	0.0103509	0.0103509	0.0103509	0.0103509	0.0103509	0.0103509	0.0103509	0.0103509
338	1.5	0.5	18000	16500	30	15	0.0102957	0.00978032	0.00980132	0.00980046	0.0098005	0.0098005	0.0098005	0.0098005	0.0098005	0.0098005
339	1.5	0.5	18000	16500	25	20	0.00998681	0.00959741	0.00960657	0.00960641	0.00960641	0.00960641	0.00960641	0.00960641	0.00960641	0.00960641
340	1.5	0.5	18000	16500	25	15	0.00960668	0.00887337	0.00891184	0.00890983	0.00890993	0.00890994	0.00890993	0.00890992	0.00890993	0.00890993
341	1.5	0.5	18000	11000	30	20	0.00936795	0.0103932	0.0104378	0.0104394	0.0104394	0.0104394	0.0104394	0.0104394	0.0104395	0.0104395
342	1.5	0.5	18000	11000	30	15	0.00792423	0.00983735	0.0100004	0.0100114	0.0100121	0.0100121	0.0100121	0.0100121	0.0100121	0.0100121

343	1.5	0.5	18000	11000	25	20	0.00868811	0.00993908	0.0100024	0.010005	0.0100051	0.0100051	0.0100051	0.0100051	0.0100051	0.0100051
344	1.5	0.5	18000	11000	25	15	0.00700804	0.00928899	0.00951541	0.0095333	0.00953452	0.00953459	0.0095346	0.0095346	0.0095346	0.0095346
345	1.5	0.5	18000	5500	30	20	0.0123749	0.0111758	0.0111938	0.0111939	0.0111939	0.0111939	0.0111939	0.0111939	0.0111939	0.0111939
346	1.5	0.5	18000	5500	30	15	0.0138239	0.0108612	0.0109604	0.0109631	0.0109627	0.0109627	0.0109627	0.0109627	0.0109627	0.0109627
347	1.5	0.5	18000	5500	25	20	0.0123666	0.011098	0.0111171	0.0111173	0.0111172	0.0111172	0.0111172	0.0111172	0.0111172	0.0111172
348	1.5	0.5	18000	5500	25	15	0.013939	0.0108046	0.0109106	0.0109141	0.0109136	0.0109136	0.0109136	0.0109136	0.0109136	0.0109136
349	1.5	0.5	16000	16500	30	20	0.0107464	0.0130327	0.0130472	0.013046	0.013046	0.013046	0.013046	0.013046	0.013046	0.013046
350	1.5	0.5	16000	16500	30	15	0.00902394	0.0125482	0.0125745	0.0125699	0.0125698	0.0125698	0.0125698	0.0125698	0.0125698	0.0125698
351	1.5	0.5	16000	16500	25	20	0.00930959	0.0124881	0.0125153	0.0125121	0.012512	0.012512	0.012512	0.012512	0.012512	0.012512
352	1.5	0.5	16000	16500	25	15	0.00701219	0.0119029	0.0119583	0.0119461	0.0119457	0.0119457	0.0119457	0.0119457	0.0119457	0.0119457
353	1.5	0.5	16000	11000	30	20	0.014868	0.0135354	0.0134675	0.0134659	0.0134659	0.0134659	0.0134659	0.0134659	0.0134659	0.0134659
354	1.5	0.5	16000	11000	30	15	0.0156377	0.0133914	0.0131749	0.0131648	0.0131646	0.0131646	0.0131646	0.0131646	0.0131646	0.0131646
355	1.5	0.5	16000	11000	25	20	0.0149108	0.0133771	0.0132787	0.0132757	0.0132757	0.0132757	0.0132757	0.0132757	0.0132757	0.0132757
356	1.5	0.5	16000	11000	25	15	0.0159004	0.0133097	0.0129968	0.0129786	0.0129783	0.0129783	0.0129783	0.0129783	0.0129783	0.0129783
357	1.5	0.5	16000	5500	30	20	0.0153191	0.0141871	0.0142201	0.0142194	0.0142194	0.0142194	0.0142194	0.0142194	0.0142194	0.0142194
358	1.5	0.5	16000	5500	30	15	0.01665	0.0138945	0.0140817	0.0140728	0.0140731	0.0140731	0.0140731	0.0140731	0.0140731	0.0140731
359	1.5	0.5	16000	5500	25	20	0.015485	0.0142496	0.0142873	0.0142864	0.0142865	0.0142865	0.0142865	0.0142865	0.0142865	0.0142865
360	1.5	0.5	16000	5500	25	15	0.0169721	0.0139725	0.0141861	0.0141755	0.0141759	0.0141759	0.0141759	0.0141759	0.0141759	0.0141759
361	1.5	0.5	12000	16500	30	20	0.0210301	0.0208277	0.0208291	0.0208291	0.0208291	0.0208291	0.0208291	0.0208291	0.0208291	0.0208291
362	1.5	0.5	12000	16500	30	15	0.0209625	0.0205532	0.0205627	0.0205625	0.0205625	0.0205625	0.0205625	0.0205625	0.0205625	0.0205625
363	1.5	0.5	12000	16500	25	20	0.0210478	0.0207068	0.0207102	0.0207103	0.0207103	0.0207103	0.0207103	0.0207103	0.0207103	0.0207103
364	1.5	0.5	12000	16500	25	15	0.021053	0.0204281	0.0204475	0.0204469	0.0204469	0.0204469	0.0204469	0.0204469	0.0204469	0.0204469
365	1.5	0.5	12000	11000	30	20	0.0198237	0.0214421	0.0213677	0.0213694	0.0213694	0.0213694	0.0213694	0.0213694	0.0213694	0.0213694
366	1.5	0.5	12000	11000	30	15	0.0188675	0.0214194	0.0212262	0.021234	0.0212339	0.0212339	0.0212339	0.0212339	0.0212339	0.0212339
367	1.5	0.5	12000	11000	25	20	0.0194699	0.021582	0.0214502	0.0214543	0.0214543	0.0214543	0.0214543	0.0214543	0.0214543	0.0214542
368	1.5	0.5	12000	11000	25	15	0.018256	0.0216759	0.0213303	0.0213488	0.0213483	0.0213483	0.0213483	0.0213483	0.0213483	0.0213483
369	1.5	0.5	12000	5500	30	20	0.0207518	0.0219705	0.0219113	0.0219125	0.0219125	0.0219125	0.0219125	0.0219125	0.0219125	0.0219125
370	1.5	0.5	12000	5500	30	15	0.0194358	0.02211	0.0218394	0.0218522	0.021852	0.021852	0.021852	0.021852	0.021852	0.021852
371	1.5	0.5	12000	5500	25	20	0.0208838	0.0221665	0.0220945	0.0220963	0.0220963	0.0220963	0.0220963	0.0220963	0.0220963	0.0220963
372	1.5	0.5	12000	5500	25	15	0.0195018	0.0223775	0.0220461	0.022064	0.0220638	0.0220637	0.0220637	0.0220637	0.0220637	0.0220637

373	1.5	0.5	20000	16500	30	20	0.0105426	0.00833296	0.00834125	0.00834288	0.00834286	0.00834286	0.00834286	0.00834286	0.00834286	0.00834286
374	1.5	0.5	20000	16500	30	15	0.0115013	0.00777711	0.00778705	0.00779515	0.00779503	0.00779502	0.00779501	0.00779501	0.00779502	0.00779501
375	1.5	0.5	20000	16500	25	20	0.0103041	0.00747237	0.00748247	0.00748596	0.0074859	0.0074859	0.00748591	0.00748591	0.0074859	0.0074859
376	1.5	0.5	20000	16500	25	15	0.0115231	0.00675004	0.00676023	0.00677734	0.00677705	0.00677699	0.006777	0.006777	0.006777	0.006777
377	1.5	0.5	20000	11000	30	20	0.00595509	0.00786001	0.00788418	0.00788309	0.00788305	0.00788306	0.00788306	0.00788306	0.00788306	0.00788306
378	1.5	0.5	20000	11000	30	15	0.00350358	0.00724439	0.00733597	0.00732716	0.0073266	0.0073266	0.00732661	0.00732661	0.00732661	0.00732661
379	1.5	0.5	20000	11000	25	20	0.00494449	0.00713378	0.00716327	0.00716156	0.0071615	0.0071615	0.0071615	0.00716149	0.00716151	0.00716149
380	1.5	0.5	20000	11000	25	15	0.00209818	0.00638861	0.00650492	0.0064914	0.00649044	0.00649046	0.00649046	0.00649046	0.00649046	0.00649046
381	1.5	0.5	20000	5500	30	20	0.00947486	0.00846138	0.00846125	0.00846164	0.00846164	0.00846163	0.00846163	0.00846163	0.00846163	0.00846164
382	1.5	0.5	20000	5500	30	15	0.0106342	0.00811325	0.00810268	0.00810884	0.0081088	0.00810878	0.00810878	0.00810877	0.00810878	0.00810878
383	1.5	0.5	20000	5500	25	20	0.00920787	0.008156	0.0081551	0.00815554	0.00815554	0.00815553	0.00815553	0.00815553	0.00815553	0.00815553
384	1.5	0.5	20000	5500	25	15	0.0104265	0.00780843	0.00779306	0.00780007	0.00780002	0.00780001	0.00780001	0.0078	0.00780001	0.00780001
385	1.4	0.6	18000	16500	30	20	0.0121272	0.0101795	0.0101813	0.0101824	0.0101824	0.0101824	0.0101824	0.0101824	0.0101824	0.0101824
386	1.4	0.6	18000	16500	30	15	0.0130715	0.00981235	0.00980618	0.0098112	0.00981118	0.00981118	0.00981118	0.00981118	0.00981118	0.00981118
387	1.4	0.6	18000	16500	25	20	0.0124329	0.00985939	0.00986018	0.00986263	0.0098626	0.00986259	0.00986259	0.0098626	0.00986261	0.00986261
388	1.4	0.6	18000	16500	25	15	0.0137905	0.0094845	0.00947044	0.00948209	0.00948204	0.00948201	0.00948201	0.00948201	0.00948201	0.00948201
389	1.4	0.6	18000	11000	30	20	0.00921154	0.011049	0.0110625	0.0110615	0.0110615	0.0110615	0.0110615	0.0110615	0.0110615	0.0110615
390	1.4	0.6	18000	11000	30	15	0.00720665	0.0108341	0.0108822	0.0108739	0.0108736	0.0108736	0.0108736	0.0108736	0.0108736	0.0108736
391	1.4	0.6	18000	11000	25	20	0.00891492	0.0110591	0.0110753	0.0110736	0.0110736	0.0110736	0.0110736	0.0110736	0.0110736	0.0110736
392	1.4	0.6	18000	11000	25	15	0.00664671	0.0108755	0.0109375	0.0109242	0.0109237	0.0109237	0.0109237	0.0109237	0.0109237	0.0109237
393	1.4	0.6	18000	5500	30	20	0.0129363	0.0119292	0.0119268	0.0119272	0.0119272	0.0119272	0.0119272	0.0119272	0.0119272	0.0119272
394	1.4	0.6	18000	5500	30	15	0.0143696	0.0118674	0.011842	0.0118477	0.0118477	0.0118477	0.0118477	0.0118477	0.0118477	0.0118477
395	1.4	0.6	18000	5500	25	20	0.0131696	0.0121218	0.0121183	0.0121187	0.0121187	0.0121187	0.0121187	0.0121187	0.0121187	0.0121187
396	1.4	0.6	18000	5500	25	15	0.0147111	0.0121058	0.0120739	0.0120804	0.0120805	0.0120805	0.0120805	0.0120805	0.0120805	0.0120805
397	1.4	0.6	16000	16500	30	20	0.0132325	0.0133336	0.0133231	0.0133235	0.0133235	0.0133235	0.0133235	0.0133235	0.0133235	0.0133235
398	1.4	0.6	16000	16500	30	15	0.0130385	0.0130918	0.0130735	0.0130747	0.0130746	0.0130746	0.0130746	0.0130746	0.0130746	0.0130746
399	1.4	0.6	16000	16500	25	20	0.013152	0.0132551	0.013236	0.013237	0.013237	0.013237	0.013237	0.013237	0.013237	0.013237
400	1.4	0.6	16000	16500	25	15	0.0129777	0.0130458	0.0130083	0.0130116	0.0130114	0.0130114	0.0130114	0.0130114	0.0130114	0.0130114
401	1.4	0.6	16000	11000	30	20	0.0131891	0.0140909	0.0141257	0.0141268	0.0141269	0.0141269	0.0141269	0.0141269	0.0141269	0.0141269
402	1.4	0.6	16000	11000	30	15	0.0121672	0.0138743	0.0140031	0.0140109	0.0140113	0.0140113	0.0140113	0.0140113	0.0140113	0.0140113

403	1.4	0.6	16000	11000	25	20	0.0130731	0.014206	0.014258	0.01426	0.0142601	0.0142601	0.0142601	0.0142601	0.0142601	0.0142601
404	1.4	0.6	16000	11000	25	15	0.0118963	0.0139819	0.0141691	0.0141828	0.0141837	0.0141837	0.0141837	0.0141837	0.0141837	0.0141837
405	1.4	0.6	16000	5500	30	20	0.0160418	0.0148147	0.0148318	0.014832	0.014832	0.014832	0.014832	0.014832	0.014832	0.014832
406	1.4	0.6	16000	5500	30	15	0.0177186	0.0146909	0.0147844	0.0147872	0.0147869	0.0147869	0.0147869	0.0147869	0.0147869	0.0147869
407	1.4	0.6	16000	5500	25	20	0.0163569	0.0150493	0.0150676	0.0150678	0.0150678	0.0150678	0.0150678	0.0150678	0.0150678	0.0150678
408	1.4	0.6	16000	5500	25	15	0.0181782	0.0149525	0.0150532	0.0150569	0.0150565	0.0150565	0.0150565	0.0150565	0.0150565	0.0150565
409	1.4	0.6	12000	16500	30	20	0.0229917	0.0213134	0.0213066	0.021307	0.021307	0.021307	0.021307	0.021307	0.021307	0.021307
410	1.4	0.6	12000	16500	30	15	0.0235756	0.0212185	0.0211991	0.0212001	0.0212001	0.0212001	0.0212001	0.0212001	0.0212001	0.0212001
411	1.4	0.6	12000	16500	25	20	0.0241255	0.021466	0.0214486	0.02145	0.0214501	0.0214501	0.0214501	0.0214501	0.0214501	0.0214501
412	1.4	0.6	12000	16500	25	15	0.0251583	0.0214179	0.0213714	0.0213754	0.0213754	0.0213754	0.0213754	0.0213754	0.0213754	0.0213754
413	1.4	0.6	12000	11000	30	20	0.0233105	0.0217939	0.0218458	0.0218445	0.0218445	0.0218445	0.0218445	0.0218445	0.0218445	0.0218445
414	1.4	0.6	12000	11000	30	15	0.0243726	0.0216493	0.0218039	0.0217976	0.0217978	0.0217978	0.0217978	0.0217978	0.0217978	0.0217978
415	1.4	0.6	12000	11000	25	20	0.024059	0.0219736	0.0220664	0.0220633	0.0220634	0.0220634	0.0220634	0.0220634	0.0220634	0.0220634
416	1.4	0.6	12000	11000	25	15	0.0254866	0.0217838	0.0220579	0.0220431	0.0220437	0.0220437	0.0220437	0.0220437	0.0220437	0.0220437
417	1.4	0.6	12000	5500	30	20	0.022391	0.0222832	0.0222751	0.0222755	0.0222755	0.0222755	0.0222755	0.0222755	0.0222755	0.0222755
418	1.4	0.6	12000	5500	30	15	0.0225611	0.0222833	0.0222549	0.0222586	0.0222583	0.0222583	0.0222583	0.0222583	0.0222583	0.0222583
419	1.4	0.6	12000	5500	25	20	0.0227194	0.0225275	0.0225205	0.022521	0.022521	0.022521	0.022521	0.022521	0.022521	0.022521
420	1.4	0.6	12000	5500	25	15	0.0229749	0.0225451	0.02252	0.0225242	0.0225238	0.0225239	0.0225239	0.0225239	0.0225239	0.0225239
421	1.4	0.6	20000	16500	30	20	0.00766554	0.00744693	0.00742934	0.00742864	0.00742861	0.00742862	0.00742861	0.00742862	0.00742861	0.00742861
422	1.4	0.6	20000	16500	30	15	0.00724601	0.00695809	0.00691348	0.00691028	0.00691009	0.00691008	0.00691008	0.00691008	0.00691008	0.00691008
423	1.4	0.6	20000	16500	25	20	0.00704726	0.0068215	0.00679514	0.00679377	0.00679371	0.00679371	0.00679371	0.00679371	0.00679371	0.00679371
424	1.4	0.6	20000	16500	25	15	0.00658419	0.00627686	0.00620778	0.00620134	0.00620086	0.00620083	0.00620083	0.00620083	0.00620083	0.00620083
425	1.4	0.6	20000	11000	30	20	0.00715711	0.00828756	0.00824615	0.00824668	0.00824669	0.00824669	0.00824669	0.00824669	0.00824669	0.00824669
426	1.4	0.6	20000	11000	30	15	0.00573048	0.00812334	0.00794272	0.00794803	0.00794818	0.00794816	0.00794815	0.00794815	0.00794815	0.00794815
427	1.4	0.6	20000	11000	25	20	0.00684591	0.00810239	0.00804839	0.00804926	0.00804927	0.00804926	0.00804927	0.00804927	0.00804927	0.00804927
428	1.4	0.6	20000	11000	25	15	0.00530017	0.007998	0.00776234	0.00777053	0.00777073	0.00777069	0.00777069	0.00777069	0.00777069	0.00777069
429	1.4	0.6	20000	5500	30	20	0.00998666	0.00926273	0.00925146	0.00925158	0.00925159	0.00925159	0.00925159	0.00925159	0.00925159	0.00925159
430	1.4	0.6	20000	5500	30	15	0.0109951	0.00919096	0.0091107	0.00911232	0.00911253	0.00911253	0.00911253	0.00911253	0.00911253	0.00911253
431	1.4	0.6	20000	5500	25	20	0.0100974	0.00935745	0.00934493	0.00934505	0.00934506	0.00934506	0.00934506	0.00934506	0.00934507	0.00934506
432	1.4	0.6	20000	5500	25	15	0.0111842	0.00933778	0.00924936	0.00925101	0.00925125	0.00925126	0.00925126	0.00925126	0.00925126	0.00925126

433	1.4	0.5	18000	16500	30	20	0.011073	0.00990672	0.00987039	0.00986998	0.00986998	0.00986999	0.00986999	0.00986999	0.00986999	0.00986999
434	1.4	0.5	18000	16500	30	15	0.0112356	0.00939232	0.00929114	0.00928884	0.00928886	0.00928886	0.00928886	0.00928886	0.00928886	0.00928886
435	1.4	0.5	18000	16500	25	20	0.0106385	0.00911616	0.00905025	0.00904918	0.00904919	0.00904921	0.0090492	0.0090492	0.0090492	0.0090492
436	1.4	0.5	18000	16500	25	15	0.0109091	0.00849695	0.00831443	0.00830877	0.00830884	0.00830886	0.00830886	0.00830886	0.00830886	0.00830886
437	1.4	0.5	18000	11000	30	20	0.00878752	0.00992526	0.009886	0.00988649	0.0098865	0.0098865	0.0098865	0.0098865	0.0098865	0.0098865
438	1.4	0.5	18000	11000	30	15	0.00727775	0.00957295	0.00941452	0.00941896	0.00941903	0.00941902	0.00941902	0.00941902	0.00941902	0.00941902
439	1.4	0.5	18000	11000	25	20	0.00812014	0.00943154	0.00937635	0.00937724	0.00937726	0.00937726	0.00937725	0.00937725	0.00937726	0.00937725
440	1.4	0.5	18000	11000	25	15	0.00637897	0.00906946	0.00884651	0.00885412	0.00885424	0.00885422	0.00885421	0.00885422	0.00885422	0.00885422
441	1.4	0.5	18000	5500	30	20	0.0110336	0.0106607	0.0106468	0.0106466	0.0106465	0.0106465	0.0106465	0.0106465	0.0106465	0.0106465
442	1.4	0.5	18000	5500	30	15	0.011357	0.0104758	0.0103886	0.0103846	0.0103845	0.0103845	0.0103845	0.0103845	0.0103845	0.0103845
443	1.4	0.5	18000	5500	25	20	0.0109122	0.0105434	0.0105283	0.010528	0.010528	0.010528	0.010528	0.010528	0.010528	0.010528
444	1.4	0.5	18000	5500	25	15	0.0112678	0.0103907	0.0102955	0.0102908	0.0102907	0.0102907	0.0102907	0.0102907	0.0102907	0.0102907
445	1.4	0.5	16000	16500	30	20	0.0136275	0.0125545	0.0125854	0.0125848	0.0125848	0.0125848	0.0125848	0.0125848	0.0125848	0.0125848
446	1.4	0.5	16000	16500	30	15	0.0138146	0.0119945	0.0120769	0.0120742	0.0120743	0.0120743	0.0120743	0.0120743	0.0120743	0.0120743
447	1.4	0.5	16000	16500	25	20	0.0134922	0.0119237	0.0119868	0.0119849	0.011985	0.011985	0.011985	0.011985	0.011985	0.011985
448	1.4	0.5	16000	16500	25	15	0.0138462	0.0112125	0.0113794	0.0113717	0.011372	0.011372	0.011372	0.011372	0.011372	0.011372
449	1.4	0.5	16000	11000	30	20	0.0111815	0.0129524	0.012983	0.0129827	0.0129827	0.0129826	0.0129826	0.0129826	0.0129826	0.0129826
450	1.4	0.5	16000	11000	30	15	0.00922123	0.0125487	0.0126527	0.0126502	0.0126499	0.0126499	0.0126499	0.0126499	0.0126499	0.0126499
451	1.4	0.5	16000	11000	25	20	0.0105303	0.0127042	0.0127481	0.0127475	0.0127475	0.0127475	0.0127475	0.0127475	0.0127475	0.0127475
452	1.4	0.5	16000	11000	25	15	0.00820763	0.0122659	0.0124187	0.012414	0.0124133	0.0124133	0.0124133	0.0124133	0.0124133	0.0124133
453	1.4	0.5	16000	5500	30	20	0.0146532	0.0137831	0.0137664	0.0137665	0.0137665	0.0137665	0.0137665	0.0137665	0.0137665	0.0137665
454	1.4	0.5	16000	5500	30	15	0.0157707	0.0137046	0.0135989	0.0135995	0.0135998	0.0135998	0.0135998	0.0135998	0.0135998	0.0135998
455	1.4	0.5	16000	5500	25	20	0.0147419	0.0138374	0.0138179	0.013818	0.013818	0.013818	0.013818	0.013818	0.013818	0.013818
456	1.4	0.5	16000	5500	25	15	0.0159626	0.0138095	0.0136871	0.0136876	0.0136878	0.0136879	0.0136879	0.0136879	0.0136879	0.0136879
457	1.4	0.5	12000	16500	30	20	0.0216632	0.0204929	0.0204739	0.0204738	0.0204738	0.0204738	0.0204738	0.0204738	0.0204738	0.0204738
458	1.4	0.5	12000	16500	30	15	0.0218007	0.0202235	0.020184	0.0201838	0.0201838	0.0201838	0.0201838	0.0201838	0.0201838	0.0201838
459	1.4	0.5	12000	16500	25	20	0.0222536	0.0203809	0.0203307	0.0203305	0.0203305	0.0203305	0.0203305	0.0203305	0.0203305	0.0203305
460	1.4	0.5	12000	16500	25	15	0.022685	0.020145	0.0200419	0.0200409	0.020041	0.020041	0.020041	0.020041	0.020041	0.020041
461	1.4	0.5	12000	11000	30	20	0.0231243	0.0210042	0.0210359	0.0210362	0.0210362	0.0210362	0.0210362	0.0210362	0.0210362	0.0210362
462	1.4	0.5	12000	11000	30	15	0.024415	0.0208049	0.0208838	0.0208859	0.0208858	0.0208858	0.0208858	0.0208858	0.0208858	0.0208858

463	1.4	0.5	12000	11000	25	20	0.0239501	0.0210606	0.0211161	0.021117	0.021117	0.021117	0.021117	0.021117	0.021117	0.021117
464	1.4	0.5	12000	11000	25	15	0.0257627	0.0208509	0.0209924	0.0209978	0.0209973	0.0209973	0.0209973	0.0209973	0.0209973	0.0209973
465	1.4	0.5	12000	5500	30	20	0.022877	0.0215781	0.0216141	0.0216134	0.0216134	0.0216134	0.0216134	0.0216134	0.0216134	0.0216134
466	1.4	0.5	12000	5500	30	15	0.0243863	0.0213666	0.0215523	0.0215443	0.0215445	0.0215445	0.0215445	0.0215445	0.0215445	0.0215445
467	1.4	0.5	12000	5500	25	20	0.0232569	0.0217622	0.0218073	0.0218063	0.0218063	0.0218063	0.0218063	0.0218063	0.0218063	0.0218063
468	1.4	0.5	12000	5500	25	15	0.0250026	0.0215463	0.0217781	0.0217672	0.0217675	0.0217675	0.0217675	0.0217675	0.0217675	0.0217675
469	1.4	0.5	20000	16500	30	20	0.00675034	0.0078404	0.00787593	0.00787673	0.00787674	0.00787674	0.00787674	0.00787674	0.00787674	0.00787674
470	1.4	0.5	20000	16500	30	15	0.00528191	0.00719715	0.0073063	0.00731039	0.00731048	0.00731048	0.00731048	0.00731048	0.00731047	0.00731047
471	1.4	0.5	20000	16500	25	20	0.00545675	0.00688909	0.00694885	0.00695055	0.00695062	0.00695062	0.00695062	0.00695062	0.00695062	0.00695062
472	1.4	0.5	20000	16500	25	15	0.00355838	0.00602589	0.00620752	0.00621633	0.0062166	0.0062166	0.0062166	0.0062166	0.0062166	0.0062166
473	1.4	0.5	20000	11000	30	20	0.00764035	0.00727844	0.00728756	0.00728733	0.00728733	0.00728733	0.00728734	0.00728733	0.00728734	0.00728736
474	1.4	0.5	20000	11000	30	15	0.0074613	0.00664316	0.00669099	0.00668831	0.00668846	0.00668844	0.00668845	0.00668845	0.00668845	0.00668845
475	1.4	0.5	20000	11000	25	20	0.00691894	0.00645256	0.00646659	0.00646617	0.00646619	0.00646618	0.00646619	0.00646618	0.00646619	0.00646618
476	1.4	0.5	20000	11000	25	15	0.00669139	0.00566942	0.0057385	0.00573404	0.00573432	0.0057343	0.0057343	0.0057343	0.0057343	0.0057343
477	1.4	0.5	20000	5500	30	20	0.00785025	0.00782532	0.00782067	0.00782054	0.00782054	0.00782054	0.00782054	0.00782054	0.00782054	0.00782054
478	1.4	0.5	20000	5500	30	15	0.00748405	0.00745366	0.00742662	0.00742458	0.00742447	0.00742446	0.00742446	0.00742446	0.00742446	0.00742446
479	1.4	0.5	20000	5500	25	20	0.00744719	0.00744025	0.00743562	0.00743548	0.00743547	0.00743547	0.00743547	0.00743547	0.00743548	0.00743547
480	1.4	0.5	20000	5500	25	15	0.00704732	0.00705599	0.00702884	0.00702663	0.00702649	0.00702649	0.00702649	0.00702649	0.00702649	0.00702649
481	1.3	0.6	18000	16500	30	20	0.0083785	0.00949361	0.00952015	0.0095205	0.0095205	0.0095205	0.0095205	0.0095205	0.0095205	0.0095205
482	1.3	0.6	18000	16500	30	15	0.00705364	0.00902319	0.00910343	0.00910506	0.00910505	0.00910505	0.00910505	0.00910505	0.00910505	0.00910505
483	1.3	0.6	18000	16500	25	20	0.00756717	0.00907209	0.00911947	0.00912029	0.00912029	0.00912029	0.00912029	0.00912029	0.00912029	0.00912029
484	1.3	0.6	18000	16500	25	15	0.00592247	0.0085377	0.00868061	0.00868456	0.00868454	0.00868454	0.00868454	0.00868454	0.00868454	0.00868453
485	1.3	0.6	18000	11000	30	20	0.010965	0.0104088	0.0104248	0.0104244	0.0104244	0.0104244	0.0104244	0.0104244	0.0104244	0.0104244
486	1.3	0.6	18000	11000	30	15	0.0113825	0.0101312	0.0102092	0.010205	0.0102052	0.0102052	0.0102052	0.0102052	0.0102052	0.0102052
487	1.3	0.6	18000	11000	25	20	0.0110742	0.0103746	0.0103981	0.0103974	0.0103974	0.0103974	0.0103974	0.0103974	0.0103974	0.0103974
488	1.3	0.6	18000	11000	25	15	0.0116551	0.0101103	0.0102219	0.0102149	0.0102153	0.0102153	0.0102153	0.0102153	0.0102153	0.0102153
489	1.3	0.6	18000	5500	30	20	0.0113094	0.011352	0.0113495	0.0113495	0.0113495	0.0113495	0.0113495	0.0113495	0.0113495	0.0113495
490	1.3	0.6	18000	5500	30	15	0.0111196	0.0112645	0.0112526	0.0112515	0.0112514	0.0112514	0.0112514	0.0112514	0.0112514	0.0112514
491	1.3	0.6	18000	5500	25	20	0.011472	0.0115373	0.0115351	0.011535	0.011535	0.011535	0.011535	0.011535	0.011535	0.011535
492	1.3	0.6	18000	5500	25	15	0.0112979	0.0114931	0.0114825	0.0114813	0.0114812	0.0114812	0.0114812	0.0114812	0.0114812	0.0114812

493	1.3	0.6	16000	16500	30	20	0.0138103	0.0127829	0.0127589	0.0127587	0.0127587	0.0127587	0.0127587	0.0127587	0.0127587	0.0127587
494	1.3	0.6	16000	16500	30	15	0.0141557	0.0125452	0.0124787	0.0124776	0.0124776	0.0124776	0.0124776	0.0124776	0.0124776	0.0124776
495	1.3	0.6	16000	16500	25	20	0.0140881	0.0126778	0.0126301	0.0126296	0.0126296	0.0126296	0.0126296	0.0126296	0.0126296	0.0126296
496	1.3	0.6	16000	16500	25	15	0.0147238	0.0125027	0.0123715	0.0123685	0.0123686	0.0123686	0.0123686	0.0123686	0.0123686	0.0123686
497	1.3	0.6	16000	11000	30	20	0.0126245	0.0136373	0.0136052	0.0136056	0.0136056	0.0136056	0.0136056	0.0136056	0.0136056	0.0136056
498	1.3	0.6	16000	11000	30	15	0.0115834	0.0135947	0.0134664	0.0134701	0.0134701	0.0134701	0.0134701	0.0134701	0.0134701	0.0134701
499	1.3	0.6	16000	11000	25	20	0.0125828	0.0137747	0.0137274	0.0137282	0.0137282	0.0137282	0.0137282	0.0137282	0.0137282	0.0137282
500	1.3	0.6	16000	11000	25	15	0.0114029	0.0138163	0.0136271	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338	0.0136338
501	1.3	0.6	16000	5500	30	20	0.0147169	0.0143787	0.0143659	0.0143657	0.0143657	0.0143657	0.0143657	0.0143657	0.0143657	0.0143657
502	1.3	0.6	16000	5500	30	15	0.0151838	0.0143927	0.0143133	0.0143096	0.0143095	0.0143095	0.0143095	0.0143095	0.0143095	0.0143095
503	1.3	0.6	16000	5500	25	20	0.0149572	0.0146258	0.0146118	0.0146115	0.0146115	0.0146115	0.0146115	0.0146115	0.0146115	0.0146115
504	1.3	0.6	16000	5500	25	15	0.0154656	0.0146841	0.0145968	0.0145923	0.0145921	0.0145921	0.0145921	0.0145921	0.0145921	0.0145921
505	1.3	0.6	12000	16500	30	20	0.0198948	0.020913	0.0209269	0.020927	0.020927	0.020927	0.020927	0.020927	0.020927	0.020927
506	1.3	0.6	12000	16500	30	15	0.0192336	0.0207767	0.0208058	0.020806	0.020806	0.020806	0.020806	0.020806	0.020806	0.020806
507	1.3	0.6	12000	16500	25	20	0.0193853	0.0210339	0.0210708	0.0210712	0.0210712	0.0210712	0.0210712	0.0210712	0.0210712	0.0210712
508	1.3	0.6	12000	16500	25	15	0.0184414	0.0209055	0.0209833	0.0209844	0.0209843	0.0209843	0.0209843	0.0209843	0.0209843	0.0209843
509	1.3	0.6	12000	11000	30	20	0.0224125	0.0215347	0.0215037	0.0215032	0.0215032	0.0215032	0.0215032	0.0215032	0.0215032	0.0215032
510	1.3	0.6	12000	11000	30	15	0.0229254	0.0215414	0.0214517	0.0214486	0.0214485	0.0214485	0.0214485	0.0214485	0.0214485	0.0214485
511	1.3	0.6	12000	11000	25	20	0.0229028	0.0217918	0.0217372	0.0217359	0.0217358	0.0217358	0.0217358	0.0217358	0.0217358	0.0217358
512	1.3	0.6	12000	11000	25	15	0.0236505	0.021877	0.0217184	0.0217109	0.0217107	0.0217107	0.0217107	0.0217107	0.0217107	0.0217107
513	1.3	0.6	12000	5500	30	20	0.0233966	0.021957	0.0219692	0.0219695	0.0219695	0.0219695	0.0219695	0.0219695	0.0219695	0.0219695
514	1.3	0.6	12000	5500	30	15	0.0252706	0.0218882	0.0219435	0.0219482	0.0219479	0.0219479	0.0219479	0.0219479	0.0219479	0.0219479
515	1.3	0.6	12000	5500	25	20	0.0238144	0.0222218	0.022235	0.0222354	0.0222354	0.0222354	0.0222354	0.0222354	0.0222354	0.0222354
516	1.3	0.6	12000	5500	25	15	0.0259105	0.0221692	0.0222299	0.0222364	0.022236	0.022236	0.022236	0.022236	0.022236	0.022236
517	1.3	0.6	20000	16500	30	20	0.00563477	0.00671939	0.00669263	0.00669278	0.00669279	0.00669279	0.00669279	0.00669279	0.00669279	0.00669279
518	1.3	0.6	20000	16500	30	15	0.00423102	0.00621298	0.00612051	0.00612181	0.00612187	0.00612187	0.00612186	0.00612187	0.00612187	0.00612187
519	1.3	0.6	20000	16500	25	20	0.00460309	0.00598181	0.00593707	0.00593743	0.00593745	0.00593745	0.00593745	0.00593745	0.00593745	0.00593746
520	1.3	0.6	20000	16500	25	15	0.00286549	0.00542232	0.00526848	0.00527123	0.00527141	0.0052714	0.0052714	0.0052714	0.0052714	0.0052714
521	1.3	0.6	20000	11000	30	20	0.00868062	0.0074695	0.0074868	0.00748694	0.00748693	0.00748693	0.00748693	0.00748693	0.00748693	0.00748693
522	1.3	0.6	20000	11000	30	15	0.00975045	0.00706399	0.00714048	0.0071425	0.00714228	0.00714229	0.00714229	0.00714229	0.00714229	0.00714228

523	1.3	0.6	20000	11000	25	20	0.00858243	0.00718484	0.00720643	0.00720669	0.00720668	0.00720667	0.00720667	0.00720667	0.00720667	0.00720667
524	1.3	0.6	20000	11000	25	15	0.00987237	0.00677434	0.00687092	0.00687421	0.00687384	0.00687384	0.00687385	0.00687384	0.00687384	0.00687384
525	1.3	0.6	20000	5500	30	20	0.00828222	0.00853622	0.00854022	0.00854029	0.00854029	0.00854029	0.00854029	0.00854029	0.00854029	0.00854029
526	1.3	0.6	20000	5500	30	15	0.00764261	0.00833633	0.00836944	0.00837094	0.008371	0.008371	0.00837101	0.00837101	0.00837101	0.00837101
527	1.3	0.6	20000	5500	25	20	0.00831565	0.00859361	0.0085982	0.00859828	0.00859828	0.00859828	0.00859828	0.00859828	0.00859828	0.00859828
528	1.3	0.6	20000	5500	25	15	0.0076853	0.00843611	0.00847346	0.00847521	0.00847529	0.00847529	0.00847526	0.00847526	0.00847526	0.00847526
529	1.3	0.5	18000	16500	30	20	0.00803241	0.00932111	0.00931742	0.00931708	0.00931709	0.00931708	0.00931708	0.00931708	0.00931708	0.00931708
530	1.3	0.5	18000	16500	30	15	0.0064787	0.00872107	0.00870523	0.00870348	0.00870349	0.00870349	0.00870349	0.00870349	0.00870349	0.00870349
531	1.3	0.5	18000	16500	25	20	0.00665487	0.00841509	0.00840739	0.00840652	0.00840652	0.00840653	0.00840653	0.00840652	0.00840653	0.00840653
532	1.3	0.5	18000	16500	25	15	0.00458099	0.00765238	0.00762376	0.00761921	0.00761927	0.00761928	0.00761928	0.00761928	0.00761928	0.00761928
533	1.3	0.5	18000	11000	30	20	0.0104736	0.0092249	0.00923712	0.00923736	0.00923735	0.00923735	0.00923735	0.00923735	0.00923735	0.00923735
534	1.3	0.5	18000	11000	30	15	0.0113208	0.00867419	0.00872118	0.00872373	0.00872359	0.00872359	0.00872359	0.00872359	0.00872359	0.00872359
535	1.3	0.5	18000	11000	25	20	0.0101133	0.00861721	0.00863322	0.00863365	0.00863363	0.00863363	0.00863363	0.00863363	0.00863363	0.00863363
536	1.3	0.5	18000	11000	25	15	0.0111505	0.00798144	0.00804423	0.00804871	0.00804845	0.00804845	0.00804845	0.00804845	0.00804845	0.00804845
537	1.3	0.5	18000	5500	30	20	0.00939686	0.00998291	0.00999124	0.0099913	0.0099913	0.0099913	0.0099913	0.0099913	0.0099913	0.0099913
538	1.3	0.5	18000	5500	30	15	0.00805946	0.00962949	0.00969029	0.00969141	0.00969139	0.00969139	0.00969139	0.00969139	0.00969139	0.00969139
539	1.3	0.5	18000	5500	25	20	0.00917832	0.00980554	0.00981471	0.00981478	0.00981478	0.00981478	0.00981478	0.00981478	0.00981478	0.00981478
540	1.3	0.5	18000	5500	25	15	0.00779181	0.00946657	0.00953371	0.00953493	0.00953491	0.00953491	0.0095349	0.0095349	0.0095349	0.0095349
541	1.3	0.5	16000	16500	30	20	0.0120923	0.0120528	0.0120486	0.0120484	0.0120484	0.0120484	0.0120484	0.0120484	0.0120484	0.0120484
542	1.3	0.5	16000	16500	30	15	0.0114837	0.0115066	0.0114999	0.0114996	0.0114995	0.0114995	0.0114995	0.0114995	0.0114995	0.0114995
543	1.3	0.5	16000	16500	25	20	0.0114012	0.0113769	0.0113689	0.0113685	0.0113685	0.0113685	0.0113685	0.0113685	0.0113685	0.0113685
544	1.3	0.5	16000	16500	25	15	0.0106696	0.0107178	0.0107034	0.0107023	0.0107023	0.0107023	0.0107023	0.0107023	0.0107023	0.0107023
545	1.3	0.5	16000	11000	30	20	0.0127315	0.0124041	0.0124107	0.0124105	0.0124106	0.0124105	0.0124105	0.0124105	0.0124106	0.0124105
546	1.3	0.5	16000	11000	30	15	0.0127479	0.0120081	0.0120419	0.0120404	0.0120405	0.0120405	0.0120405	0.0120405	0.0120405	0.0120405
547	1.3	0.5	16000	11000	25	20	0.0125615	0.0121052	0.0121168	0.0121165	0.0121166	0.0121166	0.0121166	0.0121166	0.0121166	0.0121165
548	1.3	0.5	16000	11000	25	15	0.0126688	0.0116856	0.0117404	0.0117375	0.0117377	0.0117377	0.0117377	0.0117377	0.0117377	0.0117377
549	1.3	0.5	16000	5500	30	20	0.0127709	0.0132142	0.0132224	0.0132225	0.0132225	0.0132225	0.0132225	0.0132225	0.0132225	0.0132225
550	1.3	0.5	16000	5500	30	15	0.0118247	0.0129695	0.0130278	0.0130303	0.0130304	0.0130304	0.0130304	0.0130304	0.0130304	0.0130304
551	1.3	0.5	16000	5500	25	20	0.0127451	0.0132402	0.01325	0.0132501	0.0132502	0.0132502	0.0132502	0.0132502	0.0132502	0.0132502
552	1.3	0.5	16000	5500	25	15	0.0117596	0.0130234	0.0130917	0.0130949	0.013095	0.013095	0.013095	0.013095	0.013095	0.013095

553	1.3	0.5	12000	16500	30	20	0.0189514	0.0200522	0.0200557	0.0200556	0.0200556	0.0200556	0.0200556	0.0200556	0.0200556	0.0200556
554	1.3	0.5	12000	16500	30	15	0.0181218	0.0197336	0.0197385	0.0197382	0.0197381	0.0197381	0.0197381	0.0197381	0.0197381	0.0197381
555	1.3	0.5	12000	16500	25	20	0.0180354	0.019871	0.0198812	0.0198808	0.0198808	0.0198808	0.0198808	0.0198808	0.0198808	0.0198808
556	1.3	0.5	12000	16500	25	15	0.0168616	0.0195448	0.0195615	0.0195601	0.01956	0.01956	0.01956	0.01956	0.01956	0.01956
557	1.3	0.5	12000	11000	30	20	0.020347	0.0206354	0.0206397	0.0206398	0.0206398	0.0206398	0.0206398	0.0206398	0.0206398	0.0206398
558	1.3	0.5	12000	11000	30	15	0.0198429	0.02045	0.0204707	0.0204714	0.0204714	0.0204714	0.0204714	0.0204714	0.0204714	0.0204714
559	1.3	0.5	12000	11000	25	20	0.0202595	0.0207034	0.020713	0.0207132	0.0207132	0.0207132	0.0207132	0.0207132	0.0207132	0.0207132
560	1.3	0.5	12000	11000	25	15	0.0196811	0.0205361	0.0205747	0.0205764	0.0205765	0.0205765	0.0205765	0.0205765	0.0205765	0.0205765
561	1.3	0.5	12000	5500	30	20	0.0221992	0.0212739	0.0212546	0.0212546	0.0212546	0.0212546	0.0212546	0.0212546	0.0212546	0.0212546
562	1.3	0.5	12000	5500	30	15	0.023365	0.0212861	0.021176	0.0211752	0.0211753	0.0211753	0.0211753	0.0211753	0.0211753	0.0211753
563	1.3	0.5	12000	5500	25	20	0.0224726	0.0214812	0.0214566	0.0214565	0.0214565	0.0214565	0.0214565	0.0214565	0.0214565	0.0214565
564	1.3	0.5	12000	5500	25	15	0.0237879	0.0215495	0.021411	0.0214095	0.0214097	0.0214097	0.0214097	0.0214097	0.0214097	0.0214097
565	1.3	0.5	20000	16500	30	20	0.00748849	0.00734718	0.00734841	0.0073484	0.00734844	0.00734844	0.00734844	0.00734844	0.00734844	0.00734844
566	1.3	0.5	20000	16500	30	15	0.00708946	0.0067559	0.00676577	0.00676548	0.00676549	0.00676549	0.00676549	0.00676549	0.00676549	0.00676548
567	1.3	0.5	20000	16500	25	20	0.00656457	0.00634114	0.00634446	0.00634442	0.00634442	0.00634442	0.00634443	0.00634442	0.00634442	0.00634442
568	1.3	0.5	20000	16500	25	15	0.00604575	0.00557046	0.0055887	0.005588	0.00558803	0.00558803	0.00558803	0.00558803	0.00558803	0.00558803
569	1.3	0.5	20000	11000	30	20	0.00739727	0.0066167	0.00659776	0.00659765	0.00659765	0.00659766	0.00659765	0.00659766	0.00659765	0.00659766
570	1.3	0.5	20000	11000	30	15	0.00767652	0.00604531	0.00595385	0.00595232	0.00595239	0.0059524	0.0059524	0.0059524	0.0059524	0.0059524
571	1.3	0.5	20000	11000	25	20	0.00655634	0.00568175	0.00565608	0.00565588	0.00565588	0.00565588	0.00565589	0.00565589	0.00565589	0.00565588
572	1.3	0.5	20000	11000	25	15	0.0068144	0.00498149	0.00485839	0.00485582	0.00485594	0.00485595	0.00485595	0.00485595	0.00485595	0.00485595
573	1.3	0.5	20000	5500	30	20	0.00649274	0.00705581	0.00705947	0.00705943	0.00705943	0.00705943	0.00705943	0.00705943	0.00705943	0.00705943
574	1.3	0.5	20000	5500	30	15	0.00501288	0.00658601	0.00661299	0.00661194	0.00661189	0.00661189	0.00661188	0.00661189	0.00661189	0.00661189
575	1.3	0.5	20000	5500	25	20	0.0059768	0.0065667	0.00657046	0.00657041	0.00657041	0.00657041	0.00657041	0.00657041	0.00657041	0.00657041
576	1.3	0.5	20000	5500	25	15	0.00442031	0.00606881	0.00609693	0.00609568	0.00609561	0.00609561	0.0060956	0.00609561	0.0060956	0.00609561
577	1.2	0.6	18000	16500	30	20	0.00915498	0.00871803	0.00872736	0.00872719	0.00872719	0.00872719	0.00872719	0.00872719	0.00872719	0.00872719
578	1.2	0.6	18000	16500	30	15	0.00911236	0.00822449	0.00826007	0.00825891	0.00825894	0.00825894	0.00825894	0.00825894	0.00825894	0.00825894
579	1.2	0.6	18000	16500	25	20	0.00882733	0.00820264	0.00822049	0.00822006	0.00822007	0.00822007	0.00822007	0.00822007	0.00822007	0.00822007
580	1.2	0.6	18000	16500	25	15	0.0088915	0.00765286	0.00771908	0.00771617	0.00771628	0.00771628	0.00771628	0.00771628	0.00771628	0.00771627
581	1.2	0.6	18000	11000	30	20	0.0101835	0.00966048	0.0096454	0.00964519	0.00964518	0.00964518	0.00964519	0.00964519	0.00964519	0.00964518
582	1.2	0.6	18000	11000	30	15	0.0105317	0.0094598	0.00938882	0.00938642	0.00938638	0.00938638	0.00938638	0.00938638	0.00938638	0.00938638

583	1.2	0.6	18000	11000	25	20	0.0101626	0.00958055	0.00955992	0.00955957	0.00955957	0.00955957	0.00955956	0.00955957	0.00955957	0.00955957
584	1.2	0.6	18000	11000	25	15	0.0106386	0.00943612	0.00933903	0.00933506	0.00933497	0.00933497	0.00933497	0.00933497	0.00933497	0.00933497
585	1.2	0.6	18000	5500	30	20	0.0100755	0.0106307	0.0106326	0.0106326	0.0106326	0.0106326	0.0106326	0.0106326	0.0106326	0.0106326
586	1.2	0.6	18000	5500	30	15	0.00892584	0.0104982	0.0105114	0.01051	0.01051	0.01051	0.01051	0.01051	0.01051	0.01051
587	1.2	0.6	18000	5500	25	20	0.0102179	0.0108	0.0108019	0.0108019	0.0108019	0.0108019	0.0108019	0.0108019	0.0108019	0.0108018
588	1.2	0.6	18000	5500	25	15	0.00906455	0.0107151	0.0107281	0.0107265	0.0107264	0.0107264	0.0107264	0.0107264	0.0107264	0.0107264
589	1.2	0.6	16000	16500	30	20	0.0110485	0.0120815	0.0120764	0.0120762	0.0120762	0.0120762	0.0120762	0.0120762	0.0120762	0.0120762
590	1.2	0.6	16000	16500	30	15	0.00999457	0.0117753	0.0117563	0.0117556	0.0117556	0.0117556	0.0117556	0.0117556	0.0117556	0.0117556
591	1.2	0.6	16000	16500	25	20	0.0104225	0.0118989	0.0118881	0.0118876	0.0118876	0.0118876	0.0118876	0.0118876	0.0118876	0.0118876
592	1.2	0.6	16000	16500	25	15	0.00906277	0.0116214	0.0115839	0.0115818	0.0115818	0.0115818	0.0115818	0.0115818	0.0115818	0.0115818
593	1.2	0.6	16000	11000	30	20	0.0141141	0.0129608	0.0129661	0.0129664	0.0129664	0.0129664	0.0129664	0.0129664	0.0129664	0.0129663
594	1.2	0.6	16000	11000	30	15	0.0152148	0.0127869	0.0128029	0.0128053	0.0128052	0.0128052	0.0128052	0.0128052	0.0128052	0.0128052
595	1.2	0.6	16000	11000	25	20	0.0144706	0.013062	0.0130686	0.0130691	0.013069	0.013069	0.013069	0.013069	0.013069	0.013069
596	1.2	0.6	16000	11000	25	15	0.0158896	0.0129257	0.0129464	0.0129507	0.0129506	0.0129506	0.0129506	0.0129506	0.0129506	0.0129506
597	1.2	0.6	16000	5500	30	20	0.0131732	0.0137806	0.013788	0.013788	0.013788	0.013788	0.013788	0.013788	0.013788	0.013788
598	1.2	0.6	16000	5500	30	15	0.0120184	0.0136627	0.0137164	0.0137169	0.0137169	0.0137169	0.0137169	0.0137169	0.0137169	0.0137169
599	1.2	0.6	16000	5500	25	20	0.0133795	0.0140326	0.0140407	0.0140408	0.0140408	0.0140408	0.0140407	0.0140407	0.0140407	0.0140407
600	1.2	0.6	16000	5500	25	15	0.0121872	0.01395	0.0140096	0.0140101	0.01401	0.01401	0.01401	0.01401	0.01401	0.01401
601	1.2	0.6	12000	16500	30	20	0.0208053	0.020462	0.0204663	0.0204662	0.0204662	0.0204662	0.0204662	0.0204662	0.0204662	0.0204662
602	1.2	0.6	12000	16500	30	15	0.0209201	0.0203159	0.0203278	0.0203276	0.0203276	0.0203276	0.0203276	0.0203276	0.0203276	0.0203276
603	1.2	0.6	12000	16500	25	20	0.0211787	0.0205967	0.0206088	0.0206085	0.0206085	0.0206085	0.0206085	0.0206085	0.0206085	0.0206085
604	1.2	0.6	12000	16500	25	15	0.0214675	0.0204747	0.0205068	0.020506	0.020506	0.020506	0.020506	0.020506	0.020506	0.020506
605	1.2	0.6	12000	11000	30	20	0.0197555	0.0210853	0.0210862	0.0210859	0.0210859	0.0210859	0.0210859	0.0210859	0.0210859	0.0210859
606	1.2	0.6	12000	11000	30	15	0.0185902	0.021026	0.0210227	0.0210211	0.0210211	0.0210211	0.0210211	0.0210211	0.0210211	0.0210211
607	1.2	0.6	12000	11000	25	20	0.0195322	0.0213329	0.0213336	0.021333	0.021333	0.021333	0.021333	0.021333	0.021333	0.021333
608	1.2	0.6	12000	11000	25	15	0.0180093	0.0213094	0.0213044	0.0213003	0.0213003	0.0213003	0.0213003	0.0213003	0.0213003	0.0213003
609	1.2	0.6	12000	5500	30	20	0.0217198	0.0216008	0.0215933	0.0215931	0.0215931	0.0215931	0.0215931	0.0215931	0.0215931	0.0215931
610	1.2	0.6	12000	5500	30	15	0.0218279	0.0216075	0.0215682	0.0215659	0.0215658	0.0215658	0.0215658	0.0215658	0.0215658	0.0215658
611	1.2	0.6	12000	5500	25	20	0.0219784	0.0218908	0.0218825	0.0218823	0.0218823	0.0218823	0.0218823	0.0218823	0.0218823	0.0218823
612	1.2	0.6	12000	5500	25	15	0.0220886	0.0219268	0.0218827	0.0218798	0.0218796	0.0218796	0.0218796	0.0218796	0.0218796	0.0218796

613	1.2	0.6	20000	16500	30	20	0.00682747	0.00582957	0.00582383	0.00582403	0.00582404	0.00582404	0.00582404	0.00582404	0.00582404	0.00582404
614	1.2	0.6	20000	16500	30	15	0.00711268	0.00521865	0.0051924	0.00519372	0.00519376	0.00519376	0.00519376	0.00519376	0.00519376	0.00519377
615	1.2	0.6	20000	16500	25	20	0.00621192	0.00492604	0.0049155	0.00491594	0.00491594	0.00491594	0.00491595	0.00491594	0.00491594	0.00491594
616	1.2	0.6	20000	16500	25	15	0.00664886	0.00420691	0.00416123	0.00416412	0.00416422	0.00416421	0.00416423	0.00416421	0.00416421	0.00416421
617	1.2	0.6	20000	11000	30	20	0.00626772	0.00655909	0.00656493	0.00656503	0.00656504	0.00656504	0.00656504	0.00656504	0.00656504	0.00656504
618	1.2	0.6	20000	11000	30	15	0.00543985	0.00612763	0.00616192	0.00616346	0.00616351	0.00616351	0.00616352	0.00616351	0.00616352	0.00616352
619	1.2	0.6	20000	11000	25	20	0.00579862	0.00616041	0.0061688	0.00616896	0.00616897	0.00616897	0.00616897	0.00616897	0.00616897	0.00616897
620	1.2	0.6	20000	11000	25	15	0.00488897	0.00571681	0.00576389	0.00576628	0.0057664	0.0057664	0.0057664	0.00576639	0.0057664	0.00576641
621	1.2	0.6	20000	5500	30	20	0.00722073	0.00766106	0.00765931	0.00765926	0.00765926	0.00765926	0.00765926	0.00765926	0.00765926	0.00765926
622	1.2	0.6	20000	5500	30	15	0.00617378	0.00746906	0.00745153	0.0074504	0.00745042	0.00745042	0.00745043	0.00745043	0.00745043	0.00745042
623	1.2	0.6	20000	5500	25	20	0.00720774	0.00766155	0.00765947	0.00765942	0.00765942	0.00765942	0.00765942	0.00765942	0.00765942	0.00765942
624	1.2	0.6	20000	5500	25	15	0.00617827	0.00751749	0.00749737	0.00749613	0.00749616	0.00749616	0.00749616	0.00749616	0.00749616	0.00749616
625	1.2	0.5	18000	16500	30	20	0.00959318	0.00867061	0.00867656	0.00867664	0.00867664	0.00867664	0.00867664	0.00867664	0.00867664	0.00867664
626	1.2	0.5	18000	16500	30	15	0.00971847	0.0080125	0.00802895	0.00802958	0.00802956	0.00802956	0.00802956	0.00802956	0.00802956	0.00802957
627	1.2	0.5	18000	16500	25	20	0.00893981	0.00764876	0.0076597	0.00765994	0.00765993	0.00765993	0.00765993	0.00765993	0.00765993	0.00765994
628	1.2	0.5	18000	16500	25	15	0.00917425	0.00679058	0.006822	0.0068237	0.00682364	0.00682364	0.00682364	0.00682364	0.00682364	0.00682364
629	1.2	0.5	18000	11000	30	20	0.00799239	0.00845819	0.00846754	0.0084677	0.0084677	0.0084677	0.0084677	0.0084677	0.0084677	0.0084677
630	1.2	0.5	18000	11000	30	15	0.00680317	0.00785089	0.00789884	0.00790057	0.00790062	0.00790062	0.00790062	0.00790062	0.00790062	0.00790062
631	1.2	0.5	18000	11000	25	20	0.00714416	0.00772909	0.00774302	0.00774329	0.0077433	0.0077433	0.0077433	0.0077433	0.0077433	0.0077433
632	1.2	0.5	18000	11000	25	15	0.00572669	0.00701147	0.00708124	0.00708423	0.00708434	0.00708434	0.00708434	0.00708434	0.00708434	0.00708434
633	1.2	0.5	18000	5500	30	20	0.00899273	0.00920079	0.00919651	0.00919655	0.00919655	0.00919655	0.00919655	0.00919655	0.00919655	0.00919655
634	1.2	0.5	18000	5500	30	15	0.00827506	0.00888559	0.0088489	0.00884995	0.00884993	0.00884993	0.00884993	0.00884993	0.00884993	0.00884993
635	1.2	0.5	18000	5500	25	20	0.00873686	0.00894323	0.00893853	0.00893858	0.00893858	0.00893858	0.00893858	0.00893858	0.00893858	0.00893858
636	1.2	0.5	18000	5500	25	15	0.00802816	0.00864344	0.00860295	0.00860421	0.00860419	0.00860419	0.00860419	0.00860419	0.00860419	0.00860419
637	1.2	0.5	16000	16500	30	20	0.0110283	0.0114273	0.0114184	0.0114185	0.0114185	0.0114185	0.0114185	0.0114185	0.0114185	0.0114186
638	1.2	0.5	16000	16500	30	15	0.0102211	0.0108511	0.0108262	0.0108268	0.0108268	0.0108268	0.0108268	0.0108268	0.0108268	0.0108268
639	1.2	0.5	16000	16500	25	20	0.0100871	0.0106588	0.0106397	0.0106401	0.0106401	0.0106401	0.0106401	0.0106401	0.0106401	0.0106401
640	1.2	0.5	16000	16500	25	15	0.00902801	0.00996622	0.00991131	0.00991319	0.00991314	0.00991314	0.00991314	0.00991314	0.00991314	0.00991314
641	1.2	0.5	16000	11000	30	20	0.012306	0.0117386	0.011725	0.0117249	0.0117249	0.0117249	0.0117249	0.0117249	0.0117249	0.0117249
642	1.2	0.5	16000	11000	30	15	0.0124786	0.0113698	0.0113116	0.0113102	0.0113102	0.0113102	0.0113102	0.0113102	0.0113102	0.0113102

643	1.2	0.5	16000	11000	25	20	0.0120428	0.0113737	0.011353	0.0113527	0.0113527	0.0113527	0.0113527	0.0113527	0.0113527	0.0113527
644	1.2	0.5	16000	11000	25	15	0.0123299	0.0110106	0.0109217	0.0109188	0.0109187	0.0109187	0.0109187	0.0109187	0.0109187	0.0109187
645	1.2	0.5	16000	5500	30	20	0.0120932	0.0125642	0.0125594	0.0125594	0.0125594	0.0125594	0.0125594	0.0125594	0.0125594	0.0125594
646	1.2	0.5	16000	5500	30	15	0.0110411	0.0123763	0.0123358	0.0123355	0.0123355	0.0123355	0.0123355	0.0123355	0.0123355	0.0123355
647	1.2	0.5	16000	5500	25	20	0.012064	0.0125562	0.0125504	0.0125504	0.0125504	0.0125504	0.0125504	0.0125504	0.0125504	0.0125504
648	1.2	0.5	16000	5500	25	15	0.0110039	0.0124105	0.0123629	0.0123626	0.0123627	0.0123627	0.0123627	0.0123627	0.0123627	0.0123627
649	1.2	0.5	12000	16500	30	20	0.0202004	0.0195518	0.0195572	0.0195571	0.0195571	0.0195571	0.0195571	0.0195571	0.0195571	0.0195571
650	1.2	0.5	12000	16500	30	15	0.0202165	0.0191956	0.0192072	0.0192072	0.0192072	0.0192072	0.0192072	0.0192072	0.0192072	0.0192072
651	1.2	0.5	12000	16500	25	20	0.0204521	0.0193251	0.0193411	0.019341	0.019341	0.019341	0.019341	0.019341	0.019341	0.019341
652	1.2	0.5	12000	16500	25	15	0.0207084	0.0189473	0.0189827	0.0189824	0.0189824	0.0189824	0.0189824	0.0189824	0.0189824	0.0189824
653	1.2	0.5	12000	11000	30	20	0.0194103	0.0201756	0.0201613	0.0201614	0.0201614	0.0201614	0.0201614	0.0201614	0.0201614	0.0201614
654	1.2	0.5	12000	11000	30	15	0.0186781	0.0200184	0.0199701	0.0199709	0.0199709	0.0199709	0.0199709	0.0199709	0.0199709	0.0199709
655	1.2	0.5	12000	11000	25	20	0.0192099	0.0202497	0.0202219	0.0202222	0.0202222	0.0202222	0.0202222	0.0202222	0.0202222	0.0202222
656	1.2	0.5	12000	11000	25	15	0.0182899	0.0201558	0.020062	0.0200641	0.0200641	0.0200641	0.0200641	0.0200641	0.0200641	0.0200641
657	1.2	0.5	12000	5500	30	20	0.0201506	0.0208067	0.0208174	0.0208175	0.0208175	0.0208175	0.0208175	0.0208175	0.0208175	0.0208175
658	1.2	0.5	12000	5500	30	15	0.0190145	0.0206548	0.020723	0.0207251	0.0207251	0.0207251	0.0207251	0.0207251	0.0207251	0.0207251
659	1.2	0.5	12000	5500	25	20	0.0202525	0.0210138	0.0210274	0.0210275	0.0210275	0.0210275	0.0210275	0.0210275	0.0210275	0.0210275
660	1.2	0.5	12000	5500	25	15	0.0190023	0.0208813	0.0209673	0.0209701	0.0209702	0.0209702	0.0209702	0.0209702	0.0209702	0.0209702
661	1.2	0.5	20000	16500	30	20	0.00696261	0.00675394	0.00674692	0.00674683	0.00674683	0.00674683	0.00674683	0.00674683	0.00674683	0.00674683
662	1.2	0.5	20000	16500	30	15	0.00650849	0.0061741	0.00615143	0.00615058	0.00615056	0.00615059	0.00615059	0.00615059	0.00615059	0.00615059
663	1.2	0.5	20000	16500	25	20	0.00591508	0.00566765	0.00565586	0.00565556	0.00565554	0.00565555	0.00565555	0.00565555	0.00565555	0.00565555
664	1.2	0.5	20000	16500	25	15	0.00533254	0.00492306	0.004884	0.00488202	0.00488194	0.00488193	0.00488193	0.00488193	0.00488193	0.00488193
665	1.2	0.5	20000	11000	30	20	0.00499324	0.00579064	0.00579418	0.00579407	0.00579406	0.00579407	0.00579407	0.00579407	0.00579407	0.00579407
666	1.2	0.5	20000	11000	30	15	0.00322997	0.00508396	0.00510067	0.00509914	0.0050991	0.00509911	0.0050991	0.00509911	0.0050991	0.0050991
667	1.2	0.5	20000	11000	25	20	0.00377039	0.00470148	0.00470566	0.00470548	0.00470547	0.00470547	0.00470547	0.00470547	0.00470547	0.00470547
668	1.2	0.5	20000	11000	25	15	0.00164941	0.00381219	0.00383316	0.00383068	0.00383062	0.00383062	0.00383062	0.00383062	0.00383062	0.00383062
669	1.2	0.5	20000	5500	30	20	0.00612816	0.00614819	0.00614665	0.00614667	0.00614667	0.00614667	0.00614667	0.00614667	0.00614667	0.00614667
670	1.2	0.5	20000	5500	30	15	0.00560465	0.0056495	0.00563722	0.0056378	0.00563778	0.00563778	0.00563778	0.00563778	0.00563778	0.00563778
671	1.2	0.5	20000	5500	25	20	0.00551061	0.00552041	0.00551886	0.00551888	0.00551889	0.00551889	0.00551889	0.00551888	0.00551889	0.00551889
672	1.2	0.5	20000	5500	25	15	0.00495502	0.00497438	0.00496193	0.00496257	0.00496255	0.00496255	0.00496255	0.00496255	0.00496255	0.00496255

673	1.1	0.6	18000	16500	30	20	0.00759869	0.00776272	0.00776482	0.00776484	0.00776484	0.00776484	0.00776484	0.00776484	0.00776484	0.00776484
674	1.1	0.6	18000	16500	30	15	0.00684925	0.00722265	0.00723325	0.00723352	0.00723353	0.00723352	0.00723353	0.00723353	0.00723352	0.00723353
675	1.1	0.6	18000	16500	25	20	0.00686346	0.00710883	0.00711315	0.00711323	0.00711323	0.00711323	0.00711323	0.00711323	0.00711323	0.00711323
676	1.1	0.6	18000	16500	25	15	0.00598304	0.00650449	0.00652427	0.00652496	0.00652498	0.00652498	0.00652498	0.00652498	0.00652498	0.00652498
677	1.1	0.6	18000	11000	30	20	0.00798872	0.00867976	0.00867683	0.00867674	0.00867674	0.00867674	0.00867674	0.00867674	0.00867674	0.00867674
678	1.1	0.6	18000	11000	30	15	0.00675437	0.00838738	0.00836819	0.00836714	0.00836716	0.00836716	0.00836716	0.00836716	0.00836716	0.00836716
679	1.1	0.6	18000	11000	25	20	0.00769502	0.00850667	0.00850206	0.00850192	0.00850193	0.00850192	0.00850193	0.00850192	0.00850193	0.00850192
680	1.1	0.6	18000	11000	25	15	0.00632313	0.00825027	0.00822205	0.00822032	0.00822037	0.00822037	0.00822037	0.00822037	0.00822037	0.00822037
681	1.1	0.6	18000	5500	30	20	0.00977259	0.00972626	0.00972591	0.00972592	0.00972592	0.00972592	0.00972592	0.00972592	0.00972592	0.00972592
682	1.1	0.6	18000	5500	30	15	0.00974109	0.0095707	0.00956994	0.00957009	0.00957008	0.00957008	0.00957008	0.00957008	0.00957008	0.00957008
683	1.1	0.6	18000	5500	25	20	0.0099213	0.00986109	0.00986086	0.00986087	0.00986087	0.00986087	0.00986087	0.00986087	0.00986088	0.00986087
684	1.1	0.6	18000	5500	25	15	0.0099624	0.00975402	0.00975428	0.00975441	0.00975441	0.00975441	0.00975441	0.00975441	0.00975441	0.00975441
685	1.1	0.6	16000	16500	30	20	0.0119788	0.011239	0.0112389	0.011239	0.011239	0.011239	0.011239	0.011239	0.011239	0.011239
686	1.1	0.6	16000	16500	30	15	0.012218	0.0108721	0.0108692	0.0108696	0.0108696	0.0108696	0.0108696	0.0108696	0.0108696	0.0108696
687	1.1	0.6	16000	16500	25	20	0.0120427	0.0109659	0.0109655	0.0109657	0.0109657	0.0109657	0.0109657	0.0109657	0.0109657	0.0109657
688	1.1	0.6	16000	16500	25	15	0.0125663	0.0106071	0.0106012	0.0106023	0.0106023	0.0106023	0.0106023	0.0106023	0.0106023	0.0106023
689	1.1	0.6	16000	11000	30	20	0.0115918	0.0121593	0.0121682	0.0121683	0.0121683	0.0121683	0.0121683	0.0121683	0.0121683	0.0121683
690	1.1	0.6	16000	11000	30	15	0.0106424	0.0119277	0.0119728	0.0119738	0.0119738	0.0119738	0.0119738	0.0119738	0.0119738	0.0119738
691	1.1	0.6	16000	11000	25	20	0.0115054	0.0122217	0.0122353	0.0122355	0.0122355	0.0122355	0.0122355	0.0122355	0.0122355	0.0122355
692	1.1	0.6	16000	11000	25	15	0.0104207	0.0120147	0.0120825	0.0120843	0.0120844	0.0120844	0.0120844	0.0120844	0.0120844	0.0120844
693	1.1	0.6	16000	5500	30	20	0.0129171	0.0130612	0.0130578	0.0130579	0.0130579	0.0130579	0.0130579	0.0130579	0.0130579	0.0130579
694	1.1	0.6	16000	5500	30	15	0.0125794	0.0129938	0.0129655	0.0129664	0.0129664	0.0129664	0.0129664	0.0129664	0.0129664	0.0129664
695	1.1	0.6	16000	5500	25	20	0.0131766	0.0133146	0.0133109	0.0133109	0.0133109	0.0133109	0.0133109	0.0133109	0.0133109	0.0133109
696	1.1	0.6	16000	5500	25	15	0.0128881	0.0132939	0.0132628	0.0132639	0.0132639	0.0132639	0.0132639	0.0132639	0.0132639	0.0132639
697	1.1	0.6	12000	16500	30	20	0.0198231	0.0198976	0.0198979	0.0198979	0.0198979	0.0198979	0.0198979	0.0198979	0.0198979	0.0198979
698	1.1	0.6	12000	16500	30	15	0.0195519	0.0197349	0.019737	0.019737	0.019737	0.019737	0.019737	0.019737	0.019737	0.019737
699	1.1	0.6	12000	16500	25	20	0.0199041	0.0200321	0.0200328	0.0200328	0.0200328	0.0200328	0.0200328	0.0200328	0.0200328	0.0200328
700	1.1	0.6	12000	16500	25	15	0.0196211	0.0199049	0.0199096	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097	0.0199097
701	1.1	0.6	12000	11000	30	20	0.0213084	0.0205567	0.0205663	0.0205662	0.0205662	0.0205662	0.0205662	0.0205662	0.0205662	0.0205662
702	1.1	0.6	12000	11000	30	15	0.0219869	0.0204527	0.0204885	0.0204881	0.0204881	0.0204881	0.0204881	0.0204881	0.0204881	0.0204881

703	1.1	0.6	12000	11000	25	20	0.021868	0.0208093	0.0208274	0.0208272	0.0208272	0.0208272	0.0208272	0.0208272	0.0208272	0.0208272
704	1.1	0.6	12000	11000	25	15	0.0228563	0.0207174	0.0207854	0.0207844	0.0207844	0.0207844	0.0207844	0.0207844	0.0207844	0.0207844
705	1.1	0.6	12000	5500	30	20	0.0203917	0.0211188	0.0211211	0.0211211	0.0211211	0.0211211	0.0211211	0.0211211	0.0211211	0.0211211
706	1.1	0.6	12000	5500	30	15	0.0191438	0.0210742	0.0210873	0.021086	0.021086	0.021086	0.021086	0.021086	0.021086	0.021086
707	1.1	0.6	12000	5500	25	20	0.0206283	0.0214339	0.0214363	0.0214362	0.0214362	0.0214362	0.0214362	0.0214362	0.0214362	0.0214362
708	1.1	0.6	12000	5500	25	15	0.0192766	0.0214169	0.0214306	0.0214288	0.0214288	0.0214288	0.0214288	0.0214288	0.0214288	0.0214288
709	1.1	0.6	20000	16500	30	20	0.00406535	0.00479083	0.00479087	0.00479078	0.00479081	0.0047908	0.0047908	0.0047908	0.0047908	0.0047908
710	1.1	0.6	20000	16500	30	15	0.00261317	0.00409772	0.00409521	0.00409447	0.00409447	0.00409447	0.00409447	0.00409447	0.00409447	0.00409446
711	1.1	0.6	20000	16500	25	20	0.00273646	0.00368805	0.00368771	0.00368751	0.00368751	0.00368751	0.00368751	0.00368751	0.00368751	0.00368751
712	1.1	0.6	20000	16500	25	15	0.000893914	0.00284277	0.00283844	0.00283674	0.00283674	0.00283675	0.00283674	0.00283675	0.00283675	0.00283674
713	1.1	0.6	20000	11000	30	20	0.00518971	0.00543861	0.00543256	0.00543264	0.00543264	0.00543264	0.00543264	0.00543264	0.00543264	0.00543264
714	1.1	0.6	20000	11000	30	15	0.0043919	0.00499579	0.00495959	0.0049608	0.00496077	0.00496077	0.00496077	0.00496078	0.00496078	0.00496078
715	1.1	0.6	20000	11000	25	20	0.00461337	0.00487972	0.00487192	0.00487205	0.00487205	0.00487205	0.00487205	0.00487205	0.00487205	0.00487205
716	1.1	0.6	20000	11000	25	15	0.00375989	0.00442451	0.00437712	0.00437899	0.00437894	0.00437894	0.00437894	0.00437894	0.00437894	0.00437894
717	1.1	0.6	20000	5500	30	20	0.00669517	0.00654829	0.00654963	0.00654961	0.00654961	0.00654961	0.00654962	0.00654961	0.00654961	0.00654961
718	1.1	0.6	20000	5500	30	15	0.00677257	0.00627301	0.00628866	0.0062882	0.00628822	0.00628822	0.00628822	0.00628822	0.00628822	0.00628823
719	1.1	0.6	20000	5500	25	20	0.00661409	0.00645352	0.00645504	0.00645503	0.00645502	0.00645502	0.00645503	0.00645503	0.00645502	0.00645503
720	1.1	0.6	20000	5500	25	15	0.00675932	0.00621818	0.00623581	0.00623528	0.0062353	0.00623529	0.00623527	0.00623527	0.00623527	0.00623527
721	1.1	0.5	18000	16500	30	20	0.00731156	0.00792575	0.00792925	0.00792923	0.00792923	0.00792923	0.00792923	0.00792923	0.00792923	0.00792923
722	1.1	0.5	18000	16500	30	15	0.00603968	0.00723778	0.00724924	0.00724903	0.00724903	0.00724903	0.00724903	0.00724903	0.00724903	0.00724906
723	1.1	0.5	18000	16500	25	20	0.00590737	0.00677975	0.00678655	0.00678649	0.00678648	0.00678649	0.00678652	0.00678652	0.00678652	0.00678652
724	1.1	0.5	18000	16500	25	15	0.00418742	0.0058782	0.0059014	0.00590082	0.0059008	0.00590084	0.00590084	0.00590084	0.00590084	0.00590084
725	1.1	0.5	18000	11000	30	20	0.00752727	0.00754749	0.00754574	0.00754577	0.00754577	0.00754577	0.00754577	0.00754577	0.00754577	0.00754577
726	1.1	0.5	18000	11000	30	15	0.00691659	0.00692462	0.00691733	0.00691766	0.00691765	0.00691765	0.00691765	0.00691765	0.00691765	0.00691765
727	1.1	0.5	18000	11000	25	20	0.00666826	0.00666722	0.00666512	0.00666517	0.00666517	0.00666517	0.00666518	0.00666517	0.00666517	0.00666517
728	1.1	0.5	18000	11000	25	15	0.00595634	0.00592758	0.00591826	0.00591881	0.00591878	0.00591878	0.00591878	0.00591878	0.00591878	0.00591878
729	1.1	0.5	18000	5500	30	20	0.00853552	0.00821731	0.00821882	0.00821882	0.00821882	0.00821882	0.00821882	0.00821882	0.00821882	0.00821882
730	1.1	0.5	18000	5500	30	15	0.00882768	0.00780009	0.00781389	0.00781407	0.00781406	0.00781406	0.00781406	0.00781406	0.00781406	0.00781406
731	1.1	0.5	18000	5500	25	20	0.00818121	0.00784261	0.00784421	0.00784422	0.00784422	0.00784421	0.00784421	0.00784422	0.00784422	0.00784421
732	1.1	0.5	18000	5500	25	15	0.00851687	0.00742402	0.00743876	0.007439	0.007439	0.00743899	0.00743899	0.00743899	0.00743899	0.00743899

733	1.1	0.5	16000	16500	30	20	0.0109818	0.0106755	0.0106709	0.0106709	0.0106709	0.0106709	0.0106709	0.0106709	0.0106709	0.0106709
734	1.1	0.5	16000	16500	30	15	0.0105439	0.010046	0.0100319	0.0100316	0.0100316	0.0100316	0.0100316	0.0100316	0.0100316	0.0100316
735	1.1	0.5	16000	16500	25	20	0.0102227	0.00977987	0.00976971	0.00976959	0.00976958	0.00976958	0.00976959	0.00976959	0.00976958	0.00976959
736	1.1	0.5	16000	16500	25	15	0.00974146	0.00900649	0.00897453	0.00897376	0.00897375	0.00897375	0.00897375	0.00897375	0.00897375	0.00897375
737	1.1	0.5	16000	11000	30	20	0.0102434	0.0108954	0.010892	0.010892	0.010892	0.010892	0.010892	0.010892	0.010892	0.010892
738	1.1	0.5	16000	11000	30	15	0.0089872	0.0104434	0.010424	0.0104235	0.0104235	0.0104235	0.0104235	0.0104235	0.0104235	0.0104235
739	1.1	0.5	16000	11000	25	20	0.00961003	0.0104195	0.0104138	0.0104137	0.0104137	0.0104137	0.0104137	0.0104137	0.0104137	0.0104137
740	1.1	0.5	16000	11000	25	15	0.00812312	0.00994357	0.00991221	0.00991117	0.0099112	0.0099112	0.0099112	0.0099112	0.0099112	0.0099112
741	1.1	0.5	16000	5500	30	20	0.0120431	0.0117347	0.0117376	0.0117375	0.0117375	0.0117375	0.0117375	0.0117375	0.0117375	0.0117375
742	1.1	0.5	16000	5500	30	15	0.0124235	0.0114462	0.0114737	0.0114731	0.0114731	0.0114731	0.0114731	0.0114731	0.0114731	0.0114731
743	1.1	0.5	16000	5500	25	20	0.0120086	0.0116684	0.0116718	0.0116717	0.0116717	0.0116717	0.0116717	0.0116717	0.0116717	0.0116717
744	1.1	0.5	16000	5500	25	15	0.0124823	0.0114097	0.0114414	0.0114408	0.0114408	0.0114408	0.0114408	0.0114408	0.0114408	0.0114408
745	1.1	0.5	12000	16500	30	20	0.0185873	0.0189517	0.0189542	0.0189542	0.0189542	0.0189542	0.0189542	0.0189542	0.0189542	0.0189542
746	1.1	0.5	12000	16500	30	15	0.0179504	0.0185593	0.0185654	0.0185654	0.0185654	0.0185654	0.0185654	0.0185654	0.0185654	0.0185654
747	1.1	0.5	12000	16500	25	20	0.0180268	0.0186747	0.0186825	0.0186826	0.0186826	0.0186826	0.0186826	0.0186826	0.0186826	0.0186826
748	1.1	0.5	12000	16500	25	15	0.017196	0.0182582	0.0182776	0.0182778	0.0182778	0.0182778	0.0182778	0.0182778	0.0182778	0.0182778
749	1.1	0.5	12000	11000	30	20	0.0203465	0.0195763	0.0195743	0.0195744	0.0195744	0.0195744	0.0195744	0.0195744	0.0195744	0.0195744
750	1.1	0.5	12000	11000	30	15	0.0208283	0.0193668	0.019356	0.0193564	0.0193564	0.0193564	0.0193564	0.0193564	0.0193564	0.0193564
751	1.1	0.5	12000	11000	25	20	0.0206992	0.0196186	0.0196142	0.0196143	0.0196143	0.0196143	0.0196143	0.0196143	0.0196143	0.0196143
752	1.1	0.5	12000	11000	25	15	0.0215039	0.0194496	0.0194277	0.0194288	0.0194288	0.0194288	0.0194288	0.0194288	0.0194288	0.0194288
753	1.1	0.5	12000	5500	30	20	0.0199622	0.0202806	0.0202752	0.0202752	0.0202752	0.0202752	0.0202752	0.0202752	0.0202752	0.0202752
754	1.1	0.5	12000	5500	30	15	0.0193567	0.0202052	0.0201651	0.020166	0.020166	0.020166	0.020166	0.020166	0.020166	0.020166
755	1.1	0.5	12000	5500	25	20	0.0201658	0.0204976	0.0204908	0.0204909	0.0204909	0.0204909	0.0204909	0.0204909	0.0204909	0.0204909
756	1.1	0.5	12000	5500	25	15	0.0195625	0.0204685	0.0204182	0.0204195	0.0204195	0.0204195	0.0204195	0.0204195	0.0204195	0.0204195
757	1.1	0.5	20000	16500	30	20	0.00588892	0.00606274	0.0060588	0.00605886	0.00605885	0.00605888	0.00605888	0.00605888	0.00605888	0.00605888
758	1.1	0.5	20000	16500	30	15	0.00515568	0.00546969	0.00545483	0.00545523	0.00545522	0.00545522	0.00545523	0.00545522	0.00545522	0.00545522
759	1.1	0.5	20000	16500	25	20	0.00466166	0.00487755	0.00487064	0.00487077	0.00487077	0.00487077	0.00487077	0.00487077	0.00487077	0.00487077
760	1.1	0.5	20000	16500	25	15	0.00370381	0.00411502	0.00408797	0.00408898	0.00408895	0.00408895	0.00408895	0.00408895	0.00408895	0.00408895
761	1.1	0.5	20000	11000	30	20	0.00529578	0.0048473	0.00485255	0.00485252	0.00485251	0.00485252	0.00485252	0.00485251	0.00485252	0.00485252
762	1.1	0.5	20000	11000	30	15	0.00523724	0.00407354	0.00410665	0.00410616	0.00410616	0.00410616	0.00410616	0.00410616	0.00410616	0.00410616

763	1.1	0.5	20000	11000	25	20	0.00411731	0.00357774	0.00358492	0.00358487	0.00358487	0.00358488	0.00358488	0.00358488	0.00358488	0.00358487
764	1.1	0.5	20000	11000	25	15	0.00397784	0.00258547	0.00263091	0.00263016	0.00263017	0.00263016	0.00263016	0.00263016	0.00263016	0.00263016
765	1.1	0.5	20000	5500	30	20	0.00530647	0.0050411	0.0050411	0.0050411	0.0050411	0.00504111	0.00504111	0.0050411	0.0050411	0.0050411
766	1.1	0.5	20000	5500	30	15	0.00533071	0.00446106	0.00445907	0.00445945	0.00445944	0.00445944	0.00445945	0.00445944	0.00445944	0.00445944
767	1.1	0.5	20000	5500	25	20	0.00450194	0.0042257	0.00422561	0.00422563	0.00422563	0.00422563	0.00422563	0.00422563	0.00422563	0.00422563
768	1.1	0.5	20000	5500	25	15	0.00447687	0.0035706	0.00356775	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818	0.00356818

Appendix B

Expanded Model Expressions of the Case Studies

Models Name	Models
L	0.0277878 + 0.00538059 x1 + 0.0112123 x2 - 1.75301*10^-6 x3 - 1.3171*10^-7 x4 + 0.0000333682 x5 + 0.0000516228 x6
LR	(13803.9 + 30541.9 x1 + 41734.6 x2 - 3.50591 x3 - 0.53358 x4 + 519.3 x5 + 368.795 x6)/(-1.70707*10^6 + 1.07294*10^6 x1 + 732876. x2 + 106.193 x3 - 10.055 x4 + 29518.4 x5 + 15644.8 x6)
SON	0.0306935 + 0.014922 x1 - 0.00455451 x1^2 + 0.0312512 x2 - 0.00258758 x1 x2 + 0.0131176 x2^2 - 5.06531*10^-6 x3 + 4.59696*10^-7 x1 x3 + 1.44392*10^-7 x2 x3 + 6.39452*10^-11 x3^2 - 2.7438*10^-7 x4 + 3.7902*10^-8 x1 x4 - 5.81213*10^-7 x2 x4 - 2.64558*10^-12 x3 x4 + 4.48532*10^-12 x4^2 + 0.000482835 x5 - 0.0000773759 x1 x5 - 0.000758012 x2 x5 + 1.54982*10^-8 x3 x5 + 9.8336*10^-9 x4 x5 - 4.80874*10^-6 x5^2 + 0.000702725 x6 - 0.0000468154 x1 x6 - 0.000358875 x2 x6 + 8.14809*10^-9 x3 x6 + 4.4991*10^-9 x4 x6 - 9.38309*10^-7 x5 x6 - 0.0000155358 x6^2
SONR	(1.60175*10^11 + 1.07437*10^9 x1 + 2.97395*10^7 x1^2 - 2.58543*10^12 x2 + 4.59578*10^9 x1 x2 + 2.36773*10^12 x2^2 - 322128. x3 - 151259. x1 x3 - 833301. x2 x3 + 20.0745 x3^2 - 25627. x4 - 66965. x1 x4 - 407741. x2 x4 + 11.3525 x3 x4 + 2.7671 x4^2 + 1.76236*10^9 x5 + 3.20397*10^7 x1 x5 - 3.82492*10^7 x2 x5 - 4393.16 x3 x5 + 320.434 x4 x5 - 3.04543*10^7 x5^2 + 6.11564*10^10 x6 + 2.70768*10^7 x1 x6 + 1.34247*10^7 x2 x6 - 3561.08 x3 x6 - 224.092 x4 x6 + 1.15311*10^6 x5 x6 - 1.74656*10^9 x6^2)/(-4.50379*10^10 - 6.38741*10^10 x1 + 1.8636*10^10 x1^2 + 3.65734*10^10 x2 + 1.87551*10^11 x1 x2 + 3.44397*10^11 x2^2 + 1.07236*10^7 x3 - 2.18185*10^6 x1 x3 - 9.90066*10^6 x2 x3 - 569.589 x3^2 + 5.34691*10^6 x4 - 2.73011*10^6 x1 x4 - 1.83981*10^7 x2 x4 + 91.288 x3 x4 + 120.696 x4^2 - 4.14185*10^8 x5 + 1.07301*10^9 x1 x5 + 3.04976*10^8 x2 x5 + 134440. x3 x5 - 12849.2 x4 x5 + 7.69136*10^6 x5^2 - 5.52274*10^8 x6 + 1.19624*10^9 x1 x6 + 1.90277*10^9 x2 x6 + 43958.8 x3 x6 - 25108.7 x4 x6 + 5.68427*10^7 x5 x6 - 3.92607*10^7 x6^2)
TON	0.0295467 + 0.0174625 x1 - 0.0179528 x1^2 + 0.00404946 x1^3 + 0.0261044 x2 + 0.00330036 x1 x2 - 0.0040233 x1^2 x2 - 0.00129873 x2^2 - 0.0432211 x1 x2^2 - 0.0814004 x2^3 - 5.98365*10^-6 x3 + 1.55817*10^-6 x1 x3 - 3.54962*10^-7 x1^2 x3 - 2.05948*10^-6 x2 x3 + 1.32929*10^-6 x1 x2 x3 + 0.0000115656 x2^2 x3 + 2.56237*10^-10 x3^2 + 2.72527*10^-12 x1 x3^2 - 2.15702*10^-10 x2 x3^2 - 3.07726*10^-15 x3^3 - 2.42319*10^-7 x4 - 1.19966*10^-7 x1 x4 + 4.3526*10^-8 x1^2 x4 + 9.32343*10^-7 x2 x4 + 1.28332*10^-6 x1 x2 x4 + 4.02488*10^-6 x2^2 x4 + 2.9926*10^-12 x3 x4 - 1.47182*10^-11 x1 x3 x4 - 2.61267*10^-10 x2 x3 x4 + 2.70698*10^-15 x3^2 x4 - 2.45424*10^-11 x4^2 - 9.33033*10^-12 x1 x4^2 - 1.18275*10^-10 x2 x4^2 + 1.66247*10^-15 x3 x4^2 + 2.33933*10^-15 x4^3 + 0.000579948 x5 + 0.000188278 x1 x5 + 0.000115348 x1^2 x5 + 0.00041735 x2 x5 + 0.000591288 x1 x2 x5 - 0.000358926 x2^2 x5 - 7.64577*10^-8 x3 x5 - 2.27106*10^-8 x1 x3 x5 - 8.48279*10^-8 x2 x3 x5 + 1.36999*10^-12 x3^2 x5 + 6.51488*10^-10 x4 x5 - 8.20916*10^-9 x1 x4 x5 - 2.92712*10^-8 x2 x4 x5 + 1.15386*10^-12 x3 x4 x5 + 1.09199*10^-13 x4^2 x5 + 3.44942*10^-6 x5^2 - 8.75463*10^-6 x1 x5^2 - 3.63351*10^-6 x2 x5^2 + 2.17266*10^-9 x3 x5^2 + 3.52222*10^-10 x4 x5^2 - 4.52121*10^-7 x5^3 + 0.000891292 x6 + 0.000373111 x1 x6 + 8.97532*10^-6 x1^2 x6 + 0.000609058 x2 x6 + 0.0000683902 x1 x2 x6 - 0.000665639 x2^2 x6 - 1.25403*10^-7 x3 x6 - 6.52054*10^-9 x1 x3 x6 - 2.1698*10^-8 x2 x3 x6 +

	2.42398*10^-13 x3^2 x6 - 3.05895*10^-9 x4 x6 - 4.695*10^-10 x1 x4 x6 - 6.0064*10^-9 x2 x4 x6 + 3.33451*10^-13 x3 x4 x6 - 8.24254*10^-15 x4^2 x6 + 0.0000136498 x5 x6 + 9.23183*10^-7 x1 x5 x6 + 0.0000167962 x2 x5 x6 - 2.29348*10^-10 x3 x5 x6 - 2.37834*10^-10 x4 x5 x6 - 1.66051*10^-7 x5^2 x6 + 7.56275*10^-6 x6^2 - 0.0000115091 x1 x6^2 - 0.0000120712 x2 x6^2 + 4.27698*10^-9 x3 x6^2 + 3.85045*10^-10 x4 x6^2 - 2.55131*10^-7 x5 x6^2 - 1.50819*10^-6 x6^3
TONR	(-1.07916*10^16 - 3.80583*10^15 x1 - 1.53944*10^13 x1^2 + 2.47884*10^12 x1^3 - 1.76912*10^16 x2 - 3.87053*10^15 x1 x2 - 2.22043*10^11 x1^2 x2 + 3.37959*10^16 x2^2 + 3.61698*10^15 x1 x2^2 - 1.544*10^16 x2^3 + 1.88869*10^11 x3 - 1.41435*10^9 x1 x3 - 1.10927*10^8 x1^2 x3 - 5.94463*10^11 x2 x3 - 1.98132*10^9 x1 x2 x3 + 5.42263*10^11 x2^2 x3 + 172316. x3^2 + 51351.2 x1 x3^2 - 153593. x2 x3^2 + 0.36908 x3^3 + 2.76965*10^12 x4 - 7.7976*10^8 x1 x4 + 1.99881*10^7 x1^2 x4 + 6.43334*10^10 x2 x4 - 1.37949*10^9 x1 x2 x4 - 4.89023*10^10 x2^2 x4 + 247636. x3 x4 + 23191. x1 x3 x4 - 455402. x2 x3 x4 + 2.16111 x3^2 x4 - 3.00131*10^8 x4^2 + 7700.41 x1 x4^2 - 211372. x2 x4^2 + 2.82476 x3 x4^2 + 9097.52 x4^3 - 4.51246*10^13 x5 - 1.20226*10^14 x1 x5 + 9.79483*10^10 x1^2 x5 - 4.50027*10^14 x2 x5 + 7.94775*10^11 x1 x2 x5 - 3.06173*10^14 x2^2 x5 + 1.44537*10^10 x3 x5 - 1.52439*10^7 x1 x3 x5 + 2.63346*10^7 x2 x3 x5 - 936.026 x3^2 x5 - 1.13101*10^11 x4 x5 - 8.24256*10^6 x1 x4 x5 - 2.06087*10^7 x2 x4 x5 - 801.386 x3 x4 x5 - 33.4522 x4^2 x5 + 3.02948*10^13 x5^2 + 2.18812*10^12 x1 x5^2 + 1.42688*10^13 x2 x5^2 - 2.62429*10^8 x3 x5^2 + 2.05722*10^9 x4 x5^2 - 4.07364*10^11 x5^3 + 4.522*10^14 x6 + 7.63652*10^14 x1 x6 - 4.9278*10^9 x1^2 x6 + 1.00663*10^15 x2 x6 + 6.08211*10^11 x1 x2 x6 - 3.53644*10^14 x2^2 x6 - 2.68306*10^10 x3 x6 - 1.29767*10^7 x1 x3 x6 - 3.50124*10^6 x2 x3 x6 + 252.109 x3^2 x6 + 2.07569*10^11 x4 x6 - 7.17324*10^6 x1 x4 x6 - 2.72361*10^7 x2 x4 x6 - 222.995 x3 x4 x6 + 68.4439 x4^2 x6 - 2.51146*10^13 x5 x6 + 9.31574*10^9 x1 x5 x6 - 3.6145*10^10 x2 x5 x6 + 528444. x3 x5 x6 + 451880. x4 x5 x6 - 1.2097*10^12 x5^2 x6 - 4.0366*10^13 x6^2 - 2.18097*10^13 x1 x6^2 - 1.76388*10^13 x2 x6^2 + 7.65432*10^8 x3 x6^2 - 5.93003*10^9 x4 x6^2 + 2.61879*10^12 x5 x6^2 - 5.68211*10^10 x6^3)/(-2.79195*10^14 + 4.08203*10^14 x1 - 3.73504*10^14 x1^2 + 3.53235*10^13 x1^3 - 1.03064*10^15 x2 + 2.23296*10^14 x1 x2 + 2.39106*10^13 x1^2 x2 - 1.04324*10^15 x2^2 - 3.24284*10^13 x1 x2^2 - 1.85346*10^15 x2^3 + 1.1281*10^11 x3 - 8.07791*10^10 x1 x3 + 2.35178*10^9 x1^2 x3 + 2.98538*10^11 x2 x3 + 2.63658*10^11 x1 x2 x3 + 5.30689*10^11 x2^2 x3 - 1.67578*10^7 x3^2 - 4.28806*10^6 x1 x3^2 - 3.32109*10^7 x2 x3^2 + 736.948 x3^3 - 8.21815*10^10 x4 + 2.59348*10^10 x1 x4 + 2.17469*10^8 x1^2 x4 + 3.88812*10^10 x2 x4 - 5.60186*10^10 x1 x2 x4 + 4.26142*10^11 x2^2 x4 + 5.16188*10^6 x3 x4 - 3.65187*10^6 x1 x3 x4 - 2.7421*10^7 x2 x3 x4 + 412.325 x3^2 x4 + 3.31341*10^6 x4^2 + 312801. x1 x4^2 - 8.12244*10^6 x2 x4^2 + 154.518 x3 x4^2 - 11.8344 x4^3 + 1.4856*10^13 x5 + 3.16859*10^12 x1 x5 + 4.71062*10^12 x1^2 x5 - 2.20229*10^13 x2 x5 + 2.94605*10^13 x1 x2 x5 - 5.64485*10^13 x2^2 x5 + 1.76973*10^9 x3 x5 + 1.57132*10^9 x1 x3 x5 - 2.63365*10^9 x2 x3 x5 + 31743.8 x3^2 x5 - 6.67288*10^8 x4 x5 - 3.07368*10^8 x1 x4 x5 - 5.82476*10^7 x2 x4 x5 + 15684.8 x3 x4 x5 - 7129.83 x4^2 x5 + 2.524*10^11 x5^2 - 5.04746*10^11 x1 x5^2 + 7.64394*10^11 x2 x5^2 - 1.86476*10^7 x3 x5^2 + 2.88818*10^7 x4 x5^2 - 8.70554*10^9 x5^3 + 2.3942*10^12 x6 + 1.29516*10^13 x1 x6 + 7.49425*10^10 x1^2 x6 - 9.99595*10^12 x2 x6 + 2.55336*10^13 x1 x2 x6 + 1.13392*10^13 x2^2 x6 + 3.4896*10^9 x3 x6 + 1.36877*10^9 x1 x3 x6 - 1.62149*10^9 x2 x3 x6 - 26649.3 x3^2 x6 - 2.48386*10^9 x4 x6 - 3.24259*10^8 x1 x4 x6 - 8.07862*10^8 x2 x4 x6 + 13555.6 x3 x4 x6 + 719.773 x4^2 x6 - 5.61579*10^11 x5 x6 + 4.27797*10^11 x1 x5 x6 - 1.12181*10^12 x2 x5 x6 + 7.90543*10^7 x3 x5 x6 + 1.51769*10^7 x4 x5 x6 + 5.79557*10^9 x5^2 x6 + 5.88479*10^11 x6^2 - 7.43662*10^11 x1 x6^2 + 2.78552*10^11 x2 x6^2 - 1.22399*10^8 x3 x6^2 + 8.52724*10^7 x4 x6^2 - 7.74928*10^9 x5 x6^2 + 1.82004*10^9 x6^3)
FON	0.0164528 + 0.00958861 x1 - 0.0207402 x1^2 + 0.0130297 x1^3 - 0.0021551 x1^4 + 0.0180561 x2 + 0.00765823 x1 x2 - 0.0106706 x1^2 x2 + 0.00745442 x1^3 x2 + 0.0115966 x2^2 - 0.00327085 x1 x2^2 + 0.0277019 x1^2 x2^2 - 0.0138367 x2^3 - 0.0337066 x1 x2^3 - 0.07824 x2^4 - 2.46884*10^-6 x3 + 2.82778*10^-6 x1 x3 - 1.42751*10^-6 x1^2 x3 + 2.26678*10^-7 x1^3 x3 - 2.22392*10^-6 x2 x3 + 6.3274*10^-7 x1 x2 x3 - 2.2636*10^-6 x1^2 x2 x3 - 3.87335*10^-8 x2^2 x3 - 6.65686*10^-6 x1 x2^2 x3 + 6.43036*10^-6 x2^3 x3 - 4.72415*10^-11 x3^2 - 1.7543*10^-11 x1 x3^2 + 1.24189*10^-11 x1^2 x3^2 + 8.24677*10^-11 x2 x3^2 + 3.58616*10^-10 x1 x2 x3^2 + 4.36744*10^-10 x2^2 x3^2 + 4.07916*10^-15 x3^3 - 2.82614*10^-15 x1 x3^3 - 2.38498*10^-14 x2 x3^3 + 1.35141*10^-19 x3^4 + 4.08672*10^-7 x4 + 4.39919*10^-8 x1 x4 + 6.65104*10^-7 x1^2 x4 - 1.06527*10^-7 x1^3 x4 + 4.19525*10^-7 x2 x4 + 3.51232*10^-8 x1 x2 x4 - 1.5311*10^-6 x1^2 x2 x4 + 1.86186*10^-7 x2^2 x4 - 1.53129*10^-8 x1 x2^2 x4 - 6.05155*10^-7 x2^3 x4 - 1.70102*10^-10 x3 x4

	- 9.17639*10^-11 x1 x3 x4 + 2.59102*10^-11 x1^2 x3 x4 + 5.91049*10^-11 x2 x3 x4 + 3.48426*10^-10 x1 x2 x3 x4 + 7.41845*10^-10 x2^2 x3 x4 + 1.78157*10^-14 x3^2 x4 - 4.3673*10^-15 x1 x3^2 x4 - 4.50244*10^-14 x2 x3^2 x4 + 2.96135*10^-19 x3^3 x4 + 2.8644*10^-12 x4^2 - 1.97979*10^-11 x1 x4^2 + 8.53716*10^-12 x1^2 x4^2 - 8.25771*10^-12 x2 x4^2 + 9.95975*10^-11 x1 x2 x4^2 - 3.79252*10^-11 x2^2 x4^2 - 2.11088*10^-15 x3 x4^2 - 2.06962*10^-15 x1 x3 x4^2 - 2.58467*10^-14 x2 x3 x4^2 + 2.58368*10^-19 x3^2 x4^2 + 1.81945*10^-17 x4^3 - 1.62884*10^-15 x1 x4^3 - 1.33248*10^-15 x2 x4^3 + 4.73705*10^-19 x3 x4^3 + 1.78641*10^-20 x4^4 + 0.000353783 x5 + 0.000159591 x1 x5 - 0.000232781 x1^2 x5 - 0.000126583 x1^3 x5 + 0.000468797 x2 x5 + 0.000336248 x1 x2 x5 - 0.000238223 x1^2 x2 x5 + 0.000530991 x2^2 x5 + 0.000676346 x1 x2^2 x5 + 0.000421768 x2^3 x5 - 4.26753*10^-8 x3 x5 + 2.96211*10^-8 x1 x3 x5 + 2.26465*10^-8 x1^2 x3 x5 - 9.08145*10^-8 x2 x3 x5 + 3.48752*10^-9 x1 x2 x3 x5 - 1.84034*10^-7 x2^2 x3 x5 - 1.07989*10^-13 x3^2 x5 - 1.19612*10^-12 x1 x3^2 x5 + 7.89921*10^-12 x2 x3^2 x5 - 4.36288*10^-17 x3^3 x5 + 1.21471*10^-8 x4 x5 + 8.82288*10^-10 x1 x4 x5 + 4.36648*10^-9 x1^2 x4 x5 - 2.31867*10^-8 x2 x4 x5 - 4.90387*10^-8 x1 x2 x4 x5 - 1.19258*10^-7 x2^2 x4 x5 - 1.92955*10^-12 x3 x4 x5 - 3.92621*10^-13 x1 x3 x4 x5 + 8.37042*10^-12 x2 x3 x4 x5 - 9.44226*10^-17 x3^2 x4 x5 - 1.65041*10^-13 x4^2 x5 + 4.71592*10^-13 x1 x4^2 x5 + 6.15196*10^-12 x2 x4^2 x5 - 6.76962*10^-17 x3 x4^2 x5 - 5.19149*10^-17 x4^3 x5 + 4.08468*10^-6 x5^2 - 8.69316*10^-7 x1 x5^2 + 9.72449*10^-6 x1^2 x5^2 + 0.0000101277 x2 x5^2 + 0.0000139331 x1 x2 x5^2 + 0.0000226418 x2^2 x5^2 + 1.16027*10^-10 x3 x5^2 - 1.47726*10^-9 x1 x3 x5^2 - 3.56836*10^-9 x2 x3 x5^2 + 5.18875*10^-14 x3^2 x5^2 + 3.36791*10^-10 x4 x5^2 + 6.43371*10^-12 x1 x4 x5^2 - 2.15909*10^-9 x2 x4 x5^2 + 7.92392*10^-14 x3 x4 x5^2 - 1.52319*10^-14 x4^2 x5^2 - 1.40915*10^-7 x5^3 - 2.47714*10^-7 x1 x5^3 + 1.34261*10^-7 x2 x5^3 + 5.84152*10^-11 x3 x5^3 + 8.53703*10^-12 x4 x5^3 - 1.39352*10^-8 x5^4 + 0.00054463 x6 + 0.00028949 x1 x6 - 0.000489108 x1^2 x6 - 7.5823*10^-6 x1^3 x6 + 0.000668528 x2 x6 + 0.000304816 x1 x2 x6 + 0.000250288 x1^2 x2 x6 + 0.000631066 x2^2 x6 + 0.000157986 x1 x2^2 x6 + 0.000172168 x2^3 x6 - 6.90245*10^-8 x3 x6 + 6.0633*10^-8 x1 x3 x6 + 2.27155*10^-9 x1^2 x3 x6 - 9.88098*10^-8 x2 x3 x6 - 4.44132*10^-8 x1 x2 x3 x6 - 1.29301*10^-7 x2^2 x3 x6 - 1.00011*10^-12 x3^2 x6 + 1.87888*10^-13 x1 x3^2 x6 + 7.75362*10^-12 x2 x3^2 x6 - 3.97047*10^-17 x3^3 x6 + 1.56429*10^-8 x4 x6 + 4.80371*10^-9 x1 x4 x6 - 2.68083*10^-9 x1^2 x4 x6 - 1.61209*10^-8 x2 x4 x6 - 4.91017*10^-8 x1 x2 x4 x6 - 1.05617*10^-7 x2^2 x4 x6 - 3.40772*10^-12 x3 x4 x6 + 2.1864*10^-13 x1 x3 x4 x6 + 7.12569*10^-12 x2 x3 x4 x6 - 8.62487*10^-17 x3^2 x4 x6 - 1.22673*10^-13 x4^2 x6 + 3.93406*10^-13 x1 x4^2 x6 + 5.11776*10^-12 x2 x4^2 x6 - 5.46337*10^-17 x3 x4^2 x6 - 4.24107*10^-17 x4^3 x6 + 0.0000119847 x5 x6 + 2.66922*10^-6 x1 x5 x6 + 4.17107*10^-6 x1^2 x5 x6 + 7.20328*10^-6 x2 x5 x6 - 2.6099*10^-6 x1 x2 x5 x6 - 0.0000124174 x2^2 x5 x6 - 1.09358*10^-9 x3 x5 x6 + 2.43668*10^-10 x1 x3 x5 x6 + 2.96215*10^-9 x2 x3 x5 x6 - 4.94057*10^-14 x3^2 x5 x6 + 1.935*10^-10 x4 x5 x6 + 6.17662*10^-11 x1 x4 x5 x6 + 2.23108*10^-9 x2 x4 x5 x6 - 4.85708*10^-14 x3 x4 x5 x6 - 6.57989*10^-15 x4^2 x5 x6 + 1.54242*10^-7 x5^2 x6 - 1.76848*10^-7 x1 x5^2 x6 - 3.3783*10^-7 x2 x5^2 x6 + 1.02513*10^-11 x3 x5^2 x6 - 6.16064*10^-12 x4 x5^2 x6 - 3.69772*10^-9 x5^3 x6 + 9.2765*10^-6 x6^2 + 2.53374*10^-6 x1 x6^2 + 8.79303*10^-6 x1^2 x6^2 + 0.0000175815 x2 x6^2 + 9.70887*10^-6 x1 x2 x6^2 + 0.0000324425 x2^2 x6^2 - 4.94836*10^-11 x3 x6^2 - 1.85511*10^-9 x1 x3 x6^2 - 3.88893*10^-9 x2 x3 x6^2 + 3.30099*10^-14 x3^2 x6^2 + 4.50945*10^-10 x4 x6^2 + 3.7548*10^-10 x1 x4 x6^2 - 2.91637*10^-9 x2 x4 x6^2 + 1.39866*10^-13 x3 x4 x6^2 - 2.12492*10^-14 x4^2 x6^2 + 2.27897*10^-7 x5 x6^2 - 1.82735*10^-7 x1 x5 x6^2 - 6.08772*10^-7 x2 x5 x6^2 + 9.63045*10^-12 x3 x5 x6^2 - 1.50416*10^-11 x4 x5 x6^2 + 4.29286*10^-9 x5^2 x6^2 - 4.63219*10^-7 x6^3 - 4.80243*10^-7 x1 x6^3 + 4.04716*10^-8 x2 x6^3 + 1.70013*10^-10 x3 x6^3 + 4.81741*10^-12 x4 x6^3 - 7.86278*10^-9 x5 x6^3 - 6.46622*10^-8 x6^4
FONR	(1.01712*10^19 + 2.07747*10^19 x1 + 2.0326*10^18 x1^2 - 2.47654*10^14 x1^3 - 1.9401*10^15 x1^4 + 2.05099*10^19 x2 + 3.67011*10^19 x1 x2 - 7.80428*10^18 x1^2 x2 + 1.18138*10^16 x1^3 x2 - 1.53913*10^18 x2^2 + 2.78133*10^19 x1 x2^2 + 7.08757*10^18 x1^2 x2^2 - 5.90563*10^17 x2^3 - 4.06623*10^19 x1 x2^3 - 9.44509*10^19 x2^4 - 2.35801*10^15 x3 - 1.07746*10^15 x1 x3 + 5.46913*10^11 x1^2 x3 + 2.58833*10^11 x1^3 x3 + 1.47556*10^15 x2 x3 + 1.84483*10^15 x1 x2 x3 - 1.82728*10^12 x1^2 x2 x3 - 3.23061*10^15 x2^2 x3 - 1.73308*10^15 x1 x2^2 x3 + 2.38431*10^15 x2^3 x3 + 1.37021*10^11 x3^2 + 3.92072*10^6 x1 x3^2 - 2.44048*10^7 x1^2 x3^2 - 3.39878*10^10 x2 x3^2 + 8.01103*10^7 x1 x2 x3^2 + 2.56584*10^10 x2^2 x3^2 - 1.14878*10^7 x3^3 + 8855.83 x1 x3^3 + 97655.1 x2 x3^3 + 173.356 x3^4 - 6.75465*10^15 x4 - 4.43306*10^15 x1 x4 + 1.01086*10^11 x1^2 x4 - 1.60095*10^11 x1^3 x4 - 1.58917*10^15 x2 x4 - 8.68364*10^15 x1 x2 x4 - 5.04234*10^10 x1^2 x2 x4 + 2.80318*10^15 x2^2 x4 + 7.8532*10^15 x1 x2^2 x4 - 8.86214*10^15 x2^3 x4 + 1.78009*10^11 x3 x4 + 8.43078*10^8 x1 x3 x4 +

```
3.09405*10^7 x1^2 x3 x4 - 2.42934*10^11 x2 x3 x4 - 4.51997*10^7 x1 x2 x3
x4 + 2.18469*10^11 x2^2 x3 x4 + 29021.2 x3^2 x4 - 1989.23 x1 x3^2 x4 +
85013.2 x2 x3^2 x4 - 1.24772 x3^3 x4 - 1.53208*10^11 x4^2 + 1.07039*10^12
x1 x4^2 - 1.10941*10^6 x1^2 x4^2 + 1.01607*10^12 x2 x4^2 - 2.03543*10^7 x1
x2 x4^2 +  5.07647*10^11 x2^2 x4^2 - 1.66714*10^7 x3 x4^2 + 150.938 x1 x3
x4^2 + 258.001 x2 x3 x4^2 - 0.59642 x3^2 x4^2 + 1.32379*10^8 x4^3 -
3.24259*10^7 x1 x4^3 - 4.77153*10^7 x2 x4^3 + 505.706 x3 x4^3 - 4895.12
x4^4 + 2.33662*10^18 x5 - 9.55644*10^16 x1 x5 + 1.06288*10^16 x1^2 x5 +
9.0661*10^12 x1^3 x5 - 2.61247*10^17 x2 x5 - 9.55975*10^17 x1 x2 x5 +
4.83984*10^12 x1^2 x2 x5 + 6.8778*10^17 x2^2 x5 - 6.90289*10^17 x1 x2^2 x5
+ 6.19244*10^18 x2^3 x5 - 7.51681*10^13 x3 x5 + 3.14174*10^13 x1 x3 x5 +
8.15548*10^9 x1^2 x3 x5 - 1.20675*10^13 x2 x3 x5 + 6.59798*10^10 x1 x2 x3
x5 + 5.46706*10^13 x2^2 x3 x5 + 4.58173*10^9 x3^2 x5 - 1.88185*10^6 x1
x3^2 x5 + 9.33919*10^6 x2 x3^2 x5 + 258.668 x3^3 x5 - 4.81846*10^14 x4 x5
+ 1.23353*10^13 x1 x4 x5 - 8.11503*10^8 x1^2 x4 x5 - 7.00717*10^14 x2 x4
x5 -  1.52938*10^10 x1 x2 x4 x5 + 2.04645*10^14 x2^2 x4 x5 - 8.49589*10^9
x3 x4 x5 - 705574. x1 x3 x4 x5 + 4.01995*10^6 x2 x3 x4 x5 - 142.388 x3^2
x4 x5 + 2.78188*10^10 x4^2 x5 + 120930. x1 x4^2 x5 - 2.13599*10^6 x2 x4^2
x5 - 46.2634 x3 x4^2 x5 - 2.6562*10^6 x4^3 x5 + 6.02828*10^15 x5^2 +
1.59168*10^16 x1 x5^2 - 1.97607*10^14 x1^2 x5^2 + 2.05435*10^16 x2 x5^2 +
3.10987*10^16 x1 x2 x5^2 - 7.53074*10^16 x2^2 x5^2 + 6.56983*10^11 x3 x5^2
- 5.71073*10^11 x1 x3 x5^2 - 8.76913*10^11 x2 x3 x5^2 - 8.36332*10^7 x3^2
x5^2 - 7.04976*10^12 x4 x5^2 - 2.2337*10^11 x1 x4 x5^2 + 8.65015*10^12 x2
x4 x5^2 + 1.54511*10^8 x3 x4 x5^2 + 1.08794*10^9 x4^2 x5^2 + 6.45088*10^13
x5^3 - 4.79186*10^14 x1 x5^3 - 1.22518*10^15 x2 x5^3 + 1.18286*10^10 x3
x5^3 + 3.71987*10^9 x4 x5^3 + 9.96732*10^12 x5^4 + 2.05473*10^18 x6 -
2.2221*10^17 x1 x6 - 6.32943*10^15 x1^2 x6 - 2.21065*10^13 x1^3 x6 -
4.42528*10^17 x2 x6 + 2.61718*10^15 x1 x2 x6 + 4.03055*10^14 x1^2 x2 x6 -
1.64883*10^18 x2^2 x6 - 4.81855*10^17 x1 x2^2 x6 + 4.59294*10^18 x2^3 x6
+ 4.00648*10^13 x3 x6 + 1.56395*10^13 x1 x3 x6 + 6.85566*10^9 x1^2 x3 x6
- 3.14477*10^14 x2 x3 x6 - 4.8409*10^10 x1 x2 x3 x6 + 2.15878*10^14 x2^2
x3 x6 + 1.06331*10^10 x3^2 x6 - 1.83005*10^6 x1 x3^2 x6 + 8.78713*10^6 x2
x3^2 x6 + 178.016 x3^3 x6 - 2.8692*10^14 x4 x6 - 4.87138*10^14 x1 x4 x6 -
6.02796*10^9 x1^2 x4 x6 - 1.87333*10^14 x2 x4 x6 - 1.07747*10^10 x1 x2 x4
x6 - 1.25121*10^14 x2^2 x4 x6 + 2.00575*10^10 x3 x4 x6 + 525106. x1 x3 x4
x6 + 5.17851*10^6 x2 x3 x4 x6 - 73.9176 x3^2 x4 x6 + 3.67273*10^10 x4^2 x6
- 48893.2 x1 x4^2 x6 - 688598. x2 x4^2 x6 - 16.7025 x3 x4^2 x6 + 494773.
x4^3 x6 - 5.92488*10^13 x5 x6 + 1.59307*10^15 x1 x5 x6 + 4.92776*10^12
x1^2 x5 x6 + 1.24683*10^16 x2 x5 x6 + 1.10576*10^13 x1 x2 x5 x6 -
1.27197*10^17 x2^2 x5 x6 - 1.23636*10^12 x3 x5 x6 - 1.55045*10^9 x1 x3 x5
x6 -  2.86091*10^8 x2 x3 x5 x6 - 29683.2 x3^2 x5 x6 + 5.45873*10^10 x4 x5
x6 + 1.94185*10^8 x1 x4 x5 x6 - 1.21318*10^9 x2 x4 x5 x6 - 78632.7 x3 x4
x5 x6 - 2542.77 x4^2 x5 x6 + 8.2461*10^14 x5^2 x6 + 1.98645*10^14 x1 x5^2
x6 + 2.48803*10^15 x2 x5^2 x6 + 8.3642*10^10 x3 x5^2 x6 + 8.08745*10^10 x4
x5^2 x6 - 3.18196*10^13 x5^3 x6 + 8.73862*10^16 x6^2 + 6.29637*10^15 x1
x6^2 + 1.73129*10^14 x1^2 x6^2 + 1.21607*10^17 x2 x6^2 + 1.4965*10^16 x1
x2 x6^2 - 1.68079*10^17 x2^2 x6^2 + 5.9294*10^12 x3 x6^2 - 4.4517*10^11 x1
x3 x6^2 + 2.19875*10^12 x2 x3 x6^2 - 3.04183*10^8 x3^2 x6^2 + 1.05957*10^12
x4 x6^2 + 1.39192*10^13 x1 x4 x6^2 + 9.2848*10^12 x2 x4 x6^2 - 5.73106*10^8
x3 x4 x6^2 - 1.51584*10^9 x4^2 x6^2 - 1.66488*10^15 x5 x6^2 - 3.5627*10^14
x1 x5 x6^2 - 2.68827*10^14 x2 x5 x6^2 - 9.60414*10^10 x3 x5 x6^2 -
1.28584*10^11 x4 x5 x6^2 + 8.25558*10^12 x5^2 x6^2 - 8.69199*10^14 x6^3 +
9.72385*10^12 x1 x6^3 + 2.33271*10^15 x2 x6^3 - 1.2969*10^11 x3 x6^3 +
4.73045*10^11 x4 x6^3 + 3.38328*10^13 x5 x6^3 - 2.75443*10^14
x6^4)/(5.38739*10^17 - 5.74303*10^17 x1 - 7.65656*10^16 x1^2 +
3.65972*10^16 x1^3 - 1.82187*10^16 x1^4 - 6.39944*10^17 x2 - 1.9706*10^17
x1 x2 - 5.15886*10^17 x1^2 x2 + 3.91369*10^17 x1^3 x2 - 3.24883*10^18 x2^2
+ 2.44432*10^17 x1 x2^2 - 1.51856*10^18 x1^2 x2^2 - 9.5259*10^18 x2^3 +
4.26499*10^18 x1 x2^3 - 2.52699*10^19 x2^4 - 4.97221*10^13 x3 +
4.16189*10^13 x1 x3 + 2.23928*10^13 x1^2 x3 - 2.76633*10^13 x1^3 x3 +
2.41546*10^14 x2 x3 + 1.67962*10^15 x1 x2 x3 + 1.34565*10^14 x1^2 x2 x3 +
1.14155*10^15 x2^2 x3 + 5.72717*10^15 x1 x2^2 x3 + 3.05268*10^15 x2^3 x3
+ 2.77985*10^9 x3^2 - 1.91577*10^10 x1 x3^2 + 2.70861*10^9 x1^2 x3^2 -
7.3625*10^10 x2 x3^2 - 2.19809*10^11 x1 x2 x3^2 - 2.66031*10^11 x2^2 x3^2
+ 437457. x3^3 + 1.55235*10^6 x1 x3^3 +  6.02153*10^6 x2 x3^3 - 16.29 x3^4
+ 6.08215*10^13 x4 - 1.4939*10^10 x1 x4 + 2.55213*10^13 x1^2 x4 -
5.42026*10^12 x1^3 x4 + 1.92613*10^14 x2 x4 - 2.25897*10^13 x1 x2 x4 +
5.79371*10^12 x1^2 x2 x4 + 7.03494*10^13 x2^2 x4 + 4.35417*10^13 x1 x2^2
x4 + 2.93864*10^14 x2^3 x4 - 1.50606*10^10 x3 x4 - 1.89542*10^10 x1 x3 x4
- 1.60156*10^9 x1^2 x3 x4 - 2.68445*10^10 x2 x3 x4 - 1.54952*10^11 x1 x2
x3 x4 - 2.84906*10^10 x2^2 x3 x4 + 3.29205*10^6 x3^2 x4 + 2.89306*10^6 x1
x3^2 x4 + 3.24156*10^6 x2 x3^2 x4 - 83.1963 x3^3 x4 + 5.26373*10^9 x4^2 -
2.49379*10^9 x1 x4^2 - 7.70717*10^7 x1^2 x4^2 + 8.77234*10^9 x2 x4^2 -
5.87329*10^8 x1 x2 x4^2 + 8.75947*10^9 x2^2 x4^2 - 799547. x3 x4^2 +
1.04597*10^6 x1 x3 x4^2 - 442404. x2 x3 x4^2 - 27.1064 x3^2 x4^2 - 91567.1
```

	x4^3 + 93991.9 x1 x4^3 - 519522. x2 x4^3 + 48.4396 x3 x4^3 + 22.9745 x4^4 + 1.49944*10^16 x5 - 2.84962*10^15 x1 x5 - 6.86046*10^15 x1^2 x5 + 5.3519*10^13 x1^3 x5 + 1.39321*10^15 x2 x5 - 1.62194*10^16 x1 x2 x5 + 3.73476*10^15 x1^2 x2 x5 - 1.07829*10^16 x2^2 x5 + 2.04454*10^15 x1 x2^2 x5 - 1.57196*10^16 x2^3 x5 - 1.7679*10^12 x3 x5 + 1.70241*10^12 x1 x3 x5 + 9.28755*10^11 x1^2 x3 x5 - 3.05864*10^12 x2 x3 x5 - 1.30332*10^13 x1 x2 x3 x5 - 4.94644*10^12 x2^2 x3 x5 - 1.8721*10^7 x3^2 x5 - 1.35108*10^8 x1 x3^2 x5 + 1.88952*10^9 x2 x3^2 x5 - 20623.4 x3^3 x5 + 8.33214*10^11 x4 x5 + 6.01893*10^11 x1 x4 x5 - 8.1673*10^10 x1^2 x4 x5 + 7.54832*10^11 x2 x4 x5 - 7.32839*10^11 x1 x2 x4 x5 + 1.92658*10^12 x2^2 x4 x5 - 2.85516*10^8 x3 x4 x5 + 1.26672*10^8 x1 x3 x4 x5 + 4.4519*10^8 x2 x3 x4 x5 - 27349.9 x3^2 x4 x5 - 3.88464*10^8 x4^2 x5 + 4.22797*10^6 x1 x4^2 x5 - 8.2814*10^7 x2 x4^2 x5 - 2724.97 x3 x4^2 x5 - 17470.2 x4^3 x5 + 1.82483*10^13 x5^2 + 5.42429*10^13 x1 x5^2 - 1.7186*10^14 x1^2 x5^2 + 1.44734*10^14 x2 x5^2 - 2.57392*10^14 x1 x2 x5^2 + 1.52505*10^15 x2^2 x5^2 + 1.01864*10^11 x3 x5^2 + 1.60931*10^11 x1 x3 x5^2 - 5.01594*10^11 x2 x3 x5^2 + 1.30645*10^6 x3^2 x5^2 - 3.32412*10^10 x4 x5^2 - 3.30811*10^8 x1 x4 x5^2 - 1.81702*10^10 x2 x4 x5^2 + 1.01122*10^7 x3 x4 x5^2 + 1.86015*10^7 x4^2 x5^2 - 7.62151*10^12 x5^3 + 1.53762*10^13 x1 x5^3 + 3.88797*10^13 x2 x5^3 + 4.39136*10^9 x3 x5^3 - 3.28429*10^9 x4 x5^3 - 5.00676*10^11 x5^4 + 2.8737*10^15 x6 - 2.06845*10^16 x1 x6 - 5.67438*10^15 x1^2 x6 - 6.19844*10^13 x1^3 x6 + 1.63661*10^16 x2 x6 - 2.42848*10^16 x1 x2 x6 + 1.28229*10^16 x1^2 x2 x6 - 6.73635*10^15 x2^2 x6 + 1.19652*10^16 x1 x2^2 x6 - 2.56131*10^17 x2^3 x6 - 3.21603*10^12 x3 x6 + 3.77868*10^12 x1 x3 x6 + 4.00708*10^11 x1^2 x3 x6 - 9.19351*10^11 x2 x3 x6 - 1.04709*10^13 x1 x2 x3 x6 + 6.62546*10^12 x2^2 x3 x6 + 1.56144*10^8 x3^2 x6 - 2.11639*10^8 x1 x3^2 x6 + 7.45032*10^8 x2 x3^2 x6 - 12210.8 x3^3 x6 + 3.96479*10^12 x4 x6 - 2.43412*10^12 x1 x4 x6 - 1.92793*10^11 x1^2 x4 x6 + 5.81401*10^12 x2 x4 x6 - 5.60764*10^11 x1 x2 x4 x6 + 3.14192*10^12 x2^2 x4 x6 - 5.38389*10^8 x3 x4 x6 + 7.0909*10^7 x1 x3 x4 x6 + 2.33016*10^8 x2 x3 x4 x6 - 11173. x3^2 x4 x6 + 1.29438*10^7 x4^2 x6 - 920919. x1 x4^2 x6 - 2.99129*10^7 x2 x4^2 x6 - 1883.91 x3 x4^2 x6 - 35210.8 x4^3 x6 - 7.55287*10^12 x5 x6 - 5.78578*10^14 x1 x5 x6 + 2.5543*10^14 x1^2 x5 x6 - 1.4978*10^15 x2 x5 x6 + 5.45811*10^14 x1 x2 x5 x6 - 5.98912*10^14 x2^2 x5 x6 - 2.14838*10^10 x3 x5 x6 + 1.06829*10^11 x1 x3 x5 x6 + 8.75418*10^9 x2 x3 x5 x6 - 1.09845*10^7 x3^2 x5 x6 - 1.05019*10^11 x4 x5 x6 + 2.72566*10^9 x1 x4 x5 x6 - 3.33531*10^10 x2 x4 x5 x6 - 319794. x3 x4 x5 x6 - 157986. x4^2 x5 x6 - 1.7168*10^13 x5^2 x6 + 4.00176*10^12 x1 x5^2 x6 - 6.48555*10^12 x2 x5^2 x6 + 3.09179*10^9 x3 x5^2 x6 + 1.44163*10^9 x4 x5^2 x6 - 3.77644*10^11 x5^3 x6 - 8.02584*10^14 x6^2 - 2.51727*10^11 x1 x6^2 - 3.16201*10^14 x1^2 x6^2 + 2.11794*10^15 x2 x6^2 - 5.78788*10^14 x1 x2 x6^2 + 2.90841*10^15 x2^2 x6^2 + 1.58602*10^10 x3 x6^2 + 2.6335*10^11 x1 x3 x6^2 - 7.14195*10^11 x2 x3 x6^2 + 9.15056*10^6 x3^2 x6^2 - 1.47183*10^11 x4 x6^2 + 8.88784*10^10 x1 x4 x6^2 - 2.85257*10^11 x2 x4 x6^2 + 2.02898*10^7 x3 x4 x6^2 + 3.38805*10^7 x4^2 x6^2 + 1.55567*10^13 x5 x6^2 + 4.35225*10^12 x1 x5 x6^2 + 4.0989*10^13 x2 x5 x6^2 + 3.05956*10^9 x3 x5 x6^2 + 2.4716*10^9 x4 x5 x6^2 - 2.72156*10^11 x5^2 x6^2 - 2.40165*10^13 x6^3 + 3.66912*10^13 x1 x6^3 + 1.76651*10^14 x2 x6^3 + 1.79965*10^9 x3 x6^3 - 1.15873*10^10 x4 x6^3 + 3.92706*10^11 x5 x6^3 - 5.96968*10^11 x6^4)
FOTN	-0.000862231 - 0.00378423 Cos[x1] - 0.00445619 Cos[x2] - 0.000543187 Cos[x3] - 0.000762211 Cos[x4] - 0.00105373 Cos[x5] + 0.000590957 Cos[x6] + 0.0128856 Sin[x1] + 0.0132587 Sin[x2] - 0.00440135 Sin[x3] + 0.00027134 Sin[x4] + 0.000892927 Sin[x5] - 0.00105997 Sin[x6]
FOTNR	(521.805 - 0.658342 Cos[x1] - 154.853 Cos[x2] - 7.40146 Cos[x3] - 0.383782 Cos[x4] - 264.627 Cos[x5] + 12.1588 Cos[x6] + 6.0395 Sin[x1] - 92.9585 Sin[x2] + 1.93934 Sin[x3] + 0.45789 Sin[x4] + 258.447 Sin[x5] - 53.9417 Sin[x6])/(209.295 - 30.6182 Cos[x1] - 19.9562 Cos[x2] - 659.281 Cos[x3] - 18.054 Cos[x4] - 26.731 Cos[x5] - 31.8661 Cos[x6] + 285.06 Sin[x1] + 80.5488 Sin[x2] + 222.881 Sin[x3] + 21.534 Sin[x4] + 7.92133 Sin[x5] + 147.079 Sin[x6])
SOTN	0.000594701 - 0.00348185 Cos[x1] - 0.0032836 Cos[x1]^2 + 0.000676033 Cos[x2] - 0.00570426 Cos[x1] Cos[x2] + 0.000765626 Cos[x2]^2 + 0.00108489 Cos[x3] - 0.00034799 Cos[x1] Cos[x3] + 0.00194208 Cos[x2] Cos[x3] + 0.0063544 Cos[x3]^2 + 0.0000965807 Cos[x4] - 0.0000768098 Cos[x1] Cos[x4] + 0.001202 Cos[x2] Cos[x4] + 0.000161927 Cos[x3] Cos[x4] + 0.000955877 Cos[x4]^2 + 0.000294638 Cos[x5] - 0.00403937 Cos[x1] Cos[x5] - 0.000303985 Cos[x2] Cos[x5] + 0.00118531 Cos[x3] Cos[x5] - 0.0000727783 Cos[x4] Cos[x5] + 0.000167498 Cos[x5]^2 + 0.000130554 Cos[x6] + 0.00114761 Cos[x1] Cos[x6] + 0.000695196 Cos[x2] Cos[x6] - 0.000282363 Cos[x3] Cos[x6] + 0.0000953567 Cos[x4] Cos[x6] + 0.00025966 Cos[x5] Cos[x6] + 0.000896525 Cos[x6]^2 - 0.000779027 Sin[x1] + 0.0121979 Cos[x1] Sin[x1] - 0.00262016 Cos[x2]

	Sin[x1] + 0.000387368 Cos[x3] Sin[x1] - 0.0000170856 Cos[x4] Sin[x1] + 0.000279204 Cos[x5] Sin[x1] - 0.000436528 Cos[x6] Sin[x1] + 0.000705358 Sin[x1]^2 + 0.00122577 Sin[x2] + 0.00036224 Cos[x1] Sin[x2] + 0.00140737 Cos[x2] Sin[x2] - 0.000820714 Cos[x3] Sin[x2] - 0.0045316 Cos[x4] Sin[x2] + 0.00336406 Cos[x5] Sin[x2] - 0.00209267 Cos[x6] Sin[x2] + 0.00589843 Sin[x1] Sin[x2] + 0.00246169 Sin[x2]^2 - 0.000317375 Sin[x3] - 0.000810762 Cos[x1] Sin[x3] - 0.000281901 Cos[x2] Sin[x3] - 0.0192636 Cos[x3] Sin[x3] + 0.0000406663 Cos[x4] Sin[x3] - 0.000488716 Cos[x5] Sin[x3] + 0.000160102 Cos[x6] Sin[x3] + 0.00247633 Sin[x1] Sin[x3] - 0.000996737 Sin[x2] Sin[x3] - 0.00556659 Sin[x3]^2 + 0.000149424 Sin[x4] + 0.00022882 Cos[x1] Sin[x4] + 0.000502096 Cos[x2] Sin[x4] + 0.0000206499 Cos[x3] Sin[x4] + 0.000145666 Cos[x4] Sin[x4] + 0.000227825 Cos[x5] Sin[x4] - 0.0000933241 Cos[x6] Sin[x4] - 0.000019137 Sin[x1] Sin[x4] - 0.00112769 Sin[x2] Sin[x4] + 0.0000769021 Sin[x3] Sin[x4] + 0.000721703 Sin[x4]^2 - 0.00108506 Sin[x5] + 0.00382805 Cos[x1] Sin[x5] - 0.00192553 Cos[x2] Sin[x5] - 0.00127571 Cos[x3] Sin[x5] - 0.000309146 Cos[x4] Sin[x5] - 0.00442382 Cos[x5] Sin[x5] - 0.0000272618 Cos[x6] Sin[x5] + 0.00213706 Sin[x1] Sin[x5] + 0.000764608 Sin[x2] Sin[x5] + 0.000217357 Sin[x3] Sin[x5] - 0.000106152 Sin[x4] Sin[x5] + 0.00110106 Sin[x5]^2 + 0.000805154 Sin[x6] - 0.00451633 Cos[x1] Sin[x6] + 0.000990251 Cos[x2] Sin[x6] + 0.00138843 Cos[x3] Sin[x6] + 0.000141334 Cos[x4] Sin[x6] + 0.000424864 Cos[x5] Sin[x6] + 0.000404533 Cos[x6] Sin[x6] - 0.00109003 Sin[x1] Sin[x6] + 0.00133502 Sin[x2] Sin[x6] - 0.000400726 Sin[x3] Sin[x6] + 0.000184659 Sin[x4] Sin[x6] - 0.0014428 Sin[x5] Sin[x6] + 0.00100192 Sin[x6]^2
SOTNR	(130.863 + 51.7896 Cos[x1] - 191.459 Cos[x1]^2 - 115.963 Cos[x2] - 376.814 Cos[x1] Cos[x2] + 265.115 Cos[x2]^2 - 157.274 Cos[x3] + 4.94363 Cos[x1] Cos[x3] + 47.9461 Cos[x2] Cos[x3] + 81.7487 Cos[x3]^2 - 475.734 Cos[x4] - 0.252307 Cos[x1] Cos[x4] + 424.925 Cos[x2] Cos[x4] + 1.61089 Cos[x3] Cos[x4] - 810.356 Cos[x4]^2 + 262.164 Cos[x5] - 514.997 Cos[x1] Cos[x5] + 426.416 Cos[x2] Cos[x5] - 29.2498 Cos[x3] Cos[x5] + 317.216 Cos[x4] Cos[x5] - 443.341 Cos[x5]^2 + 15.0345 Cos[x6] - 267.364 Cos[x1] Cos[x6] - 146.622 Cos[x2] Cos[x6] - 33.376 Cos[x3] Cos[x6] + 23.3673 Cos[x4] Cos[x6] + 27.0311 Cos[x5] Cos[x6] + 375.396 Cos[x6]^2 + 200.336 Sin[x1] + 0.225319 Cos[x1] Sin[x1] - 155.51 Cos[x2] Sin[x1] - 5.23589 Cos[x3] Sin[x1] - 2.96997 Cos[x4] Sin[x1] - 263.331 Cos[x5] Sin[x1] - 76.8791 Cos[x6] Sin[x1] - 199.017 Sin[x1]^2 + 203.341 Sin[x2] - 234.901 Cos[x1] Sin[x2] - 426.557 Cos[x2] Sin[x2] + 18.0461 Cos[x3] Sin[x2] + 253.655 Cos[x4] Sin[x2] + 115.763 Cos[x5] Sin[x2] - 158.2 Cos[x6] Sin[x2] - 99.6048 Sin[x1] Sin[x2] + 130.673 Sin[x2]^2 - 290.174 Sin[x3] - 20.4259 Cos[x1] Sin[x3] + 19.2629 Cos[x2] Sin[x3] + 59.1163 Cos[x3] Sin[x3] - 3.49537 Cos[x4] Sin[x3] + 481.228 Cos[x5] Sin[x3] + 89.1715 Cos[x6] Sin[x3] + 37.4245 Sin[x1] Sin[x3] + 57.4201 Sin[x2] Sin[x3] + 89.8387 Sin[x3]^2 - 75.6927 Sin[x4] + 0.234044 Cos[x1] Sin[x4] + 47.9025 Cos[x2] Sin[x4] - 0.490754 Cos[x3] Sin[x4] + 93.3756 Cos[x4] Sin[x4] + 403.305 Cos[x5] Sin[x4] + 59.7579 Cos[x6] Sin[x4] + 2.61721 Sin[x1] Sin[x4] + 29.7614 Sin[x2] Sin[x4] + 1.85513 Sin[x3] Sin[x4] - 12.3069 Sin[x4]^2 - 149.644 Sin[x5] + 505.642 Cos[x1] Sin[x5] + 154.266 Cos[x2] Sin[x5] + 29.5457 Cos[x3] Sin[x5] - 310.304 Cos[x4] Sin[x5] + 539.809 Cos[x5] Sin[x5] + 88.5108 Cos[x6] Sin[x5] + 257.748 Sin[x1] Sin[x5] + 238.817 Sin[x2] Sin[x5] - 474.223 Sin[x3] Sin[x5] - 394.492 Sin[x4] Sin[x5] - 112.124 Sin[x5]^2 - 206.984 Sin[x6] + 1186.57 Cos[x1] Sin[x6] + 2.28546 Cos[x2] Sin[x6] + 147.401 Cos[x3] Sin[x6] - 103.916 Cos[x4] Sin[x6] - 66.2595 Cos[x5] Sin[x6] + 430.311 Cos[x6] Sin[x6] + 340.089 Sin[x1] Sin[x6] + 300.659 Sin[x2] Sin[x6] - 390.1 Sin[x3] Sin[x6] - 265.69 Sin[x4] Sin[x6] - 446.824 Sin[x5] Sin[x6] + 6.80533 Sin[x6]^2)/(-144.042 - 14.2756 Cos[x1] + 541.757 Cos[x1]^2 + 139.561 Cos[x2] + 539.411 Cos[x1] Cos[x2] - 207.785 Cos[x2]^2 + 135.105 Cos[x3] + 168.523 Cos[x1] Cos[x3] - 67.698 Cos[x2] Cos[x3] - 84.2058 Cos[x3]^2 + 503.809 Cos[x4] - 56.5906 Cos[x1] Cos[x4] - 298.418 Cos[x2] Cos[x4] + 75.8123 Cos[x3] Cos[x4] + 146.434 Cos[x4]^2 - 370.64 Cos[x5] + 581.655 Cos[x1] Cos[x5] - 1000.96 Cos[x2] Cos[x5] + 8.4081 Cos[x3] Cos[x5] - 50.9368 Cos[x4] Cos[x5] + 1302.36 Cos[x5]^2 - 16.8563 Cos[x6] + 237.105 Cos[x1] Cos[x6] + 153.066 Cos[x2] Cos[x6] + 46.1848 Cos[x3] Cos[x6] - 48.2377 Cos[x4] Cos[x6] + 43.3201 Cos[x5] Cos[x6] - 377.309 Cos[x6]^2 - 134.793 Sin[x1] - 43.5692 Cos[x1] Sin[x1] - 0.841758 Cos[x2] Sin[x1] - 17.7024 Cos[x3] Sin[x1] - 129.091 Cos[x4] Sin[x1] + 264.792 Cos[x5] Sin[x1] + 87.8109 Cos[x6] Sin[x1] + 186.966 Sin[x1]^2 - 191.229 Sin[x2] + 120.926 Cos[x1] Sin[x2] + 349.713 Cos[x2] Sin[x2] - 334.6 Cos[x3] Sin[x2] - 563.28 Cos[x4] Sin[x2] - 422.28 Cos[x5] Sin[x2] + 157.352 Cos[x6] Sin[x2] + 595.002 Sin[x1] Sin[x2] - 666.984 Sin[x2]^2 + 305.494 Sin[x3] - 647.752 Cos[x1] Sin[x3] - 156.475 Cos[x2] Sin[x3] - 29.5675 Cos[x3] Sin[x3] - 88.3266 Cos[x4] Sin[x3] - 168.978 Cos[x5] Sin[x3] - 54.4033 Cos[x6] Sin[x3] + 1476.62 Sin[x1] Sin[x3] + 606.271 Sin[x2] Sin[x3] - 1105.45 Sin[x3]^2 + 87.6741 Sin[x4] + 21.2978 Cos[x1] Sin[x4] - 104.675 Cos[x2] Sin[x4] - 18.0114 Cos[x3] Sin[x4] -

	93.1117 Cos[x4] Sin[x4] - 425.158 Cos[x5] Sin[x4] - 55.8397 Cos[x6] Sin[x4] + 220.599 Sin[x1] Sin[x4] - 42.9454 Sin[x2] Sin[x4] + 66.8419 Sin[x3] Sin[x4] + 0.33696 Sin[x4]^2 - 122.299 Sin[x5] - 471.847 Cos[x1] Sin[x5] - 160.241 Cos[x2] Sin[x5] + 23.4858 Cos[x3] Sin[x5] + 40.3804 Cos[x4] Sin[x5] - 408.542 Cos[x5] Sin[x5] - 111.893 Cos[x6] Sin[x5] - 219.533 Sin[x1] Sin[x5] - 56.6442 Sin[x2] Sin[x5] - 100.607 Sin[x3] Sin[x5] + 416.095 Sin[x4] Sin[x5] + 80.5909 Sin[x5]^2 + 177.491 Sin[x6] - 1179.24 Cos[x1] Sin[x6] + 13.8309 Cos[x2] Sin[x6] - 218.386 Cos[x3] Sin[x6] + 217.155 Cos[x4] Sin[x6] - 67.5296 Cos[x5] Sin[x6] - 416.608 Cos[x6] Sin[x6] - 399.446 Sin[x1] Sin[x6] - 631.976 Sin[x2] Sin[x6] + 556.738 Sin[x3] Sin[x6] + 249.942 Sin[x4] Sin[x6] + 342.254 Sin[x5] Sin[x6] - 58.3288 Sin[x6]^2)
FOLN	0.284769 + 0.00770563 Log[x1] + 0.0060417 Log[x2] - 0.0272493 Log[x3] - 0.00133922 Log[x4] + 0.000920081 Log[x5] + 0.000959436 Log[x6]
FOLNR	(75.0778 +3.80553 Log[x1]+1.9118 Log[x2]-7.55979 Log[x3]-0.474654 Log[x4]+1.38162 Log[x5]+0.588373 Log[x6])/(-280.143+117.894 Log[x1]+11.1395 Log[x2]+25.9502 Log[x3]-5.70269 Log[x4]+81.947 Log[x5]+24.148 Log[x6])
SOLN	0.456164 - 0.0811424 Log[x1] - 0.00555601 Log[x1]^2 - 0.267811 Log[x2] - 0.00207178 Log[x1] Log[x2] - 0.276463 Log[x2]^2 - 0.102162 Log[x3] + 0.0102048 Log[x1] Log[x3] + 0.00158916 Log[x2] Log[x3] + 0.00260275 Log[x3]^2 - 0.00460199 Log[x4] + 0.000613469 Log[x1] Log[x4] - 0.00282638 Log[x2] Log[x4] - 0.000558676 Log[x3] Log[x4] - 0.000222135 Log[x4]^2 + 0.0371219 Log[x5] - 0.00306207 Log[x1] Log[x5] - 0.0114943 Log[x2] Log[x5] + 0.00639759 Log[x3] Log[x5] + 0.00260481 Log[x4] Log[x5] - 0.019152 Log[x5]^2 + 0.0583619 Log[x6] - 0.00106215 Log[x1] Log[x6] - 0.00339589 Log[x2] Log[x6] + 0.00210753 Log[x3] Log[x6] + 0.000768785 Log[x4] Log[x6] - 0.00040434 Log[x5] Log[x6] - 0.0149543 Log[x6]^2
SOLNR	(41580.9 + 907.344 Log[x1] + 4.30143 Log[x1]^2 - 328227. Log[x2] + 22.7015 Log[x1] Log[x2] - 273913. Log[x2]^2 - 3303.06 Log[x3] - 62.5574 Log[x1] Log[x3] - 93.3238 Log[x2] Log[x3] + 134.221 Log[x3]^2 - 1037.29 Log[x4] - 39.4094 Log[x1] Log[x4] - 56.3367 Log[x2] Log[x4] + 89.2752 Log[x3] Log[x4] + 7.7477 Log[x4]^2 - 61096.4 Log[x5] + 26.8186 Log[x1] Log[x5] - 25.9711 Log[x2] Log[x5] - 70.7112 Log[x3] Log[x5] - 6.06532 Log[x4] Log[x5] + 9338.89 Log[x5]^2 - 10917.9 Log[x6] + 10.9779 Log[x1] Log[x6] - 4.29195 Log[x2] Log[x6] - 27.7523 Log[x3] Log[x6] - 4.3311 Log[x4] Log[x6] + 11.7694 Log[x5] Log[x6] + 1963.33 Log[x6]^2)/(77880.4 + 24929. Log[x1] + 675.417 Log[x1]^2 - 18929. Log[x2] + 1259.7 Log[x1] Log[x2] - 60119.3 Log[x2]^2 - 6755.87 Log[x3] - 1351.74 Log[x1] Log[x3] - 2834.73 Log[x2] Log[x3] - 285.376 Log[x3]^2 - 14676.2 Log[x4] - 1453.69 Log[x1] Log[x4] - 2343.88 Log[x2] Log[x4] + 653.814 Log[x3] Log[x4] + 393.43 Log[x4]^2 + 16473.9 Log[x5] + 680.281 Log[x1] Log[x5] - 1045.45 Log[x2] Log[x5] + 125.401 Log[x3] Log[x5] - 549.871 Log[x4] Log[x5] - 1762.11 Log[x5]^2 + 12907.1 Log[x6] + 440.03 Log[x1] Log[x6] + 19.2606 Log[x2] Log[x6] - 101.058 Log[x3] Log[x6] - 251.748 Log[x4] Log[x6] + 533.558 Log[x5] Log[x6] - 1826.67 Log[x6]^2)

Curriculum Vitae

Name Surname : Ümmühan Gümüş

Education:

2012 – 2017 B.Sc. / Izmir Katip Celebi University, Dept. of Mechanical Eng.

2018 – 2022 MSc. / Izmir Katip Celebi University, Dept. of Mechanical Eng.

Work Experience:

2018 – 2018 Is Yapi Co. – Quality Control Engineer – İzmir

2019 – 2021 Ege Profil Tic. San. A.Ş. – R&D Engineer – İzmir

Publications:

1. Aydin L, Artem HS, Oterkus S. Designing Engineering Structures using Stochastic Optimization Methods. 1st edition. CRC Press; 2020 https://doi.org/10.1201/9780429289576