

Optimization of Drying Process of Plastic Granules Based on Stochastic Method with Neuro Regression 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

Mustafa DİNÇ

ORCID 0000-0003-1775-4549

July, 2023

This is to certify that we have read the thesis **Optimization of Drying Process of Plastic Granules Based on Stochastic Method with Neuro Regression Approach** submitted by **Mustafa Dinç**, 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:

Assoc. Prof. Dr. S. Bahar Bozkurt
Manisa Celal Bayar University

Assist. Prof. Dr. Ebubekir Atan
İzmir Kâtip Çelebi University

Date of Defense: July 20, 2023

Declaration of Authorship

I, **Mustafa Dinç**, declare that this thesis titled **Optimization of Drying Process of Plastic Granules Based on Stochastic Method with Neuro Regression 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 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.

Date: 20.07.2023

Optimization of Drying Process of Plastic Granules Based on Stochastic Method with Neuro Regression Approach

Abstract

Polymers contain moisture due to their chemical structure. Moisture in polymers is important for the quality of the final product and ease of processing. For this reason, polymers are subjected to drying before processes such as injection and extrusion. Drying is carried out at temperatures and times specified by the polymer suppliers. Over-drying leads to increased viscosity (processing difficulty) while under-drying leads to flow marks, burrs, and reduced tensile strength and impact strength. Over-drying also causes excessive energy consumption. In this study, the drying process optimization of Acrylonitrile Butadiene Styrene (ABS) polymer was studied. As the first part of the data collection part, different scenarios were determined with the Design of Experiment (DoE) approach. For these scenarios, drying time and temperature values reported by ABS polymer manufacturers were chosen as design parameters. ABS granules were dried with a desiccant type dryer at different temperatures. For the outputs, samples were taken at certain periods and the moisture by weight and energy consumption of the samples were determined by energy analyzer and moisture meter. In the second part, Wolfram Mathematica program is used for numerical operations. Mathematical models were constructed using nonlinear multiple neuro-regression modeling to describe the phenomenon between input and output parameters. R^2 model evaluation criterion was used to test the success of the proposed models in the training, testing and verification stages. The engineering limits of the models were checked and the model that gave the best results was selected. Finally,

optimization of system parameters was performed using different stochastic optimization methods Differential Evolution, Nelder–Mead, Simulated Annealing and Random Search.

Keywords: ABS, drying process, neuro-regression approach, stochastic optimization, energy consumption

Nöro Regresyon Yaklaşımı Kullanılarak Stokastik Metot Tabanlı Plastik Granüllerin Kurutma Prosesi Optimizasyonu

ÖZ

Polimerler kimyasal yapısı dolayısıyla nem içermektedir. Polimerlerde bulunan nem son ürünün kalitesi ve proses kolaylığı açısından önem arz etmektedir. Bu sebeple polimerler enjeksiyon, ekstrüzyon gibi proseslerden önce kurutma işlemine tabii tutulurlar. Kurutma işlemi polimer tedarikçileri tarafından belirtilen sıcaklık ve sürelerde gerçekleştirilir. Fazla kurutma viskozite artışına (proses etme zorluğuna), az kurutma ise akış izleri, çapak, çekme dayanımı ve darbe dayanımı düşüşüne yol açmaktadır. Bunlara ek olarak, fazla kurutma işlemi, fazla enerji tüketimine sebebiyet vermektedir. Bu çalışmada, Akrilonitril Bütadien Stiren (ABS) polimerinin kurutma proses optimizasyonu üzerine çalışma gerçekleştirilmiştir. Veri toplama bölümünde ilk kısım olarak Deney Tasarımı (DoE) yaklaşımıyla farklı senaryolar belirlenmiştir. Bu senaryolar için, ABS polimer üreticileri tarafından bildirilen kurutma zamanı ve sıcaklık değerleri dizayn parametresi olarak seçilmiştir. ABS granüller farklı sıcaklıklarda desikant tipi kurutucu ile kurutma işlemine tabii tutulmuştur. Çıktılar için, belirli periyotlarda numune alınmış, enerji analizatörü ve nem ölçüm cihazı ile numunelerin ağırlıkça nemi ve enerji tüketimi tespit edilmiştir. İkinci bölümde, sayısal işlemler için Wolfram Mathematica programı kullanılmıştır. Doğrusal olmayan çoklu nonlinear nöro-regresyon modellemesi kullanılarak girdi ve çıktı parametreleri arasındaki fenomeni tanımlamak için matematiksel modeller oluşturulmuştur. Önerilen modellerin eğitim, test ve doğrulama aşamalarındaki başarılarını test etmek için R^2 model değerlendirme kriteri kullanılmıştır.

Modellerin mhendislik sınırları kontrol edilmiř ve en iyi sonucu veren model seilmiřtir. Son olarak, farklı stokastik optimizasyon yntemleri, Differential Evolution, Nelder–Mead, Simulated Annealing ve Random Search, kullanılarak sistem parametrelerinin optimizasyonu gerekleřtirilmiřtir

Anahtar Kelimeler: ABS, kurutma prosesi, nro-regresyon yaklařımı, stokastik optimizasyon, enerji tketimi

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 dear fiancé Ece Orancı for her caring, support, and motivation.

Table of Contents

Declaration of Authorship.....	ii
Abstract	iii
Öz.....	v
Acknowledgment	viii
List of Figures	xii
List of Tables.....	xiii
List of Abbreviations.....	xiv
List of Symbols	xv
1. Introduction.....	1
1.1. Literature Survey	1
1.2. Objectives and Motivation	2
2. Mathematical Background.....	4
2.1. Introduction	4
2.2. Design of Experiments (DOE)	4
2.2.1. Randomized Complete Block Design	5
2.2.2. Full Factorial Design	6
2.2.3. Fractional Factorial Design.....	6
2.2.4. Central Composite Design.....	7
2.2.5. Box-Behnken Design	7

2.2.6.	Taguchi Design	7
2.2.7.	Latin Hypercube Design	8
2.2.8.	Optimal Design (D-Optimal).....	8
2.3.	Mathematical Modeling	9
2.3.1.	Neuro Regression Approach	10
2.3.2.	Nonlinear Regression Analysis	11
3.	Optimization Process.....	13
3.1.	Stochastic Optimization Methods	13
3.1.1.	Introduction	13
3.1.2.	Simulated Annealing	16
3.1.3.	Differential Evolution	17
3.1.4.	Nelder Mead	18
3.1.5.	Random Search	19
3.2.	Mathematica and Optimization	21
3.2.1.	Global and Local Optimization by Mathematica.....	21
3.2.2.	FindMinimum	23
3.2.3.	NMinimize and NMaximize Functions	25
3.3.4.	Random Search Solver	26
3.2.5.	Simulated Annealing Solver	31
3.2.6.	Nelder Mead Solver	36
4.	Optimization of Drying Process of Plastic Granules	40

4.1.	Methods	40
4.1.1.	NDSolver Solver	40
4.1.2.	FindFit Solver	41
4.2.	Engineering Model	42
4.2.1.	Mathematical Model.....	43
4.3.	Materials and Methods	49
4.4.	Optimization Problem Definition	50
4.5.	Optimization Scenarios	51
4.6.	Result and Discussion	52
4.7.	Conclusion.....	59
	References	60
	Appendices.....	66
	Appendix A Draft Design Chart	66
	Appendix B Expanded Model Expressions of the Case Studies – Energy Output	68
	Appendix C Expanded Model Expressions of the Case Studies – Moisture Output	73
	Appendix D Results of the Neuro-regression models boundedness.....	78
	Curriculum Vitae	80

List of Figures

Figure 2.1	Flowchart illustrating the processes in an ideal design process	4
Figure 3.1	Flowchart of the simulated annealing algorithm.....	17
Figure 3.2	Flowchart of the differential evolution algorithm	18
Figure 3.3	Flowchart of the nelder–mead algorithm	19
Figure 3.4	Flowchart of the random search	20
Figure 3.5	Mathematica optimization process.....	23
Figure 3.6	3D plot of the function $f(x_1, x_2)$	24
Figure 3.7	Flowchart of the random search algorithm	28
Figure 3.8	3D plot of Ackley function in an interval... ..	28
Figure 3.9	“Holder Table 1” function's 3D plot... ..	30
Figure 3.10	Flowchart of the simulated annealing algorithm... ..	32
Figure 4.1	Illustrating of the polymer drying system... ..	42
Figure 4.2	Energy Analyzer Interface... ..	43
Figure 4.3	Environmental Values Measurement Sensor... ..	44
Figure 4.4	Shini Dew Point Sensor.....	44
Figure 4.5	Environmental Values Measurement Sensor... ..	44

List of Tables

Table 3.1	Methods and instructions for optimization.....	22
Table 4.1	Design Variables	43
Table 4.2	Drying experiment parameters for 65°C	46
Table 4.3	Drying experiment parameters for 75°C	47
Table 4.4	Drying experiment parameters for 85°C	48
Table 4.5	Multiple regression model names, nomenclatures and formulas	49
Table 4.6	Objective Function	52
Table 4.7	Accuracy check results of obtained models for the Energy output in terms of R2 values	53
Table 4.8	Accuracy check results of obtained models for the Moisture output in terms of R2 values	55
Table 4.9	Optimization Problem Results	57

List of Abbreviations

ODE	Ordinary Differential Equation
DE	Differential Evolution
SA	Simulated Annealing
NM	Nelder–Mead
RS	Random Search
FE	Finite Element
LP	Linear Programming
NLP	Nonlinear Programming
KKT	Karush–Kuhn–Tucker Conditions

List of Symbols

E	Energy [kWh]
C	Temperature [centigrade]
t	Time [min]
M	Moisture [%]
k	The Constant of Boltzmann
$P(E)$	The Probability of Achieving the Energy Level

Chapter 1

1. Introduction

1.1. Literature Survey

Plastic injection molding is a widely used method for producing plastic products [1]. It involves melting the material with the aid of a screw and an external heating device, and then injecting it into a mold to form the desired product as the mold cools [2]. Injection molding has been applied in various fields, from daily necessities to aerospace components [3]. The process offers advantages such as the reduction of density and elimination of sink marks, and mold designers are using Finite Element Method (FEM) for optimization [4]. Different parameters, such as injection pressure, holding pressure, and injection speed, can affect the quality and dimensional stability of the plastic part. To ensure high-quality products and reduce dependency on human expertise, artificial intelligence approaches are being used to model the injection molding process [5].

Plastic injection molding processes involve several steps, including drying treatment of plastic particles, melting the plastic, injecting the molten plastic into a mold cavity, cooling and solidification, and separating the formed plastic product from the mold [6].

Polymer drying can be achieved through various methods which one is first step of these steps. One method involves using a polymer plastic drying device, which includes a working cavity and a sliding support plate [7]. Another method involves adjusting the viscosity of a polymer solution and spray-drying it to obtain polymer powder with low water content [8]. Computer simulations have shown that polymer-polymer mixtures stratify into layers during drying, with shorter polymers enriched near the drying interface [9]. A non-drying polymer hydrogel can also be produced by containing a deliquescent substance in a polymer network, which prevents drying under atmospheric conditions [10]. Additionally,

a method for drying a polymer involves spraying a polymer-containing solution into a gas and then removing the solvent by introducing the droplets into a second solvent [11].

Polymer drying optimization is a continuing challenge in various industries. Different methods and parameters have been explored to achieve optimal conditions for polymer drying. Erfando et al. developed a genetic algorithm to optimize parameters such as injection rate, injection time, and injection pressure for polymer injection in oil recovery [12]. Tavcar et al. developed an optimization algorithm for polymer gears, considering criteria such as stress, temperature, wear, and cost [13]. Kurganov et al. proposed expressions to find optimal conditions for hydrodynamic chromatography analysis of polymers [14]. Al Momani and Örmeci investigated the use of an in-line UV-vis spectrophotometer to optimize polymer dose during sludge dewatering [15]. Marchetti et al. described a mathematical programming model for optimizing the balance of feedstocks to manufacture multiple polymer grades [16]. These studies highlight the importance of optimizing various parameters and criteria to achieve efficient polymer drying processes.

1.2. Objectives and Motivation

The reasons for choosing this thesis study can be listed as follows;

- Moisture in polymers is important for the quality of the final product and ease of processing.
- Polymers are subjected to drying before processes such as injection and extrusion.
- Over-drying leads to increased viscosity (processing difficulty) while under-drying leads to flow marks, burrs, and reduced tensile strength and impact strength.
- Over-drying also causes excessive energy consumption. The aim of engineering applications is to have the best mechanical properties and at the same time to minimize the cost.

- As it is an integral part of optimization life, it is frequently used and must be used in engineering applications in order to reduce the cost in the first place.

The aims of this study can be listed as follows;

- To create a mathematical model of the drying process parameters and the moisture content by weight of the ABS polymer,
- Reaching minimum energy consumption with optimum drying parameters,
- To investigate the effects of drying parameters on polymer moisture.

Chapter 2

2. Mathematical Background

2.1. Introduction

The background of the mathematical tools used throughout the thesis is introduced in this chapter. The primary goal of this chapter is to use the techniques employed in this study to briefly explain the procedures for design-based optimization in engineering. The design of experiments, mathematical modeling, and optimization techniques are discussed in this context. All of these processes for an ideal design process are shown in Figure 2.1. Additionally, the optimization techniques used in the Mathematica software, the instrument used to carry out the mathematical procedures in this study, are mentioned.

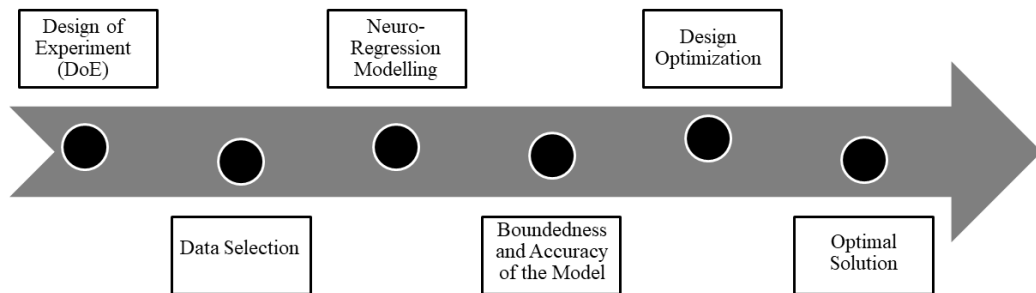


Figure 2.1: Flowchart illustrating the processes in an ideal design process

2.2. Design of Experiments (DOE)

In manufacturing processes, experiments are conducted to increase our knowledge and understanding of them. Hence, the correlations between the main factors of the inputs and the output behaviors can be observed [17, 18]. One-Variable-At-a-Time (OVAT) is one of the most common engineering techniques. The output from these operations could be inaccurate and wasteful as a result of this strategy. Furthermore, it is well recognized that not every parameter affects results equally. Therefore, a complex design seeks to establish the degree to which process

parameters affect the result [17, 19, 20]. Design of experiment (DoE) is the best course of action when a component's specific characteristic is influenced by a number of different factors [17, 18, 21]. DoE is a useful method for identifying new processes, getting a comprehensive understanding of current processes, and improving their efficiency. By reducing time and cost, this method ensures great efficiency and more reliable process results.

The choice of the best statistical tools is crucial since noise can significantly alter the outcomes of data analysis. The three guiding concepts of statistical techniques in DoE are replication, randomization, and blocking. In order to get more accurate results and reduce experimental error, replication is based on repeating experimental runs. The experiments are carried out in a random order according to the randomization process. Blocking is used to stop the main effects of vagueness and isolate known systematic bias impacts [17, 19]. The proper DoE approach is determined by the objectives of the experiments and the number of elements to be addressed. The remainder of this section lists and briefly describes some DoE methods, including Randomized Complete Block Design, Full Factorial, Fractional Factorial, Central Composite, Box-Behnken, Taguchi, Latin Hypercube, and D-Optimal Design [19]. It should be noted that the techniques presented here are not a complete list, as it is intended to inform readers about the subject by presenting approaches commonly used in practice.

2.2.1. Randomized Complete Block Design

The distribution of treatment for experimental constituents is not subject to any tight restrictions. However, there are instances where the data from studies changes greatly in the real world. The design created in such circumstances is known as a Randomized Complete Block Design (RCBD). The main goal of blocking is to maximize variance between blocks while minimizing variation between experimental units within a block.

Advantages

- It is possible to remove the treatments or replicates from the analysis.
- More frequently than others, some multiple treatments can be repeated.
- There are no strict limitations on the quantity of treatments or replicates.
- Despite the fact that the experimental error is not homogeneous, there can be meaningful comparisons. [17, 22].

Disadvantages

- For a small number of treatments, the *df* error is less.
- If there are too many treatments and significant differences between experimental units, a substantial error term may be acquired.

When data are lacking, RCBD performs poorly in terms of experimental effectiveness.

2.2.2. Full Factorial Design

In industries like manufacturing, factorial designs with two or three levels are typically acknowledged as the most frequently used DoE approach since they may help generate reliable data on the effects of the variables. There are two types of factorial designs: full and fractional. In a full factorial design, each factor setting is combined with every other factor setting to identify the experimental runs. Full factorial design involves a lot of runs and is not very useful if the concerned response is influenced by five or more factors. The fractional factorial design may be advantageous in such cases [17, 18].

2.2.3. Fractional Factorial Design

In practice, there is typically not enough time or funding to conduct tests using a full factorial design. The main influences and two-order interactions can be obtained by using a portion of the full factorial experiment, which cuts down on the time and expense of the experiments in cases where some higher-order correlations are not necessary. With a minimal number of workouts or

experimental runs, fractional factorial design, a type of orthogonal array layout, enables researchers to examine the most crucial and necessary effects of relationships [17, 18, 22, 23].

2.2.4. Central Composite Design

One of the most popular response surface designs, central composite design (CCD), generates a factorial design and is carried out using five factorial levels. One of the most significant benefits of CCD is the ability to check the corner points. Therefore, if the curvature is negligible, the task is completed. On the other hand, the main task is to create the star runs if the curvature is significant [21].

2.2.5. Box-Behnken Design

A popular DoE method that works with three factorial levels is Box-Behnken design. There are fewer runs as a result of this method's reliance on the midpoints rather than the cube edge corner points. With the exception of CCD, all runs must be completed in Box Behnken. Furthermore, it is superior in which cases the curvature specified in the screening experiment is likely necessary [17].

2.2.6. Taguchi Design

Taguchi design is a statistical technique that significantly improves engineering productivity. The primary objective of the Taguchi method is to keep output fluctuation to a minimum, even in the presence of noise. Therefore, by taking into account the noise causes and the error amount, this technique helps to ensure product quality. Additionally, Taguchi design concentrates on enhancing the fundamental purpose of the design process, allowing for the presentation of flexible designs [17, 22].

Advantages

- It is a versatile tool since it is simple to adapt to different engineering problems.

- It improves the product quality, within some qualification constraints, by considering a mean production feature value comparable to the final one rather than just a value.
- It makes it possible to study the various variables without performing an excessively high number of runs.

Disadvantages

- The obtained results are comparative and do not accurately identify which parameter has the greatest influence on the desired attribute.
- Because orthogonal arrays only consider specific parameter combinations, it cannot be utilized to determine how all variables interact.
- Parameter interactions are hard to be considered.

It is offline, which makes it inconvenient for processes involving dynamic changes, such those seen in computer simulations.

2.2.7. Latin Hypercube Design

This approach uses a Multidimensional Distribution to generate a nearly random sample of parameter values. Additionally, Latin Hypercube design is a generalization of the Latin Square concept to an arbitrary number of dimensions. The initial stage in this method is to decide how many sample points will be used, along with which row and column each sample point was used for. Additionally, whereas standard random sampling only yields a set of random numbers without assurance, this approach guarantees a set of random numbers that accurately reflect the fluctuation [17, 19].

2.2.8. Optimal Design (D-Optimal)

D-Optimal is a computer-aided design. It contains the best part of all possible experiments. The final design might vary based on the tool used so software tools may have diverse processes to create D-Optimal designs [17, 23]. Based on a predefined factor and the number of runs, the selection technique produces the

optimal design. When conventional design methodologies are not used, the D-optimal design approach is very useful. These cases are:

- When supplies of factor configurations are restrained.
- If the number of experimental runs must be reduced.
- When using the operation and mixing variables in the same design.
- When previously performed experiments must be comprised.
- In the event that the experimental area is unstable [17, 20, 21].

2.3. Mathematical Modeling

Mathematical modeling plays a key role in design-based engineering optimization studies to obtain a robust objective function for the problem. Therefore, it's crucial to use a suitable data modeling methodology in order to precisely define the phenomenon under consideration. Researchers use many modeling techniques in this way, including Artificial Neural Networks (ANN), Finite Difference Technique (FDT), Response Surface Methodology (RSM), and Regression Analysis. Nevertheless, studies preferred popular methods regarding engineering optimization have some inadequate approaches, as follows:

(i) Updating one input while keeping the others constant is not a satisfying explanation because it ignores the nonlinear effects of input variables. Therefore, from the perspective of optimization, it is necessary to take into account the interactions between all experimental and constructional parameters.

(ii) The most of data modeling methods involve using one or two traditional regression models as the problem's objective function. In order to calculate how closely the results of the fitted model match the experimental data, R^2 values are calculated. These values are used to assess the model's reliability. However, achieving a high R^2 value does not always mean a good fit for the engineering systems. In addition, the model describes only the experimental results rather than the fundamental behaviour of the phenomenon. So various regression model types and approaches should be taken into consideration.

(iii) Another important point is the model function boundedness. Since all engineering parameters are known to be finite, the function should be bounded in order to accurately represent engineering systems. Hence, the proposed model should be checked to determine whether it is limited to the parameter intervals of the respective phenomenon.

(iv) Although it is vital to unveil the inherent behaviors of the stochastic search processes, some studies on engineering systems optimization do not consider the reliability, sensitivity, and robustness of the algorithms.

In order to overcome the abovementioned deficiencies of the most widely used modeling and optimization approaches, it has been stated in the literature that it is possible to perform realistic engineering design optimization studies [24, 25]. In order to present a thorough view of the modeling-design-optimization processes, a multiple nonlinear neuro-regression analysis is introduced with the simultaneous use of four distinct direct search algorithms as part of the optimization process.

2.3.1. Neuro Regression Approach

The neuro-regression approach is a hybrid data modeling technique that improves prediction accuracy by combining the advantages of regression analysis and Artificial Neural Networks (ANN). This method begins by randomly splitting the total amount of data into three sets that will be used for training, testing and validation, respectively. By specifying the regression models and their coefficients, the training step aims to reduce the error between the experimental and predicted values. Accordingly, testing data are used to obtain the estimated results by minimizing the effects of regression model inconsistencies, and this step gives an understanding of the candidate models' prediction capability. R^2 values of the models are achieved in the training, testing and validation steps. A boundedness check is performed on appropriate models in terms of R^2 values as the approach's final step. In this regard, the minimum and maximum values in the specified interval for each design variable serve as the boundary of the candidate models. This step is essential to determine whether the models are applicable to the problem. As a result, it is expected that the selected models will satisfy each required robustness criterion.

2.3.2. Nonlinear Regression Analysis

Nonlinear regression models are those that do not have linear parameters. They can be used for three different purposes [26]:

- To test the validity of the model (or compare the hypothesis),
- To characterize the model (i.e., parameter prediction),
- To calculate the system's behavior (interpolation and calibration).

The general form of the nonlinear regression model is as follows:

$$y = f(x, \beta) + \varepsilon \quad (2.1)$$

where x , β , and ε are vector of p predictors, a vector of k parameters, and an error term, respectively. $f(-)$ represents a known regression function.

Mathematical modeling procedures for nonlinear regression can be carried out methodically following key characteristics as follows:

- Nonlinear regression allows more flexibility than linear regression because the function does not need to be linear or linearizable. As a result, nonlinear regression ensures different options to match the data.
- Nonlinear regression may be more applicable than transformations and linear regression in situations when the f function can be linearized.
- Nonlinear regression necessitates a knowledge of the f function (e.g., polynomial, trigonometric, exponential), which requires a comprehensive insight into the studied process. Although precise clarity is not necessary, linear regression models are suitable for process forecasts where the relationship between input and output parameters is approximately certain.

Mathematical terminology used in nonlinear regression models are the most general so functionally generalized states cannot be written. However, the following are some fundamental model types used in engineering fields as examples of nonlinear equations:

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (2.2)$$

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

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

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

Furthermore, a similar methodology can also be used to generate multivariable states with various inputs from the aforementioned model types. Another important point is that with a deeper understanding of mathematical functions, special functions (such as Bessel, Laguerre, Lambert, and Gamma) as well as various combinations of conventional functions can be selected.

Chapter 3

3. Optimization Process

3.1. Stochastic Optimization Methods

3.1.1. Introduction

One of the most common problem in applied mathematics is finding an approximate optimal solution for a function defined on a subset of finite-dimensional space. Some objective functions should be optimized in combinatorial optimization problems, which are essential for the majority of machine learning approaches, in order to approximate the optimal solution. For these optimization problems, there were many numerical optimization techniques available fifty years ago; the majority of them were deterministic (traditional optimization techniques). Stochastic methods, which employ non-traditional optimization techniques, have evolved into crucial tools for engineering, statistics, science and business as a result of the development of computer technology. These techniques are the most common due to certain properties that deterministic algorithms do not have [27, 28]. For example, stochastic methods always include probability, such as how much rain falls at random distribution in a reservoir, the periodic prediction of the water level, or predicting the number of dropped connections for a communications network using a constant bandwidth that is randomly variable. As opposed to that, deterministic methods include probability under no circumstances; and outcomes based on exact input values [29].

The process of minimizing or maximizing the value of a mathematical or statistical function when one or more input parameters are subject to random variables is known as stochastic optimization. The randomness may be noise in measurements, Monte Carlo randomness in the search process, or both [27, 28].

Many industrial, biological, economic, and engineering problems can be confirmed as stochastic systems, such as geography, communication area,

banking, signal processing, aerospace. 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. [30] suggested a combined qualitative and quantitative modeling specification based on a hierarchical model structure framework made up of the high-level model, the meta-meta model, and the meta-model. According to the findings of this study, the complex system could be completely described using the suggested methodology. Li and Zhang [31] studied the problem of stochastic linear-quadratic optimum control under final state inequality constraints. In this study, the Karush-Kuhn-Tucker (KKT) theorem was proved using hybrid constraints, then they obtained new types of Riccati equations. The optimal linear state feedback control existence resulting from the KKT theorem is provided by this equation. To solve the uncertain restricted stochastic linear quadratic problem, a dynamic programming algorithm design was achieved. The efficient global optimization method (EGO) was used by Aydn et al. [32] to research the design of dimensionally stable laminated composites. The high stiffness and low heat and moisture expansion coefficients optimization problem for composite plates was resolved. The experimental proof is provided for the optimization algorithm suggested in this study. Using Tsai-Hill, Hoffman, Tsai-Wu, and Hashin-Rotem criteria, failure analysis of the optimized composites was carried out after the design and optimization processes were finished. Zakaria et al. [33] thoroughly examined generic stochastic optimization steps for applications in renewable energy. The advantages and disadvantages of stochastic optimization were also emphasized. Significant optimization methods belonging to the stochastic optimization stages are emphasized.

The latest improvements and important stochastic optimization techniques were presented by Niamsup and Rajchakit [34] in their study. For the social, economic, and technical aspects of renewable energy systems, it is claimed that stochastic optimization methods are more effective than deterministic optimization techniques. Using the parameter-dependent Lyapunov-Krasovskii functional in combination with techniques for linear matrix inequality, Niamsup and Rajchakit examined polytopic discrete-time stochastic functions in the interval time-varying

delays and proposed new standards for the robust stability of the stochastic system.

In order to manage a fleet of bikes over a number of bike stations, each with a certain capacity and time-varying stochastic demand, Maggioni et al. [35] studied this problem. The optimal number of bikes to assign to each station at the beginning of the service is then determined using multi-stage and two-stage stochastic optimization models for one-track bike-sharing systems with transshipment. The solution supplied in the real system is compared to the solutions obtained using the two-stage and multi-stage models to provide managerial insights. In order to operate the hydrogen network of a petroleum refinery as efficiently as possible, Gutierrez et al. [36] studied how to cope with the difficulty of the indefinite scenario. The effect of raw network operating modifications was examined using a two-stage stochastic optimization method. Additionally, they were examined for viable stochastic and deterministic solutions to the hydrogen network problem using data from real plants.

For the manufacturing of carbon fiber during the carbonization process, Khayyam et al. [37] suggested a stochastic optimization model to reduce energy consumption in a proper range of fundamental mechanical properties. A total of fifty samples of fiber were analyzed for each set of processing operations, tensile strength, and modulus. The five distribution functions were examined to find the distribution functions that could most accurately describe the mechanical property distribution of filaments while the energy usage on the processing equipment was being tracked during the manufacturing of the samples. The Kolmogorov-Smirnov test was also performed to confirm the correlation statistics and distribution goodness of fit. The study demonstrated that, within the specified range, the production quality could be predicted using stochastic optimization models, and that this approach reduced the amount of energy used in the industrial process.

The thick epoxy/carbon fiber laminates were used to the stochastic multi-objective cure optimization technique established by Tifkitsis et al. [38]. A surrogate model was construct using the kriging approach, which replaces the Finite Element (FE) simulation for computational effectiveness. A stochastic multi-objective optimization framework based on Genetic Algorithms was constructed by

coupling and integrating the surrogate model and Monte Carlo. In comparison to normal cure profiles, the results showed a considerable reduction of 40% in temperature overshoot and curing time.

Stochastic optimization methods [27-30] include Simulated Annealing (SA), Differential Evolution (DE), Nelder-Mead (NM), and Random Search (RS). Researchers are constantly updating the literature with either new stochastic approaches, improvements, or both. Some commonly used stochastic optimization methods are briefly reviewed in the subsections that follow.

3.1.2. Simulated Annealing

Finding the global minimum of a function with a considerable number of independent variables can be done relatively effectively using the simulated annealing (SA) method, one of the most efficient and general stochastic optimization algorithms. Additionally, the SA method makes a comparison between the physical annealing process and determining the minimal function value in mixed-integer, discrete, or continuous minimization problems. The physical annealing procedure is referred to as a thermal process in condensed matter physics because it produces low energy states of a material in a heat bath. The fundamental idea behind the SA algorithm is to use random search in terms of a Markov chain, which not only accepts changes that advance the objective function but also retains some of the less-than-ideal changes.

The SA algorithm generates a new point at random after each iteration, and it ends when any stopping conditions are satisfied (Figure 3.1). The Boltzmann probability distribution with a scale based on temperature is used to determine the distance between the new and current point or the scope of the search. Equation 3.1 gives the definition of the Boltzmann Probability Distribution [28, 39-41] as

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

where,

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

k : The Constant of Boltzmann,

T : Temperature.

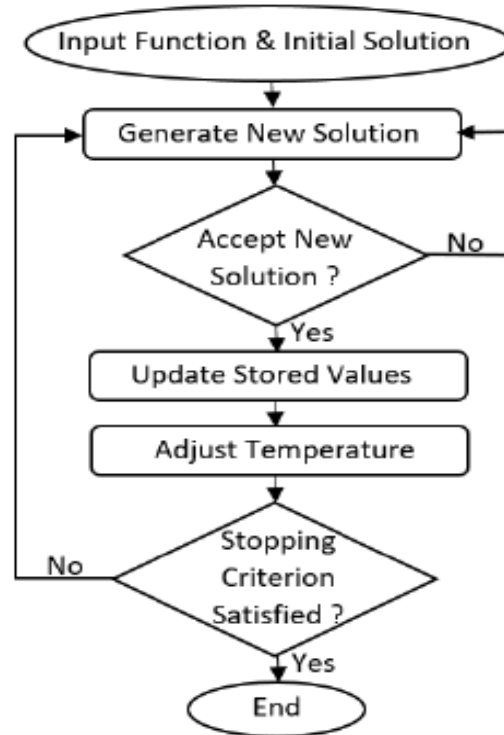


Figure 3.1: Flowchart of the simulated annealing algorithm [39]

3.1.3. Differential Evolution

Storn and Price first introduced Differential Evolution (DE) as a search method in 1996 for solving optimization issues over continuous domains. Currently, DE is among the most powerful real-parameter optimization algorithms. The four basic stages of this algorithm are selection, crossover, mutation, and initialization. Additionally, this algorithm has three real control parameters: (i) differentiation/mutation constant, (ii) crossover constant, and (iii) population size. The manipulation of the target and difference vectors to produce a trial vector is essential for the differential evolution performance. The DE algorithm also uses three control parameters: (i) the problem dimension, which scales the difficulty of the optimization case; (ii) the maximum number of generations known as a

stopping condition; and (iii) boundary constraints [28, 41]. Figure 3.2 is a flowchart that summarizes the operation of the DE algorithm.

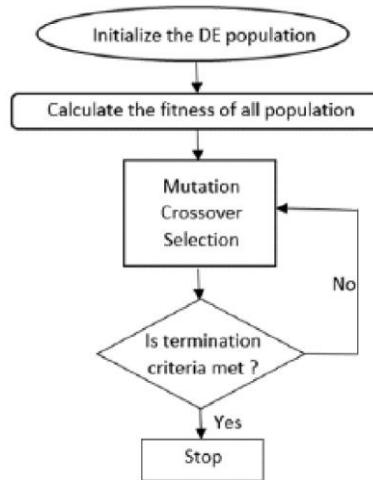


Figure 3.2: Flowchart of the differential evolution algorithm [42]

Similar operators are used by the differential evolution algorithm, a population-based algorithm like GA. The main difference between both algorithms is that GA depends on crossover, a method of valuable and probabilistic information flow across solutions to identify better solutions. DE, however, uses mutation operation as its main search mechanism. This fundamental operation is based on the variations between population pairs of solutions that were randomly sampled. Despite the fact that this method is numerically inefficient, DE is strong and effective enough to finding an optimum global value and avoid the local minimum regardless of initial points. [43-45].

3.1.4. Nelder Mead

The Nelder-Mead (NM) algorithm is additionally referred to as Simplex Search in the traditional literature on optimization. It is a traditional local search technique that was originally designed for unrestricted optimization problems [46]. Although NM is not a global optimization approach, it frequently performs admirably in the real world for problems with numerous local minima. Similar to the DE algorithm, the adjustment of the NM options is controlled by four fundamental parameters: reflection, expansion, contraction, and shrinkage. The

main characteristic of the NM algorithm is that the first few iterations produce satisfactory results.

Additionally, it is required that one or two function evaluations only are notably rare in practice for each iteration. The Simplex can change its orientation, size, and shape to adapt to the local contour of the objective function, avoiding costly or time-consuming evaluations of multiple functions. Furthermore, NM may explore complicated search spaces with a great flexibility. The main steps of the algorithm are shown in Figure 3.3.

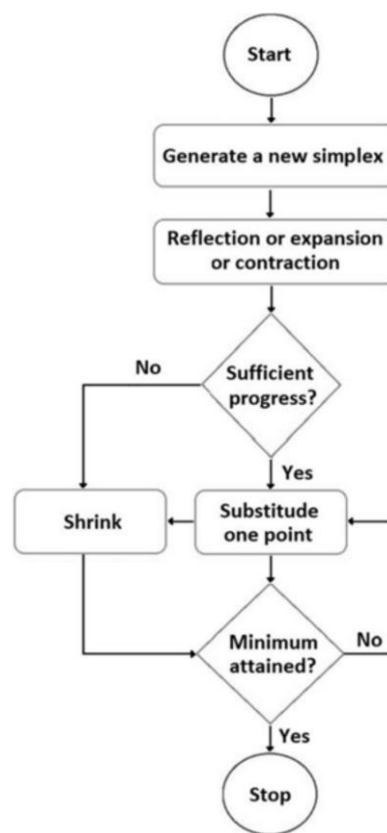


Figure 3.3: Flowchart of the nelder–mead algorithm [47]

3.1.5. Random Search

The Monte Carlo Method, commonly known as the Random Search (RS) algorithm, is founded on a stochastic methodology. It differs from deterministic techniques like Branch-Bound and Interval Analysis due to the stochastic

character of the algorithm. The main advantages of RS are that (I) when a multimodal function's absolute maximum is required, it should be simple to integrate with a true search process of some kind, (ii) it enables the global optimum to be approached for non-convex, non-differentiable objective functions with continuous and/or discontinuous domains, (iii) it is easy to apply for even the most difficult optimization problems, (iv) the RS technique is relatively stable and quickly offers fundamental information for unorganized global optimization problems. The process used by the examined RS method in this work is given in Figure 3.4. Furthermore, a thorough explanation of the Random Search approach is provided in [48, 49].

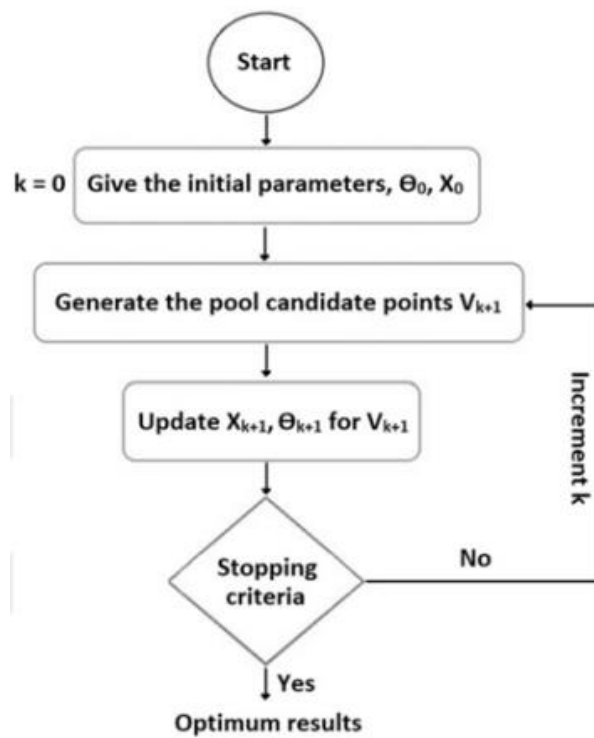


Figure 3.4: Flowchart of the random search algorithm [49]

3.2. Mathematica and Optimization

3.2.1. Global and Local Optimization by Mathematica

There are several commands in the Mathematica software that solve linear-nonlinear and unconstrained-constrained problems by exact-numeric optimization. In this regard, while `Minimize` and `Maximize` are only acceptable for precise global optimization, `NMinimize` and `NMaximize` are used in numerical global optimization methods. The `FindMinimum` command is used to do numerical local optimization. The commands mentioned above can be used to solve constrained-unconstrained, linear and nonlinear optimization problems [50]. Table 3.1 and Figure 3.5 provide comprehensive explanations of the instructions, techniques, and categories of problems that are used to solve them.

For restricted nonlinear problems, there are two types of numerical global optimization strategies: gradient-based and direct search. While Direct Search methods have a probabilistic process and do not need derivative information, Gradient-Based methods use the objective function's first or second derivatives as well as constraints to calculate results.

Linear programming (LP) problems are those in which the objective function and the constraints are linear functions of the optimization variables. Nonlinear programming (NLP) is used for solving optimization problems when some constraints or objective functions are nonlinear. For an objective function over unknown real variables, maximum, minimum, or stationary points are computed in the presence of congruent and noncongruent restrictions, collectively referred to as constraints [50].

If a global optimization is required, all four methods—`Minimize`, `Maximize`, `NMinimize`, and `NMaximize`—are applicable. `Minimize` and `Maximize` could find a correct global optimum for the class of optimization problems that include random multinomial issues. Using `NMinimize` or precisely with `Minimize`, questions involving global optimization can be solved computationally. Additionally, the method used can only be used in scenarios with a limited number of variables due to their asymptotic complexity. `FindMinimum` is the

appropriate tool if the problem requires a local optimum or if it can be solved by using just one or a few distinct points of departure. It simply tries to identify a local minimum. For minor problems and local optimal solutions, NMinimize could be useful. Only one of the four direct search methods—Nelder-Mead, differential evolution, simulated annealing, and random search—is used by NMinimize. To perfect the solving, a combination of the KKT solution, inner point, and penalty technique is used. If efficiency is not a factor, NMinimize must be more powerful than FindMinimum in addition to being a global optimal solver. FindMinimum, on the other hand, may be used when the efficiency is important if a local minimum is required, an excellent point of departure is required, the scenario has just one lower point (for example, convex), or the situation is big-budget. FindMinimum and NMinimize are used to solve a similar challenge with seven parameters. It costs a lot of time and money to calculate restrictions [50].

The capabilities of the algorithms are assessed in this chapter for finding the global minimum for various test functions using the Mathematica commands FindMinimum, NMaximize, and Nminimize, RandomSearch, SimulatedAnnealing, NelderMead, and DifferentialEvolution.

Table 3.1: Methods and instructions for optimization [50]

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

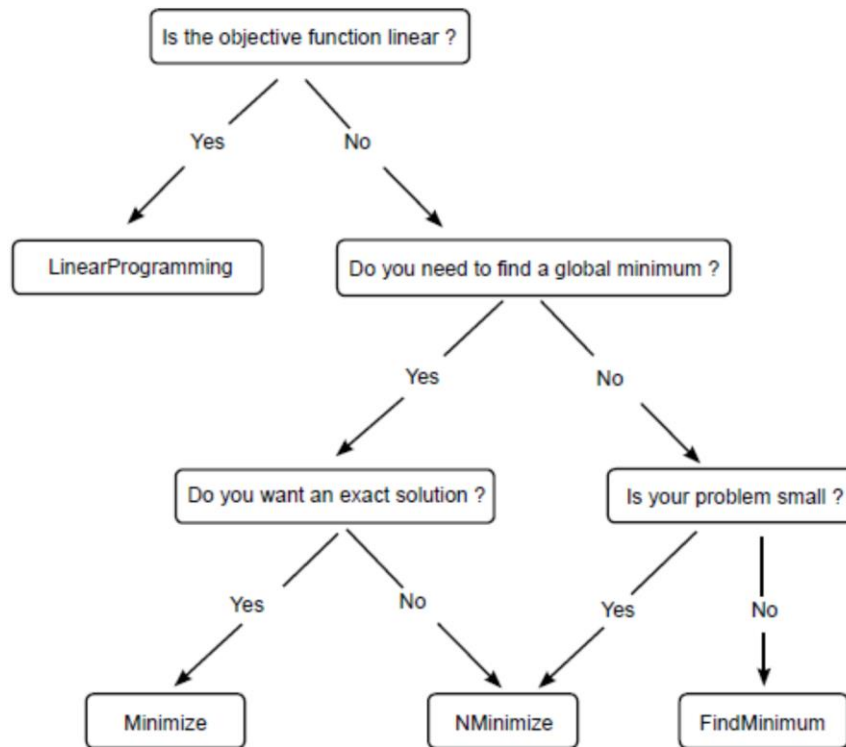


Figure 3.5: Mathematica optimization process [50]

3.2.2. FindMinimum

The FindMinimum command is utilized to determine the global lower limit function for unconstrained and constrained optimization problems [50].

MaxIterations, Method, PrecisionGoal, WorkingPrecision, and AccuracyGoal are the options for the FindMinimum command.

The FindMinimum command chooses which method to use to solve problems, according to the Method option. Here, we consider unconstrained optimization problems: (i) Newton uses the exact Hessian or a finite difference approximation; (ii) The quasi-Newton BFGS approximation, which was created by previous steps, is used in Quasi-Newton, (iii) non-linear least-squares problems are solved using the LevenbergMarquardt method, also known as the damped least-squares (DLS) method, (iv) and linear systems are solved using the ConjugateGradient method, (v) the PrincipalAxis method requires two starting conditions for each variable and does not require derivatives. Only InteriorPoint can be chosen as a method for constrained optimization problems.

The MaxIterations parameter determines how many iterations in total should be used. The standard "MaxIterations->500" is used in restricted optimization problems.

The options WorkingPrecision, PrecisionGoal, and AccuracyGoal determine the number of digits of precision. While the latter verifies the outcome, the former controls the internal calculations. WorkingPrecision->prec defaults to being equal to MachinePrecision, but if prec is greater than MachinePrecision, a constant prec value is used instead. The default settings are WorkingPrecision/3 and Infinity, respectively, when AccuracyGoal and PrecisionGoal cases are specified as Automatic [50].

The Carrom table function, a non-separable, multimodal function with numerous local minima, has been used as a test function, and the effectiveness of the FindMinimum command and its options in locating local minima is examined [51].

```
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}]
```

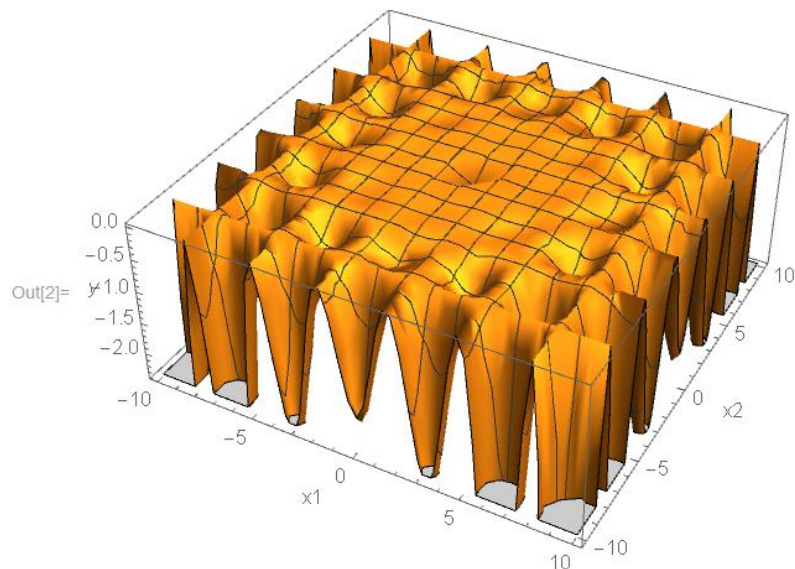


Figure 3.6: 3D plot of the function $f(x_1,x_2)$


```

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}}

```

3.2.3.NMinimize and NMaximize Functions

These Mathematica functions enable us to find the optimal solutions to complex problems in science and engineering and their unique characteristics by using search methods. They are efficient at finding global optimum solutions, but even in the absence of boundary conditions and restrictions, it may be difficult to achieve optimal results. Optimizing the given functions with various initial conditions might be the best solution to this problem. The initial test functions are used to get the following examples, which are the Ackley function of $f(x_1, x_2)$ and the 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}}

```

Initial results suggest that the Ackley Function's global minima and maximal values may be attained. It is obvious it was invalid for the Holder Table 1 function, nevertheless. It may be possible to obtain global values by changing the restriction region or the parameter values.

Limitations may take the form of lists or logical combinations of options, as well as equality and inequity within the domain. For example, if it is important to define results as integers, add the $z \in \text{Integers}$ to the line. This limitation makes integers the only possible solutions. The `NMinimize` command additionally needs a quadrilateral starting area to begin the optimum solution. Every parameter in the given function needs to have an upper and lower boundary. As observed in earlier chapters of this book utilizing the SA and RS algorithms, using the Method selection allows us to construct various types of search methods and obtains unautomated set outcomes. In this situation, it can be said that the `LinearProgramming` method is the default option in the solving process if the function being minimized or maximized (referred to as an objective function) and constraints are linear. DE is the algorithm by default if the variables are integer form and the central portion of the objective function are not numerical. *NM* should be utilized as the search algorithm in other situations. `NelderMead` switches with DE to acquire optimal values if it does not offer attractive solutions [50].

3.3.4. Random Search Solver

A stochastic technique is used in Mathematica's implementation of the Random Search (RS) algorithm. The algorithm creates a population during operation,

including random starting points, and then uses the FindMinimum local search method to assess the convergence behavior of the starting points to the local lowest limit. 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.

The Random Search algorithm's options InitialPoints, Method, PenaltyFunction, PostProcess, and SearchPoints are automatically controlled by Mathematica, and appropriate values for these options are chosen in accordance with optimization problems [50]. The RS algorithm operates according to the procedures shown in Figure 3.7.

Separable and non-separable multimodal test functions with more than one, few, or many local minima are used to test the Random Search algorithm's effectiveness in locating the global minimum. When an algorithm is not properly designed, it can be inserted into the local minima without finding the global minimums or not all global minimums, making this type of global optimization problems quite difficult. In this regard, Ackley is the first chosen test function with a global minima at $f(0, 0) = 0$ [51]. The definition of the Ackley function and its 3D plot in an interval as seen Figure 3.7. are given in the Mathematica syntax in the following commands.

```
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}]
```

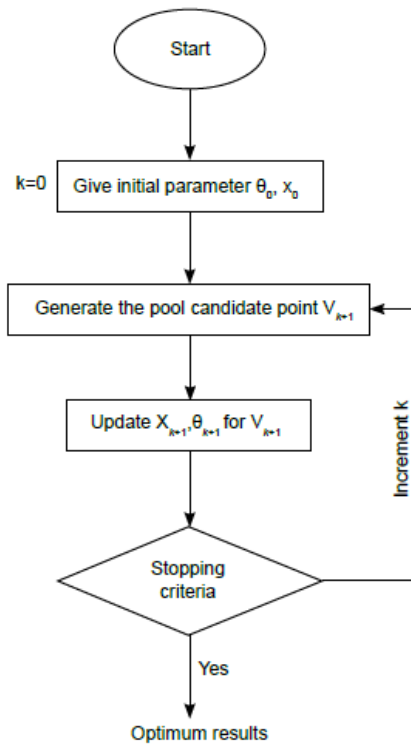


Figure 3.7: Flowchart of the random search algorithm [51]

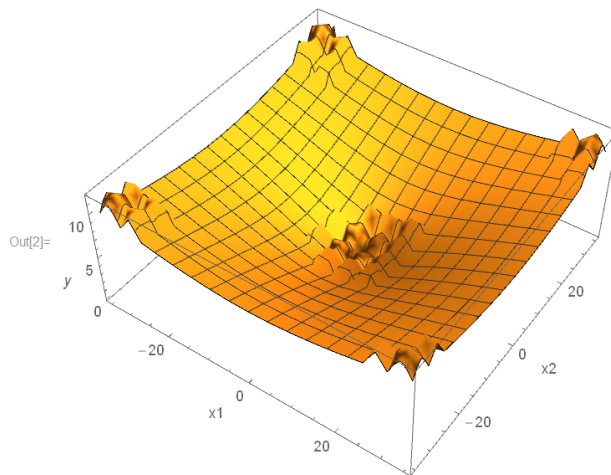


Figure 3.8: 3D plot of Ackley function in an interval

It should be noted that if the RandomSearch command's arguments are left unchanged, it might not be able to find a global minimum.

```

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 establishes the random number generator's starting value, can be investigated in the sections that follow. In the previous situation, the value of "Searchpoints"->500 is insufficient to reach the global minimum; however, in the following example, by setting the values of the SearchPoints and the RandomSeed to 500 and 5, respectively, a global minimum can be attained.

```
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}}
```

In this case, a matrix's points are created and used as starting points. It is simpler for getting at the answer if a beginning point is assigned and the problem's approximate solution range may be estimated.

```

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. This test function has global minima located at $f(\pm 9.646168, \pm 9.646168) = -26.920336$. The definition of the "Holder Table 1" function and associated 3D display are given in the Mathematica syntax shown below.

```

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}]

```

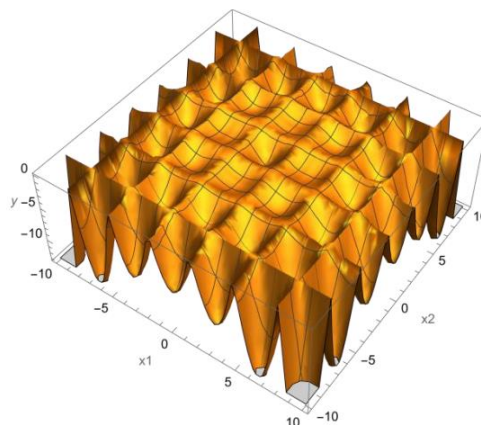


Figure 3.9: "Holder Table 1" function's 3D plot.

Without changing any of its options for this problem, the RS algorithm finds one of the global minima.

```
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}}
```

3.2.5. Simulated Annealing Solver

The Simulated Annealing (SA) algorithm, which Mathematica has implemented, is a stochastic approach that bases its operation on the solids' physical annealing process. The SA's purpose is to determine the largest or smallest values of functions with numerous variables as well as the smallest values of nonlinear functions with numerous local minimums. Simulated annealing is the name of the algorithm because it represents the ideal arrangement of atoms in solid bodies and the minimizing of potential energy during cooling. The algorithm gives the system the ability to stray from the local minimum, investigate, and find a better global minimum [52].

For each iteration, the startup solution "Z" is first produced, followed by the generation of "Z_{new}" close to the current point, "Z" and finally the definition of "Z_{best}".

If $f(Z_{new}) \leq f(Z_{best})$, Z_{new} replaces Z_{best} and Z. Otherwise, Z_{new} replaces with Z. In this loop, the initial guess, as well as its number and starting value, may be found using the variables InitialPoints, SearchPoints, and RandomSeed. Based on the Boltzmann probability distribution $(k, \Delta f, f_0)$, random movements in the search space are carried out via the SA algorithm. In the equation, D stands for the function that the Boltzmann Exponent explanation for, k for the current iteration, and Δf for the variance of the objective function. In the Mathematica, if the user does not select manually, B is defined as $\frac{-\Delta f \log(k+1)}{10}$ by BoltzmannExponent.

The working process described above is returned for all starting points once the algorithm either converges to a spot or stays at the same position due to the number of iterations specified by the `LevelIterations` parameter [53]. Following the steps shown in Figure 3.10, the SA algorithm operates.

The performance capability of the `SimulatedAnnealing` command to determine the global minimum is assessed using "Ackley" and "Holder Table 1".

```
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}]
```

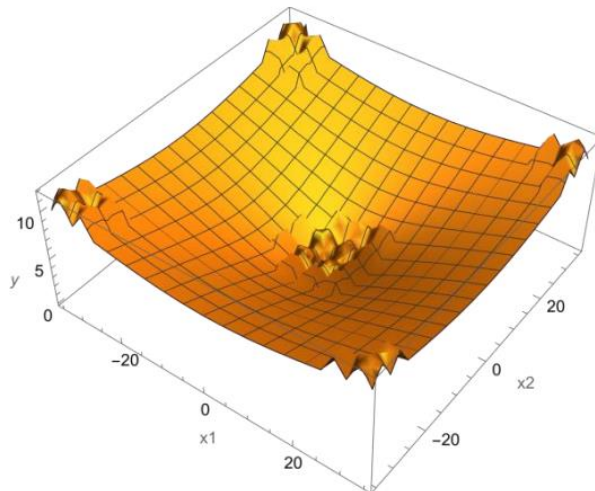
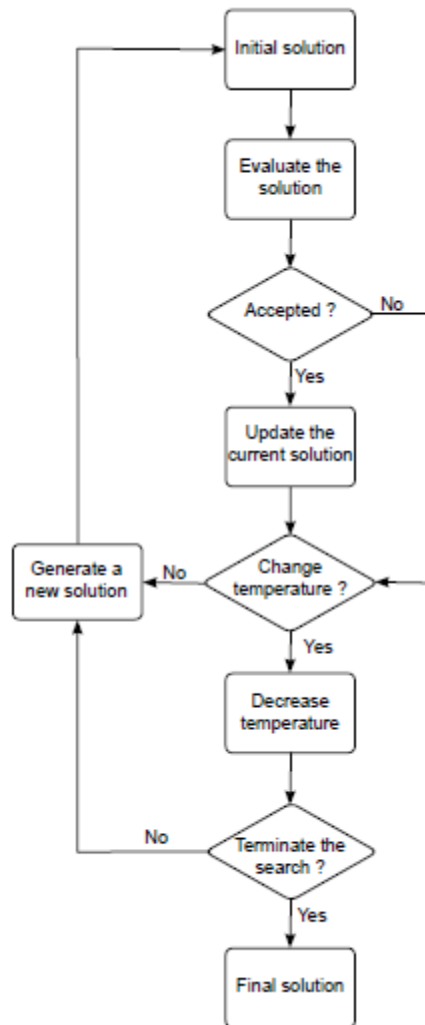


Figure 3.10: Flowchart of the simulated annealing algorithm [54]



By using the default value of its options, the SA algorithm might fail to locate a global minimum.

```

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}}
  
```

The BoltzmannExponent is an important tool that shows how to get to a global minimum because it includes a function that determines a new point with each iteration. If this function is used without specifying a default value, the result can be altered. The global lower limit has not been able to be determined in the following problem, despite changing this option alone.

```

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}}
  
```

Although the `PerturbationScale` affects the result for this problem, changing this option by itself has not been sufficient to determine the global minimum. Local minimum points are attained by the algorithm.

```
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 determine the global lower limit., in the case of conducting a search utilizing 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}}

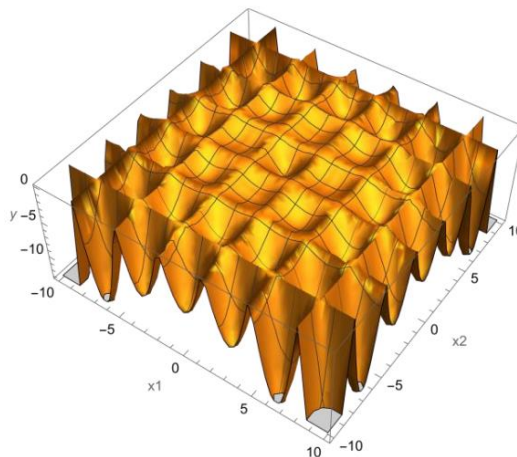
```

```
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}]
```



Without changing any of its options for this problem, the Simulated Annealing algorithm finds one of the global minimum points.

```
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}}
```

The Simulated Annealing algorithm, comparing to the Random Search algorithm, finds four distinct global minimum points.

```
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}}
```

3.2.6. Nelder Mead Solver

Nelder-Mead (NM) or Simplex is one of the derivative-free optimization methods among other traditional local search algorithms. It was first developed for challenges involving unconstrained optimization [55]. This method maintains a set of $m+ 1$ points that create the vertices of a polytope in m -dimensional space given a function of m variables. It should be highlighted that the simplex method for linear programming should not be confused with this. Iterations have been performed by forming $m+ 1$ points as $y_1, y_2, y_3, \dots, y_{m+1}$. These points form the functions are ordered as $h(y_1) \leq h(y_2) \leq h(y_3) \leq \dots h(y_{m+1})$. Once the new point is created, it will replace the old worst point, y_{m+1} . The centroid of a polytope serves as a definition $c = \sum_{i=1}^m y_i$, 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, $y_t = c+a(c - y_{m+1})$ where a is a variable being larger than 0. The new point in this section does not necessarily have to be a new worst or best point. Hence, $h(y_1) \leq h(y_t) \leq h(y_m)$, y_t replace with y_{m+1} . When a new point is obtained and is higher than the previous highest point, reflection has been

successfully attained.

It can also be continued with $y_e = c + (\beta y_t - r)$, where β is a parameter to enlarge the polytope and is greater than 1. The expansion process is complete if $h(y_e)$ is found to be less than $h(y_t)$. As a result, y_e changes with y_{m+1} . Otherwise, y_t changes to y_{m+1} in the alternative. Another certain step for the algorithm process is that if the fresh point y_t underperforms to the second-lowest point, $h(y_m) \leq h(y_{m+1})$, the polytope is thought as very large and it is required to be constricted [56].

Hence, a fresh test point is obtained using the following expressions [56].

$$y_c = \begin{cases} c + \gamma(y_{m+1} - c), & \text{if } h(y_t) \geq h(y_{m+1}) \\ c + \gamma(y_t - c), & \text{if } h(y_t) < h(y_{m+1}) \end{cases} \quad (3.2)$$

where γ is a parameter with values ranging from 0 to 1. If contraction is attained, $h(y_c)$ is smaller than $\text{Min}[h(y_{m+1}), h(y_t)]$. On the other hand, stronger contraction needs more work.

Similar to other algorithms, Nelder-Mead has specific flexible options like ContractRatio, ExpandRatio, InitialPoints, PenaltyFunction, PostProcess, RandomSeed, ReflectRatio, ShrinkRatio, and Tolerance. This algorithm works effectively for problems with less local minima, while not having the complete specifications that a reliable global optimization method should require. As with previous algorithms, Nelder-Mead is utilized to obtain the optimum global values for the Ackley and Holder Table 1 test functions [50].

```
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}}
```

As can be observed, DE outperforms the first trial's results while providing better global optima for the Ackley function with the default set than Random Search and Simulated Annealing.

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}}

```

A better minimum value of 2.32486×10^{-10} was obtained by adjusting the RandomSeed parameters as opposed to a trial performed the default settings.

Other possible useful adjustment factors in this algorithm are known as ShrinkRatio, ContractRatio, and ReflectRatio. The following, however, shows that it did not achieve a global minima in the Ackley function.

```

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}}

```

Using the NMinimize command, Holder Table 1—another test function—was minimized. The global minima with default values was, as can be seen below, -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}}

```

First, RandomSeed has been adjusted to identify global minima, as it was for the preceding test function.

```

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}}

```

The results of this testing demonstrated that RandomSeed's adjustment like that was insufficient to achieve the minimum value. Finally, additional potentially helpful parameters related to the literature for NelderMead were changed to get 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}}

```

This example showed that none of the parameters could guarantee that a global minimum is different from the result obtained with the using default settings.

Chapter 4

4. Optimization of Drying Process of Plastic Granules

4.1. Methods

4.1.1. NDSolve Solver

The `NDSolve` command in the Mathematica software can be used to numerically solve ODEs and PDEs. Instead of having to write a function, it provides `InterpolatingFunction`, a suitable interpolation function. The Dirichlet Condition (a type of boundary condition for a partial differential equation that gives the prescribed value of the function on a surface) and Neumann Condition (a type of boundary condition for a partial differential equation that gives the first derivative on a surface) can also be used to specify boundary values. 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. For the `NDSolve` solution, the iteration procedure is appropriate. A specific prescribed value is taken into account in the iteration's initial step. Second, the starting point for the next iteration is the output of the previous one. Finally, a series of outputs up to the endpoint are produced by this repeated process. To define the maximum number of steps of the iteration process, we can use the “MaxSteps” option by selecting `Automatic` mode. Additionally, the terms `StartingStepSize`, `MaxStepSize`, and `NormFunction` are used to describe, respectively, the size of the step at the beginning, the maximum size of step in the independent variable of the equation, and the norm of error estimation. Mathematica uses 10,000 as a stopping criterion if the process's maximum number of iterations is not provided. Given that error estimations have an impact on tolerances. By satisfying the following requirement, they can be scaled by combining the mistakes for different terms.

$$f\left[\left[\frac{error_1}{tolerance_r |x_1| + tolerance_a}, \frac{error_2}{tolerance_r |x_2| + tolerance_a}, \dots, \frac{error_n}{tolerance_r |x_n| + tolerance_a}\right]\right] \leq 1 \quad (4.1)$$

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. $Tolerance_a$ and $tolerance_r$, respectively, stand for absolute and relative tolerances. An improved version of the explicit Runge-Kutta, which is also an adaptive embedded pair of orders, is used in the approach to try to get an appropriate step size using an embedded error estimator.

The user has control over the NDSolve command's TimeIntegration, BoundaryValues, and EquationSimplification options. These options correspond to DE systems, ODE boundary value problems (BVPs), and simplified equations, depending on the type of DE. By adjusting the time integration parameters, the explicit Runge-Kutta method is also hybridized with the Adams, BDF, ExplicitRungeKutta, ImplicitRungeKutta, and Symplectic-PartitionedRungeKutta approaches. The method starts with a trial step at the midpoint for the domain, and this leads to reducing lower-order error terms [57].

4.1.2. FindFit Solver

This solver is used to numerically derive the best-fit function to the prescribed data. The method options include "ConjugateGradient", "Gradient", "LevenbergMarquardt", "Newton", "NMinimize", and "QuasiNewton" and can be chosen based on what is given problem. The Levenberg-Marquardt method is chosen as an appropriate process to compute the regression coefficients in the given problem, which is also a sub-problem of the least-square approximation. It is a method for minimizing a sum-of-squares objective function. Equation 5.2, which relates a version of Gauss-Newton and Gradient Descent updates for the specified parameters, is valid for this method.

$$[J^T W J + \lambda I] h = J^T (y - \hat{y}) \quad (4.2)$$

The Jacobian, traditional Jacobian, and diagonal weighting matrix, respectively,

are represented in this equation by J^T , J , W . λ stands for the damping parameter, which can be changed to be either large or small. \hat{y} stands for the fitted function, I stands for the identity matrix, h is the perturbation, and y is a set of measured points [58].

4.2. Engineering Model

Components of a typical polymer drying system are (1) main silo, (2) process blower, (3) process filter, (4) process heater, (5) desiccant wheel, (6) regeneration heater, (7) regeneration blower and (8) regeneration filter, as shown in Figure 4.1.

In addition to this, parameters of a typical drying system of polymers are temperature, drying time. These two parameters can be changed manually by the user. In addition, environmental factors also affect drying system (ambient temperature, ambient humidity, dew point of air).

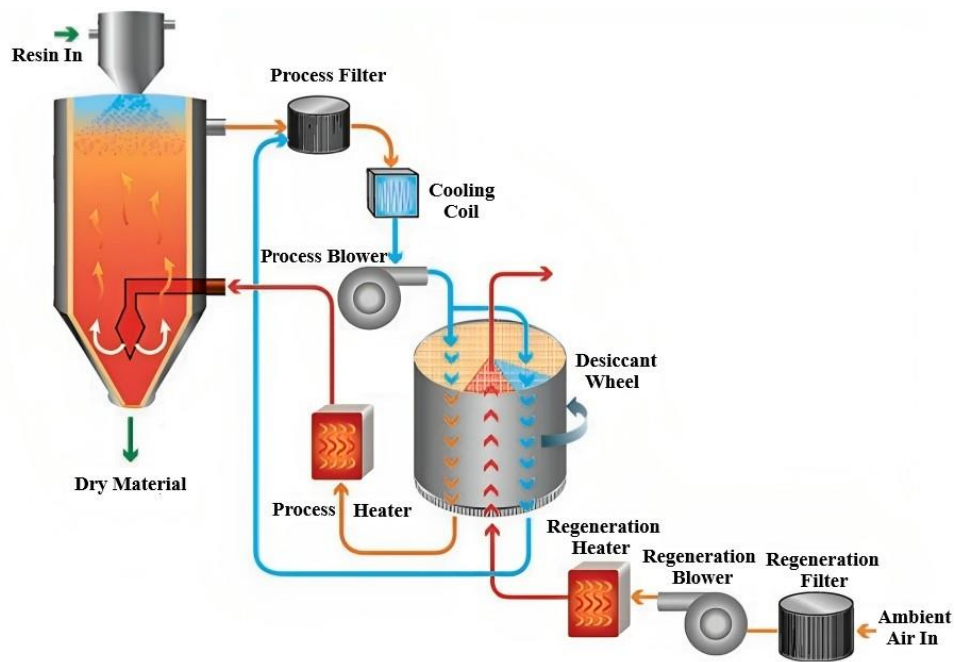


Figure 4.1: Illustrating of the polymer drying system

4.2.1. Mathematical Model

Table 4.1: Design Variables

Objective Function	Design Variables	Constraints
Energy (kW/hr) E (MIN)	Drying Temperature (°C) C Drying Time (Minute) t	Moisture (M) < (%0.02)

This study was prepared using the experimental data given in Table 4.2-4.3-4.4. Experimental data were taken with the energy analyzer integrated into the drying machine, as shown in Figure 4.2., the sensor measuring environmental temperature, pressure and moisture, as shown in Figure 4.3., the sensor mounted on the dryer machine shows the dryness level of the hot air entering the dryer machine, as shown in Figure 4.4.. and the moisture content of the sample granules taken from the dryer machine was measured with an instant moisture meter, as shown in Figure 4.5.

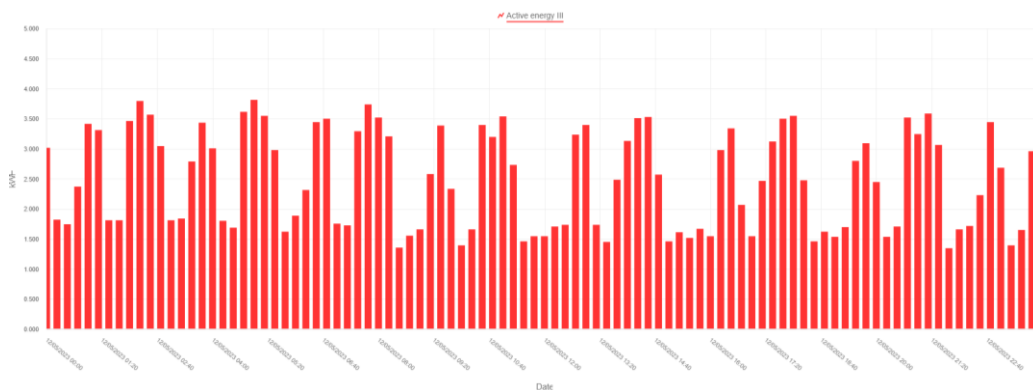


Figure 4.2: Energy Analyzer Interface



Figure 4.3: Environmental Values Measurement Sensor



Figure 4.4: Shini Dew Point Sensor



Figure 4.5: Sartorius MA45 Moisture Analyzer

Sample granules were taken from the dryer machine at certain periods and their humidity was recorded. These trials were conducted in accordance with the table obtained from the DoE study, the table is given in Appendix A. According to literature research, the drying process of previous studies is not sufficient in terms of sensitivity and reliability. The drying process parameters have been modeled using a different approach to produce more precise results by introducing a new approach to the drying process. In this approach, the experimental data were used and modeled and optimized with neuro regression analysis. The existing data was divided into three groups (training, testing, validation) and many mathematical models (linear, logarithmic, rational, hybrid, etc.) were created. As a second step, the engineering limits of the candidate models were checked to produce realistic values. Finally, the results were evaluated using different stochastic optimization algorithms (Differential Evaluation, Nelder-Mead, Random Search and Simulated Annealing algorithms).

Table 4.2: Drying experiment parameters for 65°C

Time (min)	Mositure (%)		Average Moisture (%)	Dew Point (°C)	Environmental Pressure (mmHg)	Environmental Temperature (°C)	Environmental Humidity (%)	Energy (kWh)	Total Energy (kWh)
0	0,16	0,14	0,15	-10	758	26	32	0	0
15	0,14	0,14	0,14	-10	758	26	32	1,517	1,517
30	0,11	0,08	0,095	-10	758	27	32	2,929	4,446
45	0,13	0,13	0,13	-10	757	27	32	2,712	7,158
60	0,08	0,11	0,095	E2	758	27	33	1,662	8,820
75	0,05	0,04	0,045	-10	756	26	32	1,187	10,007
90	0,02	0,02	0,02	-10	758	26	33	1,878	11,885
105	0,01	0,02	0,017	-10	757	26	33	3,151	15,036
120	0,01	0,02	0,015	-10	758	25	34	2,849	17,885
135	0,02	0,03	0,025	-10	758	25	35	3,121	21,006
150	0,04	0,03	0,037	-10	758	25	36	3,12	24,126
165	0,02	0,04	0,027	-10	757	24	36	1,757	25,883
180	0,02	0,01	0,013	-15	757	24	36	1,537	27,420
195	0,02	0,01	0,015	-10	758	24	37	1,65	29,070
210	0,02	0,01	0,015	-10	757	24	37	1,518	30,588
225	0	0	0	-10	758	23	38	1,421	32,009
240	0,01	0,02	0,015	-10	758	23	38	2,311	34,320
255	0,02	0,02	0,02	-10	758	23	38	3,035	37,355
270	0,01	0,03	0,02	-10	758	22	40	2,838	40,193
285	0,01	0,01	0,01	-10	758	21	41	1,416	41,609
300	0,01	0,01	0,01	-10	758	21	42	1,351	42,960

Table 4.3: Drying experiment parameters for 75°C

Time (min)	Mositure (%)			Average Moisture (%)	Dew Point (°C)	Environmental Pressure (mmHg)	Environmental Temperature (°C)	Environmental Humidity (%)	Energy (kWh)	Total Energy (kWh)
	0,1	0,13	0,115							
0	0,1	0,13	0,115		-10	762	25	18	0	0
15	0,1	0,13	0,115		-10	761	24	18	1,613	1,613
30	0,09	0,09	0,09		-10	761	24	18	2,916	4,529
45	0,03	0,06	0,047	0,05	-10	761	23	19	2,635	7,164
60	0,02	0,04	0,03	0,03	-10	762	23	21	1,248	8,412
75	0,03	0,03	0,03		-10	761	21	22	1,445	9,857
90	0,02	0,03	0,025		-10	760	21	21	2,389	12,246
105	0,02	0,05	0,04	0,05	-10	762	22	21	3,138	15,384
120	0,02	0,01	0,017	0,02	-10	761	22	22	3,24	18,624
135	0,01	0,04	0,027	0,03	-10	762	23	22	3,222	21,846
150	0,02	0,03	0,025		-10	762	22	25	2,358	24,204
165	0,02	0,03	0,025		-10	762	22	25	1,38	25,584
180	0,01	0,02	0,013	0,01	-24	762	22	26	1,458	27,042
195	0,01	0,01	0,01		-17	762	23	27	1,397	28,439
210	0,01	0,02	0,013	0,01	-11	762	22	28	3,336	31,775
225	0,01	0,01	0,01	0,01	-10	762	21	29	2,832	34,607
240	0,01	0,01	0,01		-10	762	21	30	1,555	36,162
255	0,01	0,01	0,01		-12	762	21	30	1,752	37,914
270	0,01	0,01	0,01		-11	762	21	30	2,615	40,529
285	0,01	0,01	0,01		-10	762	21	31	3,288	43,817
300	0,01	0,01	0,01		-10	762	20	31	3,601	47,418

Table 4.4: Drying experiment parameters for 85°C

Time (min)	Mositure (%)		Average Moisture (%)	Dew Point (°C)	Environmental Pressure (mmHg)	Environmental Temperature (°C)	Environmental Humidity (%)	Energy (kWh)	Total Energy (kWh)
0	0,13	0,17	0,15	-10	758	20	42	0	
15	0,1	0,15	0,125	-10	757	19	44	1,835	1,835
30	0,1	0,11	0,105	-10	758	20	45	3,342	5,177
45	0,08	0,05	0,04	-10	758	19	46	3,685	8,862
60	0,03	0,03	0,03	-10	757	19	46	3,689	12,551
75	0,01	0,02	0,015	-10	757	19	46	3,378	15,929
90	0,02	0,01	0,015	-10	758	19	46	2,34	18,269
105	0,01	0,02	0,017	-21	758	19	46	1,837	20,106
120	0,01	0,01	0,01	-15	757	19	45	1,733	21,839
135	0,01	0,01	0,01	-11	758	20	46	1,914	23,753
150	0,01	0,01	0,010	-10	758	19	46	2,821	26,574
165	0,01	0,01	0,010	-10	758	19	46	3,353	29,927
180	0,01	0,01	0,010	-10	758	19	46	2,863	32,790
195	0,01	0,01	0,01	-10	758	19	46	1,818	34,608
210	0,01	0,02	0,015	-10	758	19	46	1,739	36,347
225	0,01	0,01	0,01	-10	758	18	47	3,769	40,116
240	0,01	0,01	0,01	-10	758	18	47	3,426	43,542
255	0,02	0,01	0,015	-10	758	18	48	3,725	47,267
270	0,01	0,01	0,01	-10	758	18	48	3,007	50,274
285	0,01	0,02	0,015	-10	758	19	48	1,811	52,085
300	0,01	0,01	0,01	-19	758	18	48	1,741	53,826

4.3. Materials and Methods

Table 4.5: Multiple regression model names, nomenclatures and formulas

Model Name	Nomenclature	Formula
Multiple linear	L	$Y = \sum_{i=1}^t (a_i x_i) + c$
Multiple linear rational	LR	$Y = \frac{\sum_{i=1}^t (a_i x_i) + c_1}{\sum_{j=1}^t (\beta_j x_j)} + c_2$
Second order multiple nonlinear	SON	$Y = \sum_{k=1}^t \sum_{j=1}^t (a_j x_j x_k) + \sum_{i=1}^t (a_i x_i) + c$
Second order multiple nonlinear rational	SONR	$Y = \frac{\sum_{k=1}^t \sum_{j=1}^t (a_j x_j x_k) + \sum_{i=1}^t (a_i x_i) + c_1}{\sum_{m=1}^t (\beta_m x_m x_i) + \sum_{n=1}^t (\beta_n x_n)} + c_2$
Third order multiple nonlinear	TON	$Y = \sum_{l=1}^t \sum_{m=1}^t \sum_{p=1}^t (\beta_l x_l x_m x_p) + \sum_{k=1}^t \sum_{j=1}^t (a_j x_j x_k) + \sum_{i=1}^t (a_i x_i) + c$
First order trigonometric multiple nonlinear	FOTN	$Y = \sum_{i=1}^t (a_i \sin[x_i] + a_i \cos[x_i]) + c$
First order trigonometric multiple nonlinear rational	FOTNR	$Y = \frac{\sum_{i=1}^t (a_i \sin[x_i] + a_i \cos[x_i]) + c_1}{\sum_{j=1}^t (\beta_j \sin[x_j] + \gamma_j \cos[x_j])} + c_2$
Second order trigonometric multiple nonlinear	SOTN	$Y = \sum_{i=1}^t (a_i \sin[x_i] + a_i \cos[x_i]) + \sum_{j=1}^t (\beta_j \sin^2[x_j] + \gamma_j \cos^2[x_j]) + c$
Second order trigonometric multiple nonlinear rational	SOTNR	$Y = \frac{\sum_{i=1}^t (a_i \sin[x_i] + a_i \cos[x_i]) + \sum_{j=1}^t (\beta_j \sin^2[x_j] + \gamma_j \cos^2[x_j]) + c_1}{\sum_{k=1}^t (\theta_k \sin[x_k] + \theta_k \cos[x_k]) + \sum_{l=1}^t (\delta_l \sin^2[x_l] + \delta_l \cos^2[x_l])} + c_2$
First order logarithmic multiple nonlinear	FOLN	$Y = \sum_{i=1}^t (a_i \log[x_i]) + c$
First order logarithmic multiple nonlinear rational	FOLNR	$Y = \frac{\sum_{i=1}^t (a_i \log[x_i]) + c_1}{\sum_{j=1}^t (\beta_j \log[x_j])} + c_2$
Second order logarithmic multiple nonlinear	SOLN	$Y = \sum_{k=1}^t \sum_{j=1}^t (a_j \log[x_j x_k]) + \sum_{i=1}^t (a_i \log[x_i]) + c$
Second order logarithmic multiple nonlinear rational	SOLNR	$Y = \frac{\sum_{k=1}^t \sum_{j=1}^t (a_j \log[x_j x_k]) + \sum_{i=1}^t (a_i \log[x_i]) + c_1}{\sum_{m=1}^t \sum_{l=1}^t (a_l \log[x_l x_m]) + \sum_{n=1}^t (a_n \log[x_n])} + c_2$

During the modeling phase, neuro-regression analysis was used to test the accuracy of the predictions. In this approach, all experimental data are randomly divided into three. 80% of the experimental data is randomly allocated as training, 15% as testing and 5% as validation. These separated experimental data were defined as input to the Wolfram Mathematica 12.0 program. The aim is to create a mathematical model; is to obtain the best R^2 values from these models. As a first step, R^2_{training} and $R^2_{\text{trainingadjusted}}$ values are calculated from the mathematical model. These two values are expected to be greater than 95% and close to each other. The next step is to check the R^2_{testing} value to make sure that the mathematical model gives more accurate results. The R^2_{testing} value is expected to be greater than 85%. If the R^2_{training} and R^2_{testing} values are at the desired value, the $R^2_{\text{validation}}$ value is calculated as the next step. The $R^2_{\text{validation}}$ value must also be greater than 85%. If R^2_{training} , R^2_{testing} and $R^2_{\text{validation}}$ values are not within the desired ranges, go back to the beginning and change the mathematical model and have these values calculated again in the new mathematical model. This cycle continues until the desired R^2 values are reached. Table 4.5 shows all models written for neuro-regression analysis of the problem.

4.4. Optimization Problem Definition

With the methods described, the optimum design of a drying process parameters was made in the following steps.

- The parameters given in Table 4.2-4.4 were recorded as a result of the experiments.
- Different functions were written mathematically with the experimental data, and the best of these functions was determined by looking at R^2_{training} , R^2_{testing} and $R^2_{\text{validation}}$ values.
- Four different optimization scenarios were created by selecting the functional structure that gave the best R^2 value. All of these scenarios were solved with four different stochastic optimization algorithms.

4.5. Optimization Scenarios

Scenario 1: The objective function in this optimization problem is to minimize the total energy used. In Scenario 1, it is assumed that all parameters can be real numbers within certain ranges. Real experiment constraints are ignored. Under these conditions; $65 < \text{drying temperature} < 85$, $0 < \text{drying time} < 300$, restrictions were imposed and subjected to optimization algorithms. While determining the lower and upper limits, the lower and upper limits given by the plastic granule manufacturers were used as a reference and the decision was made taking into account the production experience.

Scenario 2: The objective function in this optimization problem is to minimize the total energy used. In Scenario 2, it is assumed that all parameters can be real numbers within certain ranges. Real experiment constraints are ignored. Under these conditions; $65 < \text{drying temperature} < 85$, $90 < \text{drying time} < 300$, restrictions were imposed and subjected to optimization algorithms.

Scenario 3: The objective function in this optimization problem is to minimize the total energy used. In Scenario 3, all parameters are assumed to be the same range as in Scenario 2, but these parameters can be only integers. Actual experimental constraints are ignored. Under these conditions; $65 < \text{drying temperature} < 85$, $90 < \text{drying time} < 300$ were subjected to optimization algorithms by imposing restrictions.

Scenario 4: By adhering to the real experiment parameters, real constraints have been imposed on the optimization problem. In order to minimize the total energy used, all parameters are real numbers in the previous scenario ranges, drying temperature $\in \{65, 70, 75, 80, 85\}$; drying time $\in \{90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260, 270, 280, 290, 300\}$.

Table 4.6: Objective Function

Problem No	Objectives	Constraints
Problem 1	Minimize [E]	$65 < C < 85$ $0 < t < 300$
Problem 2	Minimize [E]	$65 < C < 85$ $90 < t < 300$
Problem 3	Minimize [E]	$65 < C < 85, C \in \text{integers}$ $90 < t < 300, t \in \text{integers}$
Problem 4	Minimize [E]	$x1 == 65 \parallel x1 == 70 \parallel x1 == 75 \parallel x1 == 80 \parallel x1 == 85 \parallel$ $\parallel x2 == 90 \parallel x2 == 100 \parallel x2 == 110 \parallel x2 == 120 \parallel x2 == 130 \parallel x2 == 140 \parallel x2 == 150 \parallel x2 == 160 \parallel$ $\parallel x2 == 170 \parallel x2 == 180 \parallel x2 == 190 \parallel x2 == 200 \parallel x2 == 210 \parallel x2 == 220 \parallel x2 == 230 \parallel x2 == 240 \parallel$ $x2 == 250 \parallel x2 == 260 \parallel x2 == 270 \parallel$ $x2 == 280 \parallel x2 == 290 \parallel x2 == 300$ $0.02 \leq \%M$

In this study, in order to reduce the energy consumption of the drying system of ABS polymers, four optimization problems have been defined. Our system inputs are the set temperature and the waiting time of the polymers in the dryer, outputs are the moisture content of the polymer by weight and the energy consumed by the machine. The set temperature C , waiting time of the polymer in the dryer t are considered design variables.

4.6. Result and Discussion

Regression analysis and artificial neural networks (ANN) are combined in a hybrid method to assess the accuracy of the predictions made during the modeling phase. This methodology uses three sets of data, each comprising 80%, 15%, and 5% of the total data. The first set is used for training, the second is used for testing, and the third is used for validation. Reduction of the error between experimental and predicted values is the aim of the training procedure, which involves adjusting the regression models and coefficients given in Table 4.5 and hybrid models. In this study, 40 different mathematical models with two inputs (see Table 4.1) were tested and the results are given in Table 4.7-4.8.

While preparing Tables 4.7-4.8, R^2_{training} , R^2_{testing} and $R^2_{\text{validation}}$ values were calculated for each mathematical model. These values give information about the

success of the model and the values are expected to be 1 and/or close to 1. The closer the result is to 1, the better the model is at predicting the structure. While searching for the most successful model, the model that met the condition of being closest to 1 in all three values was selected. Because many models have R^2_{training} and R^2_{testing} values of 1 and/or close to 1 (see Table 4.7-4.8). The most successful model that simultaneously met all requirements for two outputs was selected, taking into account engineering boundedness. The model for energy output is the (TOTN+4thON) model, which is a hybrid model created by combining the third order trigonometric multiple nonlinear and fourth order multiple nonlinear models. The (TOTN+4thON) model, which is a hybrid model created by combining the third order trigonometric multiple nonlinear and fourth order multiple nonlinear models, which is the same hybrid model, was selected for moisture output.

Table 4.7: Accuracy check results of obtained models for the Energy output in terms of R^2 values

ENERGY				
	R^2_{training}	$R^2_{\text{trainingAdjusted}}$	R^2_{testing}	$R^2_{\text{validation}}$
1-L	0.995752	0.995462	0.975999	0.995508
2-LR	0.998029	0.997822	0.986149	0.999638
3-SON	0.998828	0.998657	0.99363	0.997295
4-SONR	0.993033	0.998805	0.995306	0.997645
5-TON	0.999005	0.992736	0.995782	0.997129
6-TONR	0.999467	0.999219	0.998505	0.994741
7-4thON	0.999087	0.998659	0.99634	0.997784
8-4thONR	0.999721	0.999468	0.999082	0.99933
9-5thON	0.999285	0.998708	0.996426	0.993878
10-5thONR	0.999737	0.99921	0.999258	0.999395
11-FOTN	0.996754	0.996185	0.985155	0.997248
12-FOTNR	0.848582	0.80532	0.479997	0.509306

13-SOTN	0.791458	0.727737	-1.87599	-0.335678
14-SOTNR	0.80726	0.703838	0.02351	0.22364
15-TOTN	0.794832	0.642855	-2.1913	-0.308922
16-TOTNR	0.863468	0.655939	0.369712	-0.0585805
17-FOLN	0.746591	0.729312	-0.211098	-0.0685417
18-FOLNR	0.758402	0.73297	-0.162213	0.0701223
19-SOLN	0.774711	0.741742	-0.937814	-0.250169
20-SOLNR	0.781802	0.730461	0.0468881	0.00867686
21-SOTNR+LR	0.8406	0.72859	0.173744	0.60184
22-TOTNR+LR	0.835299	0.68557	0.285233	0.0272367
23-SOLNR+LR	0.998958	0.998604	0.99551	0.997167
24-SOTN+TON	0.999047	0.998341	0.994322	0.9965
25-SOTNR+TONR	0.999502	0.998636	0.998897	0.998753
26-SOTN+4thON	0.99913	0.998142	0.994608	0.99708
27-SOTNR+4thONR	0.999662	0.998363	0.999158	0.99898
28-SOTN+5thON	0.999314	0.997985	0.994269	0.995269
29-SOTNR+5thONR	0.999888	0.992948	0.999762	0.999791
30-TOTN+TON	0.999128	0.997723	0.993669	0.996196
31-TOTNR+TONR	0.999777	0.997993	0.999437	0.99931
32-TOTN+4thON	0.999245	0.99727	0.993216	0.997641
33-TOTNR+4thONR	0.998306	1.03557	0.987386	0.999269
34-FOLNR+LR	0.998958	0.998761	0.99555	0.997203
35-FOLN+SON	0.998842	0.998604	0.993286	0.997337
36-FOLNR+SONR	0.999111	0.998808	0.995835	0.997911
37-FOLN+TON	0.999016	0.998679	0.995684	0.997119
38-FOLNR+TONR	0.999469	0.999143	0.998545	0.998775
39-FOLN+4thON	0.999095	0.998582	0.996288	0.997952
40-FOLNR+4thONR	0.999699	0.999346	0.999099	0.999032

Table 4.8: Accuracy check results of obtained models for the Moisture output in terms of R^2 values

MOISTURE				
	R^2_{training}	$R^2_{\text{trainingAdjusted}}$	R^2_{testing}	$R^2_{\text{validation}}$
1-L	0.741412	0.723781	0.572324	0.618364
2-LR	0.926342	0.918589	0.822627	0.913459
3-SON	0.906796	0.893157	0.769043	0.908181
4-SONR	0.968268	0.960801	0.867995	0.982048
5-TON	0.943542	0.928283	0.876827	0.964149
6-TONR	0.986797	0.980657	0.931553	0.969345
7-4thON	0.958956	0.939716	0.851102	0.936777
8-4thONR	0.99289	0.986426	0.961679	0.999417
9-5thON	0.968081	0.942299	0.765717	0.974122
10-5thONR	0.99641	0.98923	0.990973	0.99711
11-FOTN	0.744215	0.699453	0.5949	0.590206
12-FOTNR	0.948657	0.933987	0.900042	0.957616
13-SOTN	0.532853	0.390114	-1.78107	-0.419047
14-SOTNR	0.531586	0.280242	0.402331	-0.0627518
15-TOTN	0.557595	0.229888	-2.12602	-0.271768
16-TOTNR	0.668732	0.165205	0.427724	0.0633301
17-FOLN	0.45176	0.41438	-0.182404	-0.209413
18-FOLNR	0.442835	0.384186	-0.027849	-0.0927132
19-SOLN	0.511164	0.439627	-1.10111	-0.355042
20-SOLNR	0.705799	0.636575	0.928394	0.0727682
21-SOTNR+LR	0.97124	0.95103	0.963919	0.981696
22-TOTNR+LR	0.751001	0.524639	-0.062998	0.264564
23-SOLNR+LR	0.955135	0.939862	0.917185	0.957741
24-SOTN+TON	0.948819	0.910906	0.810242	0.964762

25-SOTNR+TONR	0.991354	0.976318	0.970072	0.983274
26-SOTN+4thON	0.963453	0.921921	0.79501	0.949241
27-SOTNR+4thONR	0.994375	0.972741	0.97261	0.999554
28-SOTN+5thON	0.971704	0.916882	0.716449	0.979891
29-SOTNR+5thONR	0.89408	-5.67299	0.734969	0.781832
30-TOTN+TON	0.95304	0.877381	0.670803	0.950888
31-TOTNR+TONR	0.997836	0.879784	0.705095	0.985742
32-TOTN+4thON	0.966749	0.980527	0.991747	0.999221
33-TOTNR+4thONR	0.971181	1.60521	0.904255	0.96511
34-FOLNR+LR	0.945914	0.935709	0.860263	0.934522
35-FOLN+SON	0.908597	0.889848	0.729506	0.90607
36-FOLNR+SONR	0.969627	0.959288	0.882462	0.979058
37-FOLN+TON	0.945775	0.927183	0.856345	0.962914
38-FOLNR+TONR	0.990337	0.984391	0.953053	0.993353
39-FOLN+4thON	0.961509	0.939697	0.82581	0.930651
40-FOLNR+4thONR	0.993198	0.985223	0.966119	0.99948

Constraints were applied to the selected model and the maximum and minimum values given by different algorithms were recorded. Expanded model expressions of energy output are given in Appendix B and expanded model expressions of moisture output are given in Appendix C. The table containing the engineering boundedness of all models is given in Appendix D.

Table 4.9: Optimization Problem Results

Scenario No	Constraints	Optimization Algorithm	Total Energy (kWh)	Suggested Design
1	$65 < C < 85$ $0 < t < 300$	MDE	15.98	$x_1=67.60, x_2=57.24$
		MRS	15.98	$x_1=67.60, x_2=57.24$
		MNM	13.22	$x_1=73.80, x_2=33.44$
		MSA	34.38	$x_1=65.22, x_2=243.41$
2	$65 < C < 85$ $90 < t < 300$	MDE	11.69	$x_1=68.13, x_2=90.24$
		MRS	11.57	$x_1=65.74, x_2=92.56$
		MNM	11.69	$x_1=68.13, x_2=90.24$
		MSA	11.57	$x_1=65.74, x_2=92.56$
3	$65 < C < 85,$ $C \in \text{integers}$ $90 < t < 300,$ $t \in \text{integers}$	MDE	12.59	$x_1=69, x_2=93$
		MRS	13.07	$x_1=66, x_2=91$
		MNM	13.07	$x_1=66, x_2=91$
		MSA	12.29	$x_1=66, x_2=93$
4	$x_1 == 65 \parallel x_1 == 70 \parallel x_1 == 75 \parallel$ $x_1 == 80 \parallel x_1 == 85 \parallel$ $\parallel x_2 == 90 \parallel x_2 == 100 \parallel x_2 ==$ $110 \parallel x_2 == 120 \parallel x_2 == 130 \parallel$ $x_2 == 140 \parallel x_2 == 150 \parallel x_2 ==$ $160 \parallel$ $\parallel x_2 == 170 \parallel x_2 == 180 \parallel x_2 ==$ $190 \parallel x_2 == 200 \parallel x_2 == 210 \parallel$ $x_2 == 220 \parallel x_2 == 230 \parallel x_2 ==$ $240 \parallel$ $x_2 == 250 \parallel x_2 == 260 \parallel x_2 ==$ $270 \parallel x_2 == 280 \parallel x_2 == 290 \parallel$ $x_2 == 300$ $0.02 \leq \%M$	MDE	15.61	$x_1=75, x_2=100$
		MRS	15.61	$x_1=75, x_2=100$
		MNM	15.61	$x_1=75, x_2=100$
		MSA	15.61	$x_1=75, x_2=100$

In the tables, x_1 represents the drying temperature and x_2 represents the drying time.

When we examine the results of different scenarios created to bring the structure closer to reality; For two input values in Scenario 1, the first input value was defined within the machine value range, and the second input value was defined between the maximum and minimum ranges. As a result: the minimum total energy used was found to be 15.98 kWh. Recommended values are drying temperature (C°)=67.60, drying time (t)=57.24 minutes. The recommended values appear close to reality, but are not within practical limits.

In Scenario 2, realistic restrictions were imposed on the second input in order to obtain more convergent values. The first input value was defined within the machine value range, and the second input value was defined with reference to the values given by the manufacturers. As a result: the minimum total energy used was found to be 11.69 kWh. Recommended values are drying temperature (C°)=68.13, drying time (t)=90.24 minutes. Recommended values appear close to reality, but are not within practical limits.

In Scenario 3, the experiment was conducted by imposing all restrictions appropriate to the experimental data. Values are restricted to be integers. As a result: the minimum total energy used was found to be 12.59 kWh. Recommended values are drying temperature (C°)=69, drying time (t)=93 minutes. The recommended values seem close to reality, but it does not seem possible to apply them to the machine as integers.

In Scenario 4, more successful results were obtained by imposing all restrictions appropriate to the experimental data. Values are restricted to be integers. As a result: the minimum total energy used was found to be 15.61 kWh. Recommended values are drying temperature (C°)=75, drying time (t)=100 minutes. The recommended values seem close to reality, but it does not seem possible to apply them to the machine as integers.

The experimental data suggested in Table 4.9 have been tested experimentally. First of all, the technical revision made for this study enabled the machine to enter input as integers. Therefore, Scenario 3 values were used and the limitation of entering numbers 5 and multiples in Scenario 4 was eliminated. With the suggested model results, drying at 66 (C°) for 91 minutes, energy consumption; It was recorded as 13.9 kWh.

4.7. Conclusion

This research was conducted to investigate the effect of drying process variables of plastic granules. Tables 4.2-4.3-4.4 were measured and recorded as experimental data. Experimental data consisting of two inputs and two outputs were modeled mathematically and the success of the model was evaluated. Then, the results of four stochastic optimization methods (Differential Evaluation, Nelder-Mead, Random Seed, Simulated Annealing) of the selected successful model were evaluated. It has been observed that the more experimental data taken as reference or performed, the more successful the mathematical models are. Real experiments were conducted with the results obtained. When drying standards for 0.02% moisture are used, energy consumption while drying for 240 minutes at 75 (°C) temperature; While it was 36.1 kWh, the energy consumption was dried at 66 (°C) for 91 minutes with the suggested model results; It was recorded as 13.9 kWh. Thus, 60% energy savings were achieved in the granule drying process in a 400kg silo.

References

Journal article with one author & doi:

- [1] Fu, H., Xu, H., Liu, Y., Yang, Z., Kormakov, S., Wu, D., & Sun, J. Overview of injection molding technology for processing polymers and their composites. *ES Materials & Manufacturing* 2020, 8(4); 3-23. doi: 10.30919/esmm5f713
- [2] Barbosa, R. C. N., R. D. S. G. Campilho, and F. J. G. Silva. Injection mold design for a plastic component with blowing agent. *Procedia Manufacturing* 17 2018; 774-782. doi: 10.1016/J.PROMFG.2018.10.128
- [3] Stormonth-Darling, J. M., Saeed, A., Reynolds, P. M., & Gadegaard, N. Injection Molding Micro-and Nanostructures in Thermoplastic Elastomers. *Macromolecular materials and engineering* 2016, 301(8); 964-971. doi: 10.1002/MAME.201600011
- [4] Aditya, Chhabra., Karanbir, Singh., Himalaya, Kanwar. Effect of molding parameter on injection molding. *Scholarly Research Journal for Humanity Science & English Language* 4(24) 2017. doi: 10.21922/SRJHSEL.V4I24.10424
- [5] Akyürek, A. , Keçe, A. , Aksoy, A. , Öztürk, N. Termoplastik Ürünlerin Yeni Ürün Devreye Alma Sürecinde Bir Yapay Zeka Yaklaşımı. *Uludağ University Journal of The Faculty of Engineering* 14 2009. doi: 10.17482/UUJFE.68249
- [6] Wu, Jinbo., Wu, Yuzheng. 2016. Plastic injection molding process.
- [7] Chen, Peijuan. 2020. Polymer plastic drying device.
- [8] Sun, Xudong., Wang, Jianqiang., Wang, Jun., Zhang, Wenming. 2018. Method for drying high K value polymer solution and polymer powder.
- [9] Michael, P., Howard., Arash, Nikoubashman., Athanassios, Z., Panagiotopoulos. Stratification in Drying Polymer-Polymer and Colloid-

- Polymer Mixtures.. Langmuir. 2017. doi: 10.1021/ACS.LANGMUIR.7B02074
- [10] Kazutoshi, Haraguchi. Katsuhito, Kuroki. 2013. Non-drying polymer hydrogel.
- [11] Masakado, Akiro., Katagiri, Tomoaki. 2009. Method for drying polymer and dried polymer.
- [12] Tomi, Erfando. Optimization of Polymer Flooding Using Genetic Algorithm. Journal of Earth Energy Science, Engineering, and Technology 2023. doi: 10.25105/jeeset.v6i1.16319
- [13] Joze, Tavcar., Borut, Černe., Jože, Duhovnik., Damijan, Zorko. A multicriteria function for polymer gear design optimization. Journal of Computational Design and Engineering 2021. doi: 10.1093/JCDE/QWAA097
- [14] A., A., Kurganov., A., A., Korolev., E., N., Viktorova. Optimization of Conditions of Polymer Separation by Hydrodynamic Chromatography. Polymer Science Series A 2020. doi: 10.1134/S0965545X20060061
- [15] Fares, A., Al, Momani., Banu, Örmeci. Optimization of Polymer Dose Based on Residual Polymer Concentration in Dewatering Supernatant. Water Air and Soil Pollution 2014. doi: 10.1007/S11270-014-2154-Z
- [16] Pablo, A., Marchetti., Ignacio, E., Grossmann., Wiley, Bucey., Rita, Majewski. A Multiproduct Feedstock Optimization Model for Polymer Production. Computer-aided chemical engineering 2013. doi: 10.1016/B978-0-444-63234-0.50098-1
- [17] Antony J. Design of Experiments for Engineers and Scientists. 2nd ed. Elsevier; 2014.
- [18] Vecchio RD. Understanding Design of Experiments: a Primer for Technologists. Hanser/Gardner Publications, Inc., Cincinnati; 1997.

- [19] Cavazzuti M. Optimization Methods: from Theory to Design Scientific and Technological Aspects in Mechanics. Springer Science & Business Media; 2012.
- [20] Antony J. Some Key Things Industrial Engineers Should Know About Experimental Design. Logistics Information Management; 1998.
- [21] Montgomery DC, Runger GC, Hubele NF. Engineering Statistics. 5th ed. John Wiley & Sons; 2009.
- [22] Gunst RF, Mason RL. How to Construct Fractional Factorial Experiments. ASQC Quality Press; 1991.
- [23] Montgomery DC. Design and Analysis of Experiments. 8th ed. John Wiley & Sons; 2017.
- [24] Polatođlu İ, Aydın L. A new design strategy with stochastic optimization on the preparation of magnetite cross-linked tyrosinase aggregates (MCLTA). Process Biochemistry 2020; 99: 131-8.
- [25] Aydın L, Artem HS, Oterkus S. (Ed.) Designing Engineering Structures using Stochastic Optimization Methods, 1st ed. CRC Press; 2020. <https://doi.org/10.1201/9780429289576>
- [26] Turhan, F. 1100 serisi alüminyum malzemelerde tig kaynađı ile oluřan kaynak dikiř geometrisinin optimizasyonu (master's thesis). İzmir: İzmir Katip Çelebi University Fen Bilimleri Enstitüsü; 2017. <https://tez.yok.gov.tr/>
- [27] Ombach J. A short introduction to stochastic optimization. Schedae Informaticae 2014; 23: 9–20.
- [28] Rao RV, Savsani VJ. Mechanical Design Optimization using Advanced Optimization Techniques, Springer; 2012.
- [29] Tech Target. <https://whatis.techtarget.com/definition/stochastic-optimization>.

- [30] 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
- [31] 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
- [32] 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.
- [33] Zakaria A, Firas BI, Hossain-Lipu MS, Hannan MA. Uncertainty models for stochastic optimization in renewable energy applications. *Renewable Energy* 2019; 1543–1571.
- [34] 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
- [35] 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.
- [36] 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.
- [37] 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.
- [38] Tifkitsis KI, Mesogitis TS, Struzziero G, Skordos AA. Stochastic multiobjective optimisation of the cure process of thick laminates. *Composites Part A* 112 2018; 383–394.

- [39] 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.
- [40] 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.
- [41] 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
- [42] 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
- [43] Das S, Suganthan PN, Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* 2011; 15(1): 4–35.
- [44] Jujjavarapu S, Chimmiri V. Dynamic modeling and metabolic flux analysis for optimized production of rhamnolipids. *Chemical Engineering Communications* 2016; 203; 326–338.
- [45] Arunachalam V. Optimization using Differential Evolution (M.S. Thesis). London: The University of Western Ontario; 2008.
- [46] Nelder JA, Mead R. A simplex method for function minimization. *Comput. J.* 1965; 7(4): 308–313.
- [47] Barati R. Parameter estimation of nonlinear Muskingum models using Nelder–Mead simplex algorithm. *J. Hydrol. Eng.* 2011; 16 (11): 946–954.
- [48] Karnopp DC, Random search techniques for optimization problems. *Automatica* 1963; 1 (2–3): 111–121.
- [49] Zabinsky ZB, Random search algorithms. *Wiley Encyclopedia of Operations Research and Management Science*; 2009.

- [50] Wolfram Mathematica software 11.3.
- [51] Momin, J., Xin-She, Y. and Hans-Jürgen, Z. 2013. Test functions for global optimization: A comprehensive survey. pp. 193–222. In: Xin-She, Y., Zhihua, C., Renbin, X., Amir Hossein, G. and Mehmet, K. [eds.]. *Swarm Intelligence and Bio-Inspired Computation*, Elsevier.
- [52] Karaboğa, D. 2018. *Yapay zeka optimizasyon algoritmaları*. Nobel akademik yayıncılık, Ankara.
- [53] Ingber, L. 1993. Simulated annealing: Practice versus theory. *Math. Comput. Model.* 18(11): 29–57.
- [54] Savran M, Aydın L. Stochastic optimization of graphite-flax/epoxy hybrid laminated composite for maximum fundamental frequency and minimum cost. *Engineering Structures* 2018; 174: 675–687.
- [55] 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.
- [56] 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.
- [57] Sofroniou M, Knapp R. *Advanced Numerical Differential Equation Solving In Mathematica*. Wolfram Mathematica Tutorial Collection 2008; Retrieved from [https:// library.wolfram.com](https://library.wolfram.com).
- [58] Wellin PR, Gaylord RJ, Kamin SN. *An Introduction to Programming with Mathematica*. Cambridge University Press; 2005.

Appendices

Appendix A

Draft Design Chart

1	Level 1 of A	Level 1 of B
2	Level 1 of A	Level 2 of B
3	Level 1 of A	Level 3 of B
4	Level 1 of A	Level 4 of B
5	Level 1 of A	Level 5 of B
6	Level 1 of A	Level 6 of B
7	Level 1 of A	Level 7 of B
8	Level 1 of A	Level 8 of B
9	Level 1 of A	Level 9 of B
10	Level 1 of A	Level 10 of B
11	Level 1 of A	Level 11 of B
12	Level 1 of A	Level 12 of B
13	Level 1 of A	Level 13 of B
14	Level 1 of A	Level 14 of B
15	Level 1 of A	Level 15 of B
16	Level 1 of A	Level 16 of B
17	Level 1 of A	Level 17 of B
18	Level 1 of A	Level 18 of B
19	Level 1 of A	Level 19 of B
20	Level 1 of A	Level 20 of B
21	Level 1 of A	Level 21 of B
22	Level 2 of A	Level 1 of B
23	Level 2 of A	Level 2 of B
24	Level 2 of A	Level 3 of B
25	Level 2 of A	Level 4 of B
26	Level 2 of A	Level 5 of B
27	Level 2 of A	Level 6 of B
28	Level 2 of A	Level 7 of B
29	Level 2 of A	Level 8 of B

30	Level 2 of A	Level 9 of B
31	Level 2 of A	Level 10 of B
32	Level 2 of A	Level 11 of B
33	Level 2 of A	Level 12 of B
34	Level 2 of A	Level 13 of B
35	Level 2 of A	Level 14 of B
36	Level 2 of A	Level 15 of B
37	Level 2 of A	Level 16 of B
38	Level 2 of A	Level 17 of B
39	Level 2 of A	Level 18 of B
40	Level 2 of A	Level 19 of B
41	Level 2 of A	Level 20 of B
42	Level 2 of A	Level 21 of B
43	Level 3 of A	Level 1 of B
44	Level 3 of A	Level 2 of B
45	Level 3 of A	Level 3 of B
46	Level 3 of A	Level 4 of B
47	Level 3 of A	Level 5 of B
48	Level 3 of A	Level 6 of B
49	Level 3 of A	Level 7 of B
50	Level 3 of A	Level 8 of B
51	Level 3 of A	Level 9 of B
52	Level 3 of A	Level 10 of B
53	Level 3 of A	Level 11 of B
54	Level 3 of A	Level 12 of B
55	Level 3 of A	Level 13 of B
56	Level 3 of A	Level 14 of B
57	Level 3 of A	Level 15 of B
58	Level 3 of A	Level 16 of B
59	Level 3 of A	Level 17 of B
60	Level 3 of A	Level 18 of B
61	Level 3 of A	Level 19 of B
62	Level 3 of A	Level 20 of B
63	Level 3 of A	Level 21 of B

Appendix B

Expanded Model Expressions of the Case Studies – Energy Output

Models Name	Models
1-L	$-21.9331 + 0.289894 x_1 + 0.161759 x_2$
2-LR	$(-7777.01 + 56.9468 x_1 + 6842.24 x_2)/(78600.4 - 483.771 x_1 + 2.69181 x_2)$
3-SON	$111.467 - 3.0248 x_1 + 0.0202703 x_1^2 + 0.0328197 x_2 + 0.00172591 x_1 x_2 - 7.3386 \cdot 10^{-6} x_2^2$
4-SONR	$(-30774.6 + 554.359 x_1 - 2.31006 x_1^2 + 6276.09 x_2 - 74.44 x_1 x_2 + 4.6039 x_2^2)/(26293.6 - 114.857 x_1 - 2.34895 x_1^2 + 62.8224 x_2 - 0.427037 x_1 x_2 - 0.0021201 x_2^2)$
5-TON	$16.5977 - 0.260945 x_1 - 0.00288916 x_1^2 + 0.0000438868 x_1^3 + 0.579852 x_2 - 0.0121952 x_1 x_2 + 0.0000887332 x_1^2 x_2 - 0.00020572 x_2^2 + 1.87585 \cdot 10^{-6} x_1 x_2^2 + 1.09167 \cdot 10^{-7} x_2^3$
6-TONR	$(-2.58648 \cdot 10^6 + 152756. x_1 - 2696.3 x_1^2 + 14.7827 x_1^3 - 652407. x_2 + 18153.1 x_1 x_2 - 122.941 x_1^2 x_2 - 286.732 x_2^2 + 2.75221 x_1 x_2^2 + 0.37027 x_2^3)/(-2.0604 \cdot 10^6 + 36244.8 x_1 + 218.544 x_1^2 - 4.2118 x_1^3 - 5107.08 x_2 + 96.1947 x_1 x_2 - 0.468245 x_1^2 x_2 + 5.19197 x_2^2 - 0.0339752 x_1 x_2^2 - 0.000671711 x_2^3)$
7-4thON	$11.3675 - 0.0893683 x_1 - 0.00199103 x_1^2 - 0.0000100943 x_1^3 + 3.20919 \cdot 10^{-7} x_1^4 + 0.219245 x_2 - 0.00217333 x_1 x_2 - 0.0000231139 x_1^2 x_2 + 5.5775 \cdot 10^{-7} x_1^3 x_2 + 0.00126262 x_2^2 - 0.0000131368 x_1 x_2^2 - 5.5352 \cdot 10^{-8} x_1^2 x_2^2 - 3.92284 \cdot 10^{-6} x_2^3 + 5.1732 \cdot 10^{-8} x_1 x_2^3 + 2.7604 \cdot 10^{-10} x_2^4$
8-4thONR	$(-7.22376 \cdot 10^8 + 2.21787 \cdot 10^7 x_1 - 52558.6 x_1^2 - 3477.37 x_1^3 + 25.9284 x_1^4 + 2.58637 \cdot 10^7 x_2 + 215319. x_1 x_2 - 15614.8 x_1^2 x_2 + 110.469 x_1^3 x_2 - 815129. x_2^2 + 18476.9 x_1 x_2^2 - 102.245 x_1^2 x_2^2 + 917.256 x_2^3 - 13.1132 x_1 x_2^3 + 0.469367 x_2^4)/(2.63726 \cdot 10^8 - 654528. x_1 - 89751.8 x_1^2 + 622.839 x_1^3 + 0.998409 x_1^4 - 4.98456 \cdot 10^6 x_2 + 85568.6 x_1 x_2 + 69.197 x_1^2 x_2 - 4.29018 x_1^3 x_2 + 10255.9 x_2^2 - 207.796 x_1 x_2^2 + 0.772572 x_1^2 x_2^2 + 6.19006 x_2^3 - 0.00475207 x_1 x_2^3 - 0.00461814 x_2^4)$
9-5thON	$2.13396 - 0.00111265 x_1 - 0.00020523 x_1^2 - 2.80514 \cdot 10^{-6} x_1^3 - 1.45122 \cdot 10^{-8} x_1^4 + 2.86932 \cdot 10^{-10} x_1^5 + 0.388169 x_2 - 0.00394147 x_1 x_2 - 0.0000759043 x_1^2 x_2 - 2.26404 \cdot 10^{-7} x_1^3 x_2 + 1.67822 \cdot 10^{-8} x_1^4 x_2 - 0.00060666 x_2^2 + 0.000074933 x_1 x_2^2 + 5.12302 \cdot 10^{-7} x_1^2 x_2^2 - 1.5954 \cdot 10^{-8} x_1^3 x_2^2 - 0.0000133636 x_2^3 - 4.74287 \cdot 10^{-7} x_1 x_2^3 + 6.99304 \cdot 10^{-9} x_1^2 x_2^3 + 1.00205 \cdot 10^{-7} x_2^4 - 8.76734 \cdot 10^{-10} x_1 x_2^4 - 4.54219 \cdot 10^{-11} x_2^5$

10-5thONR	$(-5.91724*10^{11} + 1.5924*10^{10} x_1 - 5.31164*10^7 x_1^2 - 586354. x_1^3 - 12750.8 x_1^4 + 145.54 x_1^5 - 1.31842*10^9 x_2 - 6.0386*10^6 x_1 x_2 + 1.25652*10^6 x_1^2 x_2 - 14884. x_1^3 x_2 + 52.8742 x_1^4 x_2 + 7.7203*10^7 x_2^2 - 2.50593*10^6 x_1 x_2^2 + 18641.3 x_1^2 x_2^2 - 45.9843 x_1^3 x_2^2 - 272708. x_2^3 + 12698.6 x_1 x_2^3 - 56.8033 x_1^2 x_2^3 - 1223.88 x_2^4 - 13.2575 x_1 x_2^4 + 5.03118 x_2^5)/(-2.76554*10^{10} + 6.70612*10^8 x_1 + 1.47781*10^6 x_1^2 - 99462.1 x_1^3 + 313.035 x_1^4 + 2.0602 x_1^5 + 1.15013*10^8 x_2 - 5.76438*10^6 x_1 x_2 + 28856.1 x_1^2 x_2 + 59.0248 x_1^3 x_2 - 0.629819 x_1^4 x_2 + 1.1102*10^6 x_2^2 + 42559.7 x_1 x_2^2 - 573.755 x_1^2 x_2^2 + 2.79843 x_1^3 x_2^2 - 27083.2 x_2^3 + 271.068 x_1 x_2^3 - 1.40554 x_1^2 x_2^3 + 61.3238 x_2^4 - 0.33473 x_1 x_2^4 - 0.00752563 x_2^5)$
11-FOTN	$-0.438492 + 0.000597981 x_1 + 0.161427 x_2 - 3.00312 \cos[x_1] + 0.158221 \cos[x_2] - 4.43859 \sin[x_1] - 0.0143774 \sin[x_2]$
12-FOTNR	$(40661.6 + 440.814 x_1 - 95.2483 x_2 + 24601.6 \cos[x_1] + 2182.82 \cos[x_2] + 32841.9 \sin[x_1] - 44598.4 \sin[x_2])/(-6563.82 + 143.09 x_1 - 10.2172 x_2 + 1969.13 \cos[x_1] - 17.4551 \cos[x_2] + 3584.69 \sin[x_1] - 1631.43 \sin[x_2])$
13-SOTN	$6.57696 - 4.3608 \cos[x_1] + 9.158 \cos[x_1]^2 - 5.46193 \cos[x_2] - 2.9776 \cos[x_1] \cos[x_2] + 1.66898 \cos[x_2]^2 - 1.56304 \sin[x_1] + 6.79116 \sin[x_1]^2 - 5.10525 \sin[x_2] + 0.387997 \sin[x_1] \sin[x_2] + 16.4044 \sin[x_2]^2$
14-SOTNR	$(16.0914 - 2.83823 \cos[x_1] + 13.4868 \cos[x_1]^2 + 23.4494 \cos[x_2] - 0.687061 \cos[x_1] \cos[x_2] - 14.2928 \cos[x_2]^2 - 10.9968 \sin[x_1] + 32.2789 \sin[x_1]^2 - 5.07733 \sin[x_2] + 0.00121628 \sin[x_1] \sin[x_2] - 12.531 \sin[x_2]^2)/(0.920622 + 0.250771 \cos[x_1] + 0.834167 \cos[x_1]^2 + 1.11558 \cos[x_2] + 0.00353371 \cos[x_1] \cos[x_2] - 0.2923 \cos[x_2]^2 + 1.11193 \sin[x_1] - 0.99932 \sin[x_1]^2 - 0.175298 \sin[x_2] + 0.00531203 \sin[x_1] \sin[x_2] - 0.415149 \sin[x_2]^2)$
15-TOTN	$6.52008 - 3.17251 \cos[x_1] + 9.01618 \cos[x_1]^2 - 3.859 \cos[x_1]^3 - 1.8912 \cos[x_2] - 2.75515 \cos[x_1] \cos[x_2] - 1.92103 \cos[x_1]^2 \cos[x_2] + 1.83429 \cos[x_2]^2 + 3.81503 \cos[x_1] \cos[x_2]^2 - 2.67568 \cos[x_2]^3 - 1.96928 \sin[x_1] + 6.97295 \sin[x_1]^2 + 4.25055 \sin[x_1]^3 - 10.7246 \sin[x_2] - 10.9571 \sin[x_1] \sin[x_2] + 22.8285 \sin[x_1]^2 \sin[x_2] + 16.0737 \sin[x_2]^2 - 2.44628 \sin[x_1] \sin[x_2]^2 - 0.166613 \sin[x_2]^3$
16-TOTNR	$(338.723 - 1444.81 \cos[x_1] - 1448.63 \cos[x_1]^2 + 1190.03 \cos[x_1]^3 + 1898.91 \cos[x_2] + 846.619 \cos[x_1] \cos[x_2] - 2770.44 \cos[x_1]^2 \cos[x_2] - 139.594 \cos[x_2]^2 + 1442.71 \cos[x_1] \cos[x_2]^2 - 612.979 \cos[x_2]^3 - 121.534 \sin[x_1] + 185.93 \sin[x_1]^2 + 901.832 \sin[x_1]^3 - 267.356 \sin[x_2] - 910.611 \sin[x_1] \sin[x_2] + 1673.28 \sin[x_1]^2 \sin[x_2] + 305.897 \sin[x_2]^2 - 1843.9 \sin[x_1] \sin[x_2]^2 - 267.951 \sin[x_2]^3)/(60.096 + 310.438 \cos[x_1] - 1.70268 \cos[x_1]^2 - 327.646 \cos[x_1]^3 + 78.8011 \cos[x_2] + 33.0701 \cos[x_1] \cos[x_2] - 113.031 \cos[x_1]^2 \cos[x_2] - 74.0187 \cos[x_2]^2 + 57.2544 \cos[x_1] \cos[x_2]^2 - 27.8132 \cos[x_2]^3 + 158.356 \sin[x_1] - 21.6359 \sin[x_1]^2 + 88.5441 \sin[x_1]^3 - 7.78449 \sin[x_2] - 32.9114 \sin[x_1] \sin[x_2] + 57.0037 \sin[x_1]^2 \sin[x_2] - 56.4131 \sin[x_2]^2 - 73.0478 \sin[x_1] \sin[x_2]^2 - 10.2198 \sin[x_2]^3)$
17-FOLN	$-43.1379 - 4.09369 \log[x_2] + 15.6471 \log[x_1]$
18-FOLNR	$(108.198 - 5.37001 \log[x_2] - 23.1458 \log[x_1])/(5.46204 - 0.201769 \log[x_2] - 1.18436 \log[x_1])$
19-SOLN	$5265.91 - 6.60562 \log[x_2] - 12.654 \log[x_2]^2 - 2449.29 \log[x_1] + 0.562544 \log[x_2] \log[x_1] + 286.268 \log[x_1]^2$
20-SOLNR	$(247.468 + 1527.14 \log[x_2] - 104.439 \log[x_2]^2 + 279. \log[x_1] - 42.207 \log[x_2] \log[x_1] - 70.3956 \log[x_1]^2)/(8.66624 + 79.7473 \log[x_2] - 4.03242 \log[x_2]^2 + 15.5922 \log[x_1] -$

	$17.8812 \text{ Log}[x_2] \text{ Log}[x_1] - 3.76287 \text{ Log}[x_1]^2)$
21-SOTNR+LR	$(7333.5 + 88.5054 x_1 - 896.674 x_2 + 396.149 \text{ Cos}[x_1] + 26872.8 \text{ Cos}[x_1]^2 - 19582.8 \text{ Cos}[x_2] - 114553. \text{ Cos}[x_1] \text{ Cos}[x_2] + 70463.8 \text{ Cos}[x_2]^2 - 83611.6 \text{ Sin}[x_1] - 84207.6 \text{ Sin}[x_1]^2 + 32809.9 \text{ Sin}[x_2] + 132889. \text{ Sin}[x_1] \text{ Sin}[x_2] - 36533.2 \text{ Sin}[x_2]^2)/(7246.5 + 25.8093 x_1 - 27.021 x_2 + 780.14 \text{ Cos}[x_1] + 68.7036 \text{ Cos}[x_1]^2 - 758.895 \text{ Cos}[x_2] - 4386.74 \text{ Cos}[x_1] \text{ Cos}[x_2] - 4887.92 \text{ Cos}[x_2]^2 + 608.675 \text{ Sin}[x_1] - 11920.4 \text{ Sin}[x_1]^2 + 1706.9 \text{ Sin}[x_2] + 5374.45 \text{ Sin}[x_1] \text{ Sin}[x_2] - 7869.99 \text{ Sin}[x_2]^2)$
22-TOTNR+LR	$(2133. + 320.635 x_1 - 205.903 x_2 - 12084.8 \text{ Cos}[x_1] - 5576.91 \text{ Cos}[x_1]^3 + 9750.25 \text{ Cos}[x_2] - 20141.4 \text{ Cos}[x_1]^2 \text{ Cos}[x_2] - 11069.1 \text{ Cos}[x_1] \text{ Cos}[x_2]^2 + 7243.55 \text{ Cos}[x_2]^3 - 1405.03 \text{ Sin}[x_1] + 15394.1 \text{ Sin}[x_1]^3 - 6714.19 \text{ Sin}[x_2] - 38801. \text{ Sin}[x_1]^2 \text{ Sin}[x_2] + 42111.3 \text{ Sin}[x_1] \text{ Sin}[x_2]^2 + 41073.5 \text{ Sin}[x_2]^3)/(1488.49 + 4.66931 x_1 - 9.49081 x_2 - 256.72 \text{ Cos}[x_1] - 385.345 \text{ Cos}[x_1]^3 + 176.28 \text{ Cos}[x_2] - 764.453 \text{ Cos}[x_1]^2 \text{ Cos}[x_2] - 569.073 \text{ Cos}[x_1] \text{ Cos}[x_2]^2 + 514.395 \text{ Cos}[x_2]^3 + 2758.58 \text{ Sin}[x_1] - 4566.93 \text{ Sin}[x_1]^3 - 291.338 \text{ Sin}[x_2] - 1419.3 \text{ Sin}[x_1]^2 \text{ Sin}[x_2] + 1974.05 \text{ Sin}[x_1] \text{ Sin}[x_2]^2 + 1672.5 \text{ Sin}[x_2]^3)$
23-SOLNR+LR	$(2036.91 + 14.8309 x_1 + 1.80667 x_2 + 17.1551 \text{ Log}[x_2] - 0.704283 \text{ Log}[x_2]^2 - 368.864 \text{ Log}[x_1] - 3.49455 \text{ Log}[x_2] \text{ Log}[x_1] - 83.8714 \text{ Log}[x_1]^2)/(-145.263 - 0.875921 x_1 + 0.000754009 x_2 + 0.841054 \text{ Log}[x_2] - 0.0270605 \text{ Log}[x_2]^2 + 46.1167 \text{ Log}[x_1] - 0.182322 \text{ Log}[x_2] \text{ Log}[x_1] + 1.25115 \text{ Log}[x_1]^2)$
24-SOTN+TON	$-0.140345 - 0.00104205 x_1 - 2.8312*10^{-6} x_1^2 + 9.38326*10^{-8} x_1^3 + 0.595734 x_2 - 0.0123896 x_1 x_2 + 0.0000895374 x_1^2 x_2 - 0.000251451 x_2^2 + 2.0293*10^{-6} x_1 x_2^2 + 1.7505*10^{-7} x_2^3 - 0.598247 \text{ Cos}[x_1] + 0.00524295 \text{ Cos}[x_1]^2 + 0.175067 \text{ Cos}[x_2] + 0.00719196 \text{ Cos}[x_1] \text{ Cos}[x_2] + 0.0533476 \text{ Cos}[x_2]^2 - 0.307537 \text{ Sin}[x_1] - 0.915619 \text{ Sin}[x_1]^2 + 0.0847963 \text{ Sin}[x_2] + 0.127651 \text{ Sin}[x_1] \text{ Sin}[x_2] - 0.44356 \text{ Sin}[x_2]^2$
25-SOTNR+TONR	$(1.7473*10^7 - 21085.8 x_1 - 4799.49 x_1^2 - 42.8254 x_1^3 - 535167. x_2 + 16530.7 x_1 x_2 - 121.218 x_1^2 x_2 - 1304.63 x_2^2 + 14.7893 x_1 x_2^2 + 1.13081 x_2^3 + 1.41614*10^6 \text{ Cos}[x_1] + 1.07223*10^7 \text{ Cos}[x_1]^2 + 14025.5 \text{ Cos}[x_2] + 30404.4 \text{ Cos}[x_1] \text{ Cos}[x_2] + 4.50259*10^7 \text{ Cos}[x_2]^2 + 3.40713*10^7 \text{ Sin}[x_1] - 8.74425*10^7 \text{ Sin}[x_1]^2 + 94.89 \text{ Sin}[x_2] - 17215.4 \text{ Sin}[x_1] \text{ Sin}[x_2] + 4.51065*10^7 \text{ Sin}[x_2]^2)/(1.33924*10^6 - 3195.38 x_1 + 2.9064 x_1^2 - 2.88511 x_1^3 - 22010.3 x_2 + 425.835 x_1 x_2 - 2.05851 x_1^2 x_2 + 18.0079 x_2^2 - 0.107341 x_1 x_2^2 - 0.0051834 x_2^3 - 1.06069*10^6 \text{ Cos}[x_1] - 634958. \text{ Cos}[x_1]^2 + 891.224 \text{ Cos}[x_2] + 1730.96 \text{ Cos}[x_1] \text{ Cos}[x_2] - 667141. \text{ Cos}[x_2]^2 - 4.23654*10^6 \text{ Sin}[x_1] + 5.34844*10^6 \text{ Sin}[x_1]^2 + 446.237 \text{ Sin}[x_2] - 1036.18 \text{ Sin}[x_1] \text{ Sin}[x_2] - 663615. \text{ Sin}[x_2]^2)$
26-SOTN+4thON	$-0.0955151 - 0.00105929 x_1 - 0.0000106698 x_1^2 - 9.44749*10^{-8} x_1^3 - 6.67653*10^{-10} x_1^4 + 0.19581 x_2 - 0.00177917 x_1 x_2 - 0.0000187827 x_1^2 x_2 + 4.92891*10^{-7} x_1^3 x_2 + 0.00147901 x_2^2 - 0.0000182999 x_1 x_2^2 - 2.0918*10^{-8} x_1^2 x_2^2 - 4.20667*10^{-6} x_2^3 + 5.20916*10^{-8} x_1 x_2^3 + 8.23978*10^{-10} x_2^4 - 0.328351 \text{ Cos}[x_1] - 0.102768 \text{ Cos}[x_1]^2 + 0.161134 \text{ Cos}[x_2] + 0.0291301 \text{ Cos}[x_1] \text{ Cos}[x_2] + 0.13898 \text{ Cos}[x_2]^2 + 0.133171 \text{ Sin}[x_1] - 0.214735 \text{ Sin}[x_1]^2 + 0.115924 \text{ Sin}[x_2] + 0.100275 \text{ Sin}[x_1] \text{ Sin}[x_2] - 0.409795 \text{ Sin}[x_2]^2$
27-SOTNR+4thONR	$(4.64714*10^6 + 7021.07 x_1 - 814.721 x_1^2 - 25.8387 x_1^3 + 0.395083 x_1^4 + 2.546*10^7 x_2 - 31349.3 x_1 x_2 - 1883.87 x_1^2 x_2 - 17.6464 x_1^3 x_2 - 61560.8 x_2^2 - 7308.88 x_1 x_2^2 + 98.8004 x_1^2 x_2^2 + 1215.55 x_2^3 - 19.4795 x_1 x_2^3 + 2.77512 x_2^4 - 2.6528*10^7 \text{ Cos}[x_1] - 2.8575*10^7 \text{ Cos}[x_1]^2 - 1.44771*10^6 \text{ Cos}[x_2] + 2.0372*10^6 \text{ Cos}[x_1] \text{ Cos}[x_2] - 2.57996*10^7 \text{ Cos}[x_2]^2 - 1.21337*10^8 \text{ Sin}[x_1] + 1.49174*10^8 \text{ Sin}[x_1]^2 - 722243. \text{ Sin}[x_2] - 8.71854*10^6 \text{ Sin}[x_1] \text{ Sin}[x_2] - 2.43819*10^7 \text{ Sin}[x_2]^2)/(-5.60292*10^6 - 48587.1 x_1 - 435.359 x_1^2 - 4.2769 x_1^3 + 0.874916 x_1^4 + 331718. x_2 - 29980.2 x_1 x_2 - 66.3603 x_1^2 x_2 + 4.81939 x_1^3 x_2 - 14389.7 x_2^2 + 350.151 x_1 x_2^2 - 2.65734 x_1^2 x_2^2 + 49.42 x_2^3 - 0.310336 x_1 x_2^3 - 0.0139878 x_2^4 + 2.65183*10^7 \text{ Cos}[x_1] - 1.68261*10^6 \text{ Cos}[x_1]^2 - 98976.2 \text{ Cos}[x_2] + 160267. \text{ Cos}[x_1] \text{ Cos}[x_2] - 18569*10^7 \text{ Cos}[x_2]^2 + 2.24858*10^7 \text{ Sin}[x_1] + 1.00928*10^8 \text{ Sin}[x_1]^2 - 33578.1 \text{ Sin}[x_2] - 517955. \text{ Sin}[x_1] \text{ Sin}[x_2] - 1.20816*10^7 \text{ Sin}[x_2]^2)$
28-SOTN+5thON	$-0.0099731 - 0.000541734 x_1 - 0.0000120441 x_1^2 - 2.10715*10^{-7} x_1^3 - 3.26139*10^{-9} x_1^4 - 4.67847*10^{-11} x_1^5 + 0.346837 x_2 - 0.0033633 x_1 x_2 - 0.000065053 x_1^2 x_2 -$

	$1.73432*10^{-7} x_1^3 x_2 + 1.51227*10^{-8} x_1^4 x_2 + 3.29946*10^{-6} x_2^2 + 0.0000571986 x_1 x_2^2 + 3.27266*10^{-7} x_1^2 x_2^2 - 1.3571*10^{-8} x_1^3 x_2^2 - 0.0000137083 x^3 - 3.12572*10^{-7} x_1 x_2^3 + 6.10734*10^{-9} x_1^2 x_2^3 + 7.80261*10^{-8} x_2^4 - 9.27297*10^{-10} x_1 x_2^4 - 1.01067*10^{-11} x_2^5 + 0.195688 \cos[x_1] - 0.122197 \cos[x_1]^2 + 0.173385 \cos[x_2] - 0.0403611 \cos[x_1] \cos[x_2] + 0.196717 \cos[x_2]^2 + 0.28996 \sin[x_1] + 0.405695 \sin[x_1]^2 + 0.0889131 \sin[x_2] + 0.198345 \sin[x_1] \sin[x_2] - 0.234305 \sin[x_2]^2$
29-SOTNR+5thONR	$(-7.57215*10^8 - 1.29622*10^7 x_1 - 177278. x_1^2 - 1947.02 x_1^3 - 12.3385 x_1^4 + 1.0507 x_1^5 + 5.8846*10^9 x_2 - 3.42052*10^7 x_1 x_2 - 688108. x_1^2 x_2 + 1113.54 x_1^3 x_2 + 29.7283 x_1^4 x_2 - 5.24229*10^7 x_2^2 - 178146. x_1 x_2^2 + 2078.65 x_1^2 x_2^2 + 76.3378 x_1^3 x_2^2 + 308182. x_2^3 + 282.27 x_1 x_2^3 - 31.4937 x_1^2 x_2^3 + 348.044 x_2^4 - 10.1731 x_1 x_2^4 + 0.559587 x_2^5 - 3.28754*10^9 \cos[x_1] - 1.71218*10^9 \cos[x_1]^2 + 3.29741*10^8 \cos[x_2] + 2.71996*10^8 \cos[x_1] \cos[x_2] - 3.46921*10^9 \cos[x_2]^2 - 1.20318*10^{10} \sin[x_1] + 1.87884*10^{10} \sin[x_1]^2 - 1.88787*10^7 \sin[x_2] + 5.34581*10^8 \sin[x_1] \sin[x_2] - 2.01578*10^9 \sin[x_2]^2)/(-6.92656*10^8 - 7.78012*10^6 x_1 - 81676. x_1^2 - 513.936 x_1^3 + 7.04989 x_1^4 + 1.41369 x_1^5 - 4.28251*10^8 x_2 - 681239. x_1 x_2 + 21643.9 x_1^2 x_2 + 313.269 x_1^3 x_2 + 0.220191 x_1^4 x_2 + 1.92947*10^6 x_2^2 + 10162.1 x_1 x_2^2 + 94.8713 x_1^2 x_2^2 - 2.48495 x_1^3 x_2^2 + 7318.25 x_2^3 - 371.329 x_1 x_2^3 + 1.71496 x_1^2 x_2^3 + 24.4251 x_2^4 + 0.0697913 x_1 x_2^4 - 0.0310641 x_2^5 + 1.37124*10^9 \cos[x_1] + 1.81178*10^8 \cos[x_1]^2 + 1.51708*10^7 \cos[x_2] + 1.19367*10^7 \cos[x_1] \cos[x_2] - 9.95582*10^8 \cos[x_2]^2 - 3.63799*10^9 \sin[x_1] + 2.41953*10^{10} \sin[x_1]^2 - 596607. \sin[x_2] + 1.29803*10^7 \sin[x_1] \sin[x_2] - 9.96127*10^8 \sin[x_2]^2)$
30-TOTN+TON	$-0.0717383 - 0.000568824 x_1 - 2.38021*10^{-6} x_1^2 + 3.05374*10^{-8} x_1^3 + 0.655957 x_2 - 0.0141489 x_1 x_2 + 0.000101293 x_1^2 x_2 - 0.000220343 x_2^2 + 1.86519*10^{-6} x_1 x_2^2 + 1.63652*10^{-7} x_2^3 - 0.273484 \cos[x_1] - 0.00476383 \cos[x_1]^2 - 0.387281 \cos[x_1]^3 - 0.088718 \cos[x_2] - 0.0823063 \cos[x_1] \cos[x_2] + 0.461116 \cos[x_1]^2 \cos[x_2] + 0.110016 \cos[x_2]^2 + 0.0709806 \cos[x_1] \cos[x_2]^2 - 0.281944 \cos[x_2]^3 - 0.148308 \sin[x_1] - 0.439435 \sin[x_1]^2 - 0.373843 \sin[x_1]^3 - 1.1065 \sin[x_2] + 0.0616452 \sin[x_1] \sin[x_2] + 0.20165 \sin[x_1]^2 \sin[x_2] - 0.313705 \sin[x_2]^2 - 0.420237 \sin[x_1] \sin[x_2]^2 + 1.52303 \sin[x_2]^3$
31-TOTNR+TONR	$(3.4018*10^8 + 3.26127*10^6 x_1 + 18641.3 x_1^2 + 176.819 x_1^3 + 340092. x_2 - 9329.26 x_1 x_2 + 62.8696 x_1^2 x_2 - 301.825 x_2^2 + 3.57896 x_1 x_2^2 + 0.422679 x_2^3 + 3.02228*10^8 \cos[x_1] - 2.74168*10^8 \cos[x_1]^2 + 7.65348*10^7 \cos[x_1]^3 - 100413. \cos[x_2] - 7769.18 \cos[x_1] \cos[x_2] + 4202.55 \cos[x_1]^2 \cos[x_2] - 3.80119*10^7 \cos[x_2]^2 + 25843.3 \cos[x_1] \cos[x_2]^2 + 124200. \cos[x_2]^3 + 3.59726*10^8 \sin[x_1] - 3.44634*10^9 \sin[x_1]^2 + 2.98537*10^9 \sin[x_1]^3 - 17878.4 \sin[x_2] - 6878.28 \sin[x_1] \sin[x_2] + 13630.6 \sin[x_1]^2 \sin[x_2] - 3.80348*10^7 \sin[x_2]^2 + 41969.6 \sin[x_1] \sin[x_2]^2 + 20813.5 \sin[x_2]^3)/(2.8002*10^6 - 193934. x_1 + 296.077 x_1^2 - 19.4229 x_1^3 - 6515.78 x_2 + 145.785 x_1 x_2 - 0.785409 x_1^2 x_2 + 4.78022 x_2^2 - 0.0374808 x_1 x_2^2 + 0.000855661 x_2^3 - 3.05323*10^6 \cos[x_1] + 6.17525*10^6 \cos[x_1]^2 - 1.51268*10^7 \cos[x_1]^3 - 1667.88 \cos[x_2] - 328.015 \cos[x_1] \cos[x_2] - 106.064 \cos[x_1]^2 \cos[x_2] - 6.25197*10^6 \cos[x_2]^2 + 1227.73 \cos[x_1] \cos[x_2]^2 + 2091.06 \cos[x_2]^3 - 3.63546*10^6 \sin[x_1] + 1.54832*10^8 \sin[x_1]^2 - 1.57591*10^8 \sin[x_1]^3 - 334.202 \sin[x_2] - 639.339 \sin[x_1] \sin[x_2] + 973.345 \sin[x_1]^2 \sin[x_2] - 6.25324*10^6 \sin[x_2]^2 + 2760.55 \sin[x_1] \sin[x_2]^2 + 186.185 \sin[x_2]^3)$
32-TOTN+4thON	$-0.00502754 - 0.000159129 x_1 - 3.19154*10^{-6} x_1^2 - 5.34623*10^{-8} x_1^3 - 8.08115*10^{-10} x_1^4 + 0.274477 x_2 - 0.00343606 x_1 x_2 - 0.000035982 x_1^2 x_2 + 7.63648*10^{-7} x_1^3 x_2 + 0.00157064 x_2^2 - 8.40224*10^{-6} x_1 x_2^2 - 1.08032*10^{-7} x_1^2 x_2^2 - 6.46673*10^{-6} x_2^3 + 5.91744*10^{-8} x_1 x_2^3 + 3.78244*10^{-9} x_2^4 + 0.0453686 \cos[x_1] - 0.0310349 \cos[x_1]^2 + 0.0704816 \cos[x_1]^3 + 0.0462577 \cos[x_2] - 0.113194 \cos[x_1] \cos[x_2] + 0.48607 \cos[x_1]^2 \cos[x_2] + 0.219068 \cos[x_2]^2 - 0.351471 \cos[x_1] \cos[x_2]^2 - 0.588317 \cos[x_2]^3 + 0.0649675 \sin[x_1] + 0.0871189 \sin[x_1]^2 + 0.10166 \sin[x_1]^3 - 1.09513 \sin[x_2] + 0.746161 \sin[x_1] \sin[x_2] - 1.24828 \sin[x_1]^2 \sin[x_2] - 0.244143 \sin[x_2]^2 - 0.0256369 \sin[x_1] \sin[x_2]^2 + 2.10235 \sin[x_2]^3$
33-TOTNR+4thONR	$(1.00002 + 1.00073 x_1 + 1.02579 x_1^2 + 1.57645 x_1^3 - 0.225452 x_1^4 + 1.00097 x_2 + 1.03711 x_1 x_2 + 2.00781 x_1^2 x_2 + 7.46365 x_1^3 x_2 + 1.05822 x_2^2 + 2.41051 x_1 x_2^2 - 16.4264 x_1^2 x_2^2 + 4.89508 x_2^3 + 11.6423 x_1 x_2^3 + 2.60731 x_2^4 + 0.999833 \cos[x_1] + 0.999973 \cos[x_1]^2 + 0.99988 \cos[x_1]^3 + 1.00005 \cos[x_2] + 0.999989 \cos[x_1] \cos[x_2] + 1.00003 \cos[x_1]^2 \cos[x_2] + 0.999979 \cos[x_2]^2 + 0.999916 \cos[x_1] \cos[x_2]^2 + 1.00002$

	$\begin{aligned} & \text{Cos}[x_2]^3 + 1.0001 \text{Sin}[x_1] + 1.00004 \text{Sin}[x_1]^2 + 1.00005 \text{Sin}[x_1]^3 + 0.999976 \text{Sin}[x_2] + \\ & 1.99998 \text{Sin}[x_1] \text{Sin}[x_2] + 0.999993 \text{Sin}[x_1]^2 \text{Sin}[x_2] + 1.00004 \text{Sin}[x_2]^2 + 1.00006 \text{Sin}[x_1] \\ & \text{Sin}[x_2]^2 + 0.999977 \text{Sin}[x_2]^3 / (0.999961 + 1.00037 x_1 + 1.1448 x_1^2 + 11.1201 x_1^3 + \\ & 0.395741 x_1^4 + 0.998105 x_2 + 1.1027 x_1 x_2 + 12.136 x_1^2 x_2 - 1.43365 x_1^3 x_2 + 1.03723 \\ & x_2^2 + 18.0477 x_1 x_2^2 + 0.7269 x_1^2 x_2^2 + 48.1231 x_2^3 - 0.435331 x_1 x_2^3 + \\ & 0.00253352 x_2^4 + 1.00213 \text{Cos}[x_1] + 1.00035 \text{Cos}[x_1]^2 + 1.00162 \text{Cos}[x_1]^3 + 0.999785 \\ & \text{Cos}[x_2] + 1.00011 \text{Cos}[x_1] \text{Cos}[x_2] + 0.999845 \text{Cos}[x_1]^2 \text{Cos}[x_2] + 1.00017 \text{Cos}[x_2]^2 + \\ & 1.0011 \text{Cos}[x_1] \text{Cos}[x_2]^2 + 0.999949 \text{Cos}[x_2]^3 + 0.99891 \text{Sin}[x_1] + 0.999613 \text{Sin}[x_1]^2 + \\ & 0.999462 \text{Sin}[x_1]^3 + 1.00006 \text{Sin}[x_2] + 1.99991 \text{Sin}[x_1] \text{Sin}[x_2] + 0.999988 \text{Sin}[x_1]^2 \\ & \text{Sin}[x_2] + 0.99979 \text{Sin}[x_2]^2 + 0.999486 \text{Sin}[x_1] \text{Sin}[x_2]^2 + 1.00007 \text{Sin}[x_2]^3) \end{aligned}$
34-FOLNR+LR	$(1785.26 + 7.39057 x_1 + 1.06146 x_2 + 1.12395 \text{Log}[x_2] - 542.859 \text{Log}[x_1]) / (-94.0417 - 0.494544 x_1 + 0.000463077 x_2 + 0.0265098 \text{Log}[x_2] + 31.9305 \text{Log}[x_1])$
35-FOLN+SON	$130.151 - 2.81394 x_1 + 0.0194985 x_1^2 + 0.0341223 x_2 + 0.00172313 x_1 x_2 - 0.00001009 x_2^2 + 0.155487 \text{Log}[x_2] - 6.99938 \text{Log}[x_1]$
36-FOLNR+SONR	$(40238.8 - 1007.59 x_1 + 7.38412 x_1^2 + 2680.58 x_2 - 30.8179 x_1 x_2 + 0.433313 x_2^2 + 254.254 \text{Log}[x_2] - 1861.03 \text{Log}[x_1]) / (19347.4 - 247.081 x_1 + 0.306676 x_1^2 + 9.34185 x_2 - 0.0670296 x_1 x_2 - 0.00286666 x_2^2 + 6.73411 \text{Log}[x_2] - 81.3633 \text{Log}[x_1])$
37-FOLN+TON	$15.0865 - 0.272436 x_1 - 0.00292811 x_1^2 + 0.0000447138 x_1^3 + 0.575466 x_2 - 0.0119844 x_1 x_2 + 0.0000869997 x_1^2 x_2 - 0.000224381 x_2^2 + 2.02689 * 10^{-6} x_1 x_2^2 + 1.20084 * 10^{-7} x_2^3 + 0.143385 \text{Log}[x_2] + 0.499587 \text{Log}[x_1]$
38-FOLNR+TONR	$(-1.11246 * 10^7 - 56607.3 x_1 - 953.958 x_1^2 + 8.58859 x_1^3 - 588623. x_2 + 16447.4 x_1 x_2 - 111.739 x_1^2 x_2 - 311.782 x_2^2 + 3.08547 x_1 x_2^2 + 0.381756 x_2^3 + 1489.35 \text{Log}[x_2] + 3.95033 * 10^6 \text{Log}[x_1]) / (-1.56941 * 10^6 + 17951.8 x_1 + 373.52 x_1^2 - 4.54147 x_1^3 - 5547.42 x_2 + 105.187 x_1 x_2 - 0.511742 x_1^2 x_2 + 5.45908 x_2^2 - 0.0356526 x_1 x_2^2 - 0.000799761 x_2^3 + 74.197 \text{Log}[x_2] + 32758.4 \text{Log}[x_1])$
39-FOLN+4thON	$8.33142 - 0.0982929 x_1 - 0.00191392 x_1^2 - 9.14913 * 10^{-6} x_1^3 + 3.06749 * 10^{-7} x_1^4 + 0.244321 x_2 - 0.00246401 x_1 x_2 - 0.0000268445 x_1^2 x_2 + 6.01267 * 10^{-7} x_1^3 x_2 + 0.00106257 x_2^2 - 8.5028 * 10^{-6} x_1 x_2^2 - 8.14957 * 10^{-8} x_1^2 x_2^2 - 3.84613 * 10^{-6} x_2^3 + 5.04535 * 10^{-8} x_1 x_2^3 + 3.27408 * 10^{-10} x_2^4 + 0.118661 \text{Log}[x_2] + 0.75798 \text{Log}[x_1]$
40-FOLNR+4thONR	$(1.13784 * 10^8 - 321153. x_1 - 48257. x_1^2 - 455.056 x_1^3 + 7.92674 x_1^4 + 2.87557 * 10^7 x_2 - 230267. x_1 x_2 - 2426.2 x_1^2 x_2 + 13.5905 x_1^3 x_2 - 402983. x_2^2 + 4579.09 x_1 x_2^2 + 1.90454 x_1^2 x_2^2 + 1455.34 x_2^3 - 17.1217 x_1 x_2^3 - 5.54603 * 10^{-6} x_2^4 + 1.1182 \text{Log}[x_2] + 2.87782 * 10^7 \text{Log}[x_1]) / (1.83029 * 10^8 - 1.33666 * 10^6 x_1 - 29829.1 x_1^2 - 199.456 x_1^3 + 3.40003 x_1^4 - 4.06825 * 10^6 x_2 + 38308.9 x_1 x_2 + 385.723 x_1^2 x_2 - 3.21572 x_1^3 x_2 + 18940. x_2^2 - 308.611 x_1 x_2^2 + 1.00926 x_1^2 x_2^2 - 5.30365 x_2^3 + 0.062395 x_1 x_2^3 + 5.79103 * 10^{-8} x_2^4 - 0.268222 \text{Log}[x_2] + 2.05081 * 10^7 \text{Log}[x_1])$

Appendix C

Expanded Model Expressions of the Case Studies – Moisture Output

Models Name	Models
1-L	$0.12677 - 0.000559424 x_1 - 0.000328263 x_2$
2-LR	$(71.7185 - 0.592572 x_1 - 0.0720167 x_2)/(454.917 - 3.66481 x_1 + 4.01164 x_2)$
3-SON	$0.307125 - 0.00332097 x_1 + 0.0000121166 x_1^2 - 0.00161085 x_2 + 5.59495 \cdot 10^{-6} x_1 x_2 + 2.847 \cdot 10^{-6} x_2^2$
4-SONR	$(2.26714 \cdot 10^7 - 116308. x_1 + 979.434 x_1^2 - 8113.61 x_2 + 63.9685 x_1 x_2 + 28.2714 x_2^2 - 33799.5 \text{ Cos}[x_2] - 4.42467 \cdot 10^6 \text{ Log}[x_1]) / (3.09691 \cdot 10^7 - 806338. x_1 + 3837.83 x_1^2 - 46345.5 x_2 - 205.478 x_1 x_2 + 1919.68 x_2^2 - 520826. \text{ Cos}[x_2] + 2.58932 \cdot 10^6 \text{ Log}[x_1])$
5-TON	$0.679099 - 0.00676838 x_1 - 0.0000892269 x_1^2 + 1.09937 \cdot 10^{-6} x_1^3 - 0.00684891 x_2 + 0.000127187 x_1 x_2 - 8.24113 \cdot 10^{-7} x_1^2 x_2 + 9.2 \cdot 10^{-6} x_2^2 + 8.59612 \cdot 10^{-9} x_1 x_2^2 - 1.56017 \cdot 10^{-8} x_2^3$
6-TONR	$(124531. - 1956.79 x_1 - 16.3574 x_1^2 + 0.277255 x_1^3 - 3297.76 x_2 + 88.6517 x_1 x_2 - 0.615177 x_1^2 x_2 - 0.12879 x_2^2 + 0.0199973 x_1 x_2^2 - 0.00105445 x_2^3) / (556304. - 5427.63 x_1 - 165.262 x_1^2 + 1.90438 x_1^3 + 3474.86 x_2 - 65.3973 x_1 x_2 + 0.117678 x_1^2 x_2 - 121.708 x_2^2 + 1.64637 x_1 x_2^2 + 0.205334 x_2^3)$
7-4thON	$0.760415 - 0.00431708 x_1 - 0.000110394 x_1^2 - 5.95087 \cdot 10^{-7} x_1^3 + 1.76453 \cdot 10^{-8} x_1^4 - 0.0143282 x_2 + 0.000184772 x_1 x_2 + 2.15976 \cdot 10^{-6} x_1^2 x_2 - 3.21668 \cdot 10^{-8} x_1^3 x_2 + 0.0000677951 x_2^2 - 1.84496 \cdot 10^{-6} x_1 x_2^2 + 1.47453 \cdot 10^{-8} x_1^2 x_2^2 + 2.71487 \cdot 10^{-8} x_2^3 - 8.01694 \cdot 10^{-10} x_1 x_2^3 + 2.69031 \cdot 10^{-11} x_2^4$
8-4thONR	$(5.52397 \cdot 10^9 - 1.51 \cdot 10^8 x_1 + 2.50171 \cdot 10^6 x_1^2 - 44875.9 x_1^3 + 345.095 x_1^4 + 2.89132 \cdot 10^7 x_2 - 4.86034 \cdot 10^6 x_1 x_2 + 123551. x_1^2 x_2 - 879.199 x_1^3 x_2 + 720376. x_2^2 - 23704.7 x_1 x_2^2 + 220.121 x_1^2 x_2^2 + 102.463 x_2^3 - 12.2171 x_1 x_2^3 + 1.13047 x_2^4) / (3.02846 \cdot 10^{10} - 4.28402 \cdot 10^8 x_1 - 4.0617 \cdot 10^6 x_1^2 + 8246.74 x_1^3 + 740.579 x_1^4 - 8.41975 \cdot 10^7 x_2 + 6.70892 \cdot 10^6 x_1 x_2 - 99522.7 x_1^2 x_2 + 108.008 x_1^3 x_2 - 6.36674 \cdot 10^6 x_2^2 + 133686. x_1 x_2^2 - 430.01 x_1^2 x_2^2 - 33005.7 x_2^3 + 502.109 x_1 x_2^3 - 23.3397 x_2^4)$
9-5thON	$0.58222 - 0.00130373 x_1 - 0.0000690783 x_1^2 - 7.90644 \cdot 10^{-7} x_1^3 - 1.12456 \cdot 10^{-9} x_1^4 + 1.64032 \cdot 10^{-10} x_1^5 - 0.00983683 x_2 + 0.0000860669 x_1 x_2 + 1.79766 \cdot 10^{-6} x_1^2 x_2 + 7.36726 \cdot 10^{-9} x_1^3 x_2 - 3.40231 \cdot 10^{-10} x_1^4 x_2 + 0.0000212271 x_2^2 - 1.04889 \cdot 10^{-6} x_1 x_2^2 - 7.95732 \cdot 10^{-9} x_1^2 x_2^2 + 2.19444 \cdot 10^{-10} x_1^3 x_2^2 + 3.15304 \cdot 10^{-7} x_2^3 + 2.54196 \cdot 10^{-9} x_1 x_2^3 - 6.21507 \cdot 10^{-11} x_1^2 x_2^3 - 1.42256 \cdot 10^{-9} x_2^4 + 1.00192 \cdot 10^{-10} x_1 x_2^4$

	$11 x_1 x_2^4 + 9.27437 \cdot 10^{-13} x_2^5$
10-5thONR	$(-6.24355 \cdot 10^9 + 1.27225 \cdot 10^8 x_1 + 697026. x_1^2 - 18124.6 x_1^3 - 70.699 x_1^4 + 1.22298 x_1^5 + 4.69351 \cdot 10^7 x_2 - 622122. x_1 x_2 - 12712.5 x_1^2 x_2 + 120.977 x_1^3 x_2 + 0.288804 x_1^4 x_2 - 850275. x_2^2 + 14786.6 x_1 x_2^2 + 170.488 x_1^2 x_2^2 - 2.40229 x_1^3 x_2^2 + 14079.8 x_2^3 - 427.187 x_1 x_2^3 + 2.88268 x_1^2 x_2^3 + 2.60554 x_2^4 + 0.0555936 x_1 x_2^4 - 0.00989203 x_2^5)/(-1.58162 \cdot 10^{10} - 1.756 \cdot 10^8 x_1 + 1.97975 \cdot 10^7 x_1^2 - 311901. x_1^3 + 2185.11 x_1^4 - 7.52776 x_1^5 + 1.3918 \cdot 10^9 x_2 - 3.2763 \cdot 10^7 x_1 x_2 + 240127. x_1^2 x_2 - 3259.79 x_1^3 x_2 + 31.1884 x_1^4 x_2 - 4.5572 \cdot 10^7 x_2^2 + 717797. x_1 x_2^2 + 7280.82 x_1^2 x_2^2 - 111.017 x_1^3 x_2^2 + 1.09044 \cdot 10^6 x_2^3 - 31022.1 x_1 x_2^3 + 215.409 x_1^2 x_2^3 - 37.8209 x_2^4 + 2.0234 x_1 x_2^4 - 0.215956 x_2^5)$
11-FOTN	$0.0430751 + 0.000540109 x_1 - 0.000323641 x_2 + 0.00585602 \cos[x_1] + 0.00280079 \cos[x_2] + 0.0193855 \sin[x_1] + 0.00101956 \sin[x_2]$
12-FOTNR	$(734.927 - 6.06166 x_1 - 0.432269 x_2 + 9.69916 \cos[x_1] + 32.9546 \cos[x_2] + 51.4972 \sin[x_1] + 25.9792 \sin[x_2]) / (10480.5 - 121.492 x_1 + 64.8598 x_2 - 273.649 \cos[x_1] + 927.451 \cos[x_2] - 1148.84 \sin[x_1] + 813.999 \sin[x_2])$
13-SOTN	$0.00929899 + 0.00435651 \cos[x_1] + 0.0102278 \cos[x_1]^2 + 0.0131686 \cos[x_2] + 0.00295636 \cos[x_1] \cos[x_2] + 0.0325674 \cos[x_2]^2 + 0.00176438 \sin[x_1] + 0.0200505 \sin[x_1]^2 + 0.0122466 \sin[x_2] - 0.00189071 \sin[x_1] \sin[x_2] - 0.00855751 \sin[x_2]^2$
14-SOTNR	$(-258.448 + 21.5438 \cos[x_1] - 129.705 \cos[x_1]^2 + 19.7083 \cos[x_2] + 41.9817 \cos[x_1] \cos[x_2] + 414.254 \cos[x_2]^2 + 16.956 \sin[x_1] - 171.407 \sin[x_1]^2 - 1.1461 \sin[x_2] + 4.26493 \sin[x_1] \sin[x_2] + 414.151 \sin[x_2]^2) / (98.2437 + 491.617 \cos[x_1] + 172.357 \cos[x_1]^2 + 672.438 \cos[x_2] + 1355.71 \cos[x_1] \cos[x_2] + 153.666 \cos[x_2]^2 - 456.082 \sin[x_1] + 491.147 \sin[x_1]^2 - 42.9708 \sin[x_2] + 131.747 \sin[x_1] \sin[x_2] + 150.833 \sin[x_2]^2)$
15-TOTN	$0.00893416 + 0.00183501 \cos[x_1] + 0.0107546 \cos[x_1]^2 + 0.00312068 \cos[x_1]^3 - 0.00329505 \cos[x_2] + 0.0049223 \cos[x_1] \cos[x_2] - 0.00895885 \cos[x_1]^2 \cos[x_2] + 0.0314299 \cos[x_2]^2 + 0.0000174216 \cos[x_1] \cos[x_2]^2 + 0.0333626 \cos[x_2]^3 - 0.000357978 \sin[x_1] + 0.0156993 \sin[x_1]^2 + 0.0107879 \sin[x_1]^3 + 0.0392752 \sin[x_2] + 0.00734724 \sin[x_1] \sin[x_2] - 0.0145669 \sin[x_1]^2 \sin[x_2] - 0.00836922 \sin[x_2]^2 - 0.00510014 \sin[x_1] \sin[x_2]^2 - 0.0316106 \sin[x_2]^3$
16-TOTNR	$(0.343026 + 1.02264 \cos[x_1] - 0.420857 \cos[x_1]^2 - 0.998823 \cos[x_1]^3 + 0.118933 \cos[x_2] + 0.0713686 \cos[x_1] \cos[x_2] - 0.0532898 \cos[x_1]^2 \cos[x_2] + 0.168127 \cos[x_2]^2 - 0.00491018 \cos[x_1] \cos[x_2]^2 + 0.00859075 \cos[x_2]^3 + 0.297465 \sin[x_1] - 0.215802 \sin[x_1]^2 + 0.166643 \sin[x_1]^3 + 0.0390702 \sin[x_2] + 0.256846 \sin[x_1] \sin[x_2] + 0.160711 \sin[x_1]^2 \sin[x_2] + 0.232336 \sin[x_2]^2 + 0.253004 \sin[x_1] \sin[x_2]^2 + 0.00986226 \sin[x_2]^3) / (1.49086 + 1.02933 \cos[x_1] + 2.46228 \cos[x_1]^2 + 2.42122 \cos[x_1]^3 + 2.20341 \cos[x_2] + 2.2127 \cos[x_1] \cos[x_2] - 0.220898 \cos[x_1]^2 \cos[x_2] - 0.210736 \cos[x_2]^2 + 0.254794 \cos[x_1] \cos[x_2]^2 + 0.548699 \cos[x_2]^3 + 1.67476 \sin[x_1] - 3.96066 \sin[x_1]^2 + 6.48449 \sin[x_1]^3 + 1.21535 \sin[x_2] + 7.48491 \sin[x_1] \sin[x_2] + 0.426784 \sin[x_1]^2 \sin[x_2] + 1.60681 \sin[x_2]^2 + 6.81565 \sin[x_1] \sin[x_2]^2 + 0.568696 \sin[x_2]^3)$
17-FOLN	$0.164368 + 0.0115526 \log[x_2] - 0.0299538 \log[x_1]$
18-FOLNR	$(17.6768 + 75.113 \log[x_2] - 2.82453 \log[x_1]) / (450.346 + 2046.83 \log[x_2] - 67.5324 \log[x_1])$
19-SOLN	$-3.29681 + 0.157466 \log[x_2] + 0.0363563 \log[x_2]^2 + 1.57489 \log[x_1] - 0.0338237 \log[x_2] \log[x_1] - 0.186908 \log[x_1]^2$
20-SOLNR	$(2576.9 + 268.914 \log[x_2] - 665.114 \log[x_2]^2 - 1114.45 \log[x_1] + 113.225 \log[x_2])$

	$\text{Log}[x1] + 114.51 \text{Log}[x1]^2 / (31721.2 + 20897.9 \text{Log}[x2] - 22743.8 \text{Log}[x2]^2 - 15020.9 \text{Log}[x1] + 1219.65 \text{Log}[x2] \text{Log}[x1] + 1590.41 \text{Log}[x1]^2)$
21-SOTNR+LR	$(-18.7362 - 0.166425 x1 + 0.00901176 x2 - 47.8773 \text{Cos}[x1] - 222.426 \text{Cos}[x1]^2 - 2.42156 \text{Cos}[x2] - 0.563979 \text{Cos}[x1] \text{Cos}[x2] + 164.961 \text{Cos}[x2]^2 - 220.975 \text{Sin}[x1] + 148.977 \text{Sin}[x1]^2 - 1.68722 \text{Sin}[x2] - 0.419271 \text{Sin}[x1] \text{Sin}[x2] + 159.28 \text{Sin}[x2]^2) / (97.0003 - 0.264921 x1 + 2.68138 x2 - 57.261 \text{Cos}[x1] - 303.388 \text{Cos}[x1]^2 - 38.4144 \text{Cos}[x2] - 2.55032 \text{Cos}[x1] \text{Cos}[x2] + 173.642 \text{Cos}[x2]^2 - 300.764 \text{Sin}[x1] + 231.043 \text{Sin}[x1]^2 - 45.1304 \text{Sin}[x2] - 14.989 \text{Sin}[x1] \text{Sin}[x2] + 14.1368 \text{Sin}[x2]^2)$
22-TOTNR+LR	$(966.203 - 9.52491 x1 + 0.378491 x2 + 388.441 \text{Cos}[x1] - 572.666 \text{Cos}[x1]^3 + 330.523 \text{Cos}[x2] + 17.3647 \text{Cos}[x1]^2 \text{Cos}[x2] + 201.684 \text{Cos}[x1] \text{Cos}[x2]^2 - 296.705 \text{Cos}[x2]^3 + 1538.79 \text{Sin}[x1] - 2585.88 \text{Sin}[x1]^3 + 9.37116 \text{Sin}[x2] - 346.448 \text{Sin}[x1]^2 \text{Sin}[x2] + 260.358 \text{Sin}[x1] \text{Sin}[x2]^2 - 239.921 \text{Sin}[x2]^3) / (-1395.73 - 22.6374 x1 + 9.19494 x2 - 3840.71 \text{Cos}[x1] - 2870.19 \text{Cos}[x1]^3 + 9026.66 \text{Cos}[x2] + 1950.67 \text{Cos}[x1]^2 \text{Cos}[x2] + 6612.66 \text{Cos}[x1] \text{Cos}[x2]^2 - 8202.05 \text{Cos}[x2]^3 - 340.877 \text{Sin}[x1] + 4485.9 \text{Sin}[x1]^3 + 2325.99 \text{Sin}[x2] - 11430.1 \text{Sin}[x1]^2 \text{Sin}[x2] + 10368.7 \text{Sin}[x1] \text{Sin}[x2]^2 - 11448.1 \text{Sin}[x2]^3)$
23-SOLNR+LR	$(-8402.26 + 7.47437 x1 + 0.0653856 x2 - 3.84301 \text{Log}[x2] - 3.08775 \text{Log}[x2]^2 + 4034.54 \text{Log}[x1] + 1.14012 \text{Log}[x2] \text{Log}[x1] - 516.418 \text{Log}[x1]^2) / (21040.3 + 87.4853 x1 - 13.9647 x2 - 193.175 \text{Log}[x2] - 304.049 \text{Log}[x2]^2 - 7253.85 \text{Log}[x1] + 38.5813 \text{Log}[x2] \text{Log}[x1] + 193.99 \text{Log}[x1]^2)$
24-SOTN+TON	$0.0242585 + 0.000288828 x1 + 3.29086 \cdot 10^{-6} x1^2 + 3.60585 \cdot 10^{-8} x1^3 - 0.00612441 x2 + 0.000101852 x1 x2 - 6.36616 \cdot 10^{-7} x1^2 x2 + 0.0000104422 x2^2 - 4.06359 \cdot 10^{-10} x1 x2^2 - 1.65771 \cdot 10^{-8} x2^3 + 0.0084296 \text{Cos}[x1] + 0.0217793 \text{Cos}[x1]^2 - 0.00551218 \text{Cos}[x2] - 0.00356519 \text{Cos}[x1] \text{Cos}[x2] + 0.029913 \text{Cos}[x2]^2 + 0.024088 \text{Sin}[x1] + 0.0711347 \text{Sin}[x1]^2 - 0.00166752 \text{Sin}[x2] + 0.00509679 \text{Sin}[x1] \text{Sin}[x2] + 0.0355352 \text{Sin}[x2]^2$
25-SOTNR+TONR	$(-3.49694 \cdot 10^7 - 406961. x1 + 3750.82 x1^2 - 9.74601 x1^3 - 560711. x2 + 19842.6 x1 x2 - 170.967 x1^2 x2 + 128.316 x2^2 + 9.97774 x1 x2^2 + 0.00310777 x2^3 - 4.52764 \cdot 10^6 \text{Cos}[x1] + 5.03835 \cdot 10^7 \text{Cos}[x1]^2 - 9785.99 \text{Cos}[x2] + 41072.4 \text{Cos}[x1] \text{Cos}[x2] - 5.63978 \cdot 10^6 \text{Cos}[x2]^2 - 1.29297 \cdot 10^7 \text{Sin}[x1] + 6.23008 \cdot 10^7 \text{Sin}[x1]^2 - 416201. \text{Sin}[x2] - 1.29779 \cdot 10^6 \text{Sin}[x1] \text{Sin}[x2] - 4.81228 \cdot 10^6 \text{Sin}[x2]^2) / (9.05751 \cdot 10^6 + 212547. x1 - 3528.3 x1^2 - 35.6881 x1^3 + 153625. x2 - 14207.5 x1 x2 - 65.4825 x1^2 x2 + 9147.09 x2^2 + 14.98 x1 x2^2 + 205.039 x2^3 - 1.88911 \cdot 10^7 \text{Cos}[x1] + 2.05798 \cdot 10^7 \text{Cos}[x1]^2 - 1.70927 \cdot 10^6 \text{Cos}[x2] + 15579.3 \text{Cos}[x1] \text{Cos}[x2] + 5.53372 \cdot 10^6 \text{Cos}[x2]^2 - 1.03327 \cdot 10^7 \text{Sin}[x1] - 5.01664 \cdot 10^7 \text{Sin}[x1]^2 - 835075. \text{Sin}[x2] - 9.18073 \cdot 10^6 \text{Sin}[x1] \text{Sin}[x2] + 3.34294 \cdot 10^7 \text{Sin}[x2]^2)$
26-SOTN+4thON	$0.0216462 + 0.00026166 x1 + 3.04521 \cdot 10^{-6} x1^2 + 3.43308 \cdot 10^{-8} x1^3 + 3.77622 \cdot 10^{-10} x1^4 - 0.0142937 x2 + 0.000181692 x1 x2 + 2.13653 \cdot 10^{-6} x1^2 x2 - 3.1564 \cdot 10^{-8} x1^3 x2 + 0.0000702405 x2^2 - 1.89134 \cdot 10^{-6} x1 x2^2 + 1.49415 \cdot 10^{-8} x1^2 x2^2 + 2.70059 \cdot 10^{-8} x2^3 - 7.83563 \cdot 10^{-10} x1 x2^3 + 2.3605 \cdot 10^{-11} x2^4 - 0.0018741 \text{Cos}[x1] + 0.0193819 \text{Cos}[x1]^2 - 0.00514511 \text{Cos}[x2] - 0.00266761 \text{Cos}[x1] \text{Cos}[x2] + 0.0271939 \text{Cos}[x2]^2 + 0.0269715 \text{Sin}[x1] + 0.0636743 \text{Sin}[x1]^2 + 0.000148422 \text{Sin}[x2] + 0.00425488 \text{Sin}[x1] \text{Sin}[x2] + 0.0311808 \text{Sin}[x2]^2$
27-SOTNR+4thONR	$(-1.31971 \cdot 10^6 - 36865.8 x1 - 826.838 x1^2 - 15.674 x1^3 + 0.68446 x1^4 - 74180.8 x2 + 55.8769 x1 x2 - 3.43203 x1^2 x2 + 0.549029 x1^3 x2 + 10.0705 x2^2 + 4.49059 x1 x2^2 - 0.431765 x1^2 x2^2 + 10.2347 x2^3 + 0.0010583 x1 x2^3 - 0.00421194 x2^4 + 6.43971 \cdot 10^6 \text{Cos}[x1] - 1.4234 \cdot 10^7 \text{Cos}[x1]^2 - 137290. \text{Cos}[x2] - 186692. \text{Cos}[x1] \text{Cos}[x2] - 2.34878 \cdot 10^6 \text{Cos}[x2]^2 + 6.54593 \cdot 10^6 \text{Sin}[x1] + 5.78968 \cdot 10^6 \text{Sin}[x1]^2 + 122803. \text{Sin}[x2] - 7933.64 \text{Sin}[x1] \text{Sin}[x2] - 2.49102 \cdot 10^6 \text{Sin}[x2]^2) / (323246. - 25805. x1 - 1025.21 x1^2 - 27.6709 x1^3 + 0.295251 x1^4 - 929293. x2 - 3273.64 x1 x2 + 57.2448 x1^2 x2 + 1.95182 x1^3 x2 + 19620.2 x2^2 - 2.11836 x1 x2^2 - 0.138905 x1^2 x2^2 + 61.2094 x2^3 - 6.35111 x1 x2^3 + 2.32281 x2^4 + 9.20869 \cdot 10^6 \text{Cos}[x1] - 4.16803 \cdot 10^7 \text{Cos}[x1]^2 - 4.55693 \cdot 10^6 \text{Cos}[x2] - 3.87484 \cdot 10^6 \text{Cos}[x1] \text{Cos}[x2] + 1.12193 \cdot 10^7 \text{Cos}[x2]^2 + 9.98668 \cdot 10^6 \text{Sin}[x1] + 9.35622 \cdot 10^6 \text{Sin}[x1]^2 + 500475. \text{Sin}[x2] + 768386. \text{Sin}[x1] \text{Sin}[x2])$

	$\text{Sin}[x_2] + 1.7191 \cdot 10^7 \text{Sin}[x_2]^2$
28-SOTN+5thON	$0.0192247 + 0.000232698 x_1 + 2.7161 \cdot 10^{-6} x_1^2 + 3.07678 \cdot 10^{-8} x_1^3 + 3.40769 \cdot 10^{-10} x_1^4 + 3.71669 \cdot 10^{-12} x_1^5 - 0.00980464 x_2 + 0.0000847625 x_1 x_2 + 1.77887 \cdot 10^{-6} x_1^2 x_2 + 7.33517 \cdot 10^{-9} x_1^3 x_2 - 3.35805 \cdot 10^{-10} x_1^4 x_2 + 0.0000191929 x_2^2 - 1.01489 \cdot 10^{-6} x_1 x_2^2 - 7.58966 \cdot 10^{-9} x_1^2 x_2^2 + 2.13689 \cdot 10^{-10} x_1^3 x_2^2 + 3.4126 \cdot 10^{-7} x_2^3 + 1.90724 \cdot 10^{-9} x_1 x_2^3 - 5.81711 \cdot 10^{-11} x_1^2 x_2^3 - 1.43779 \cdot 10^{-9} x_2^4 + 1.01356 \cdot 10^{-11} x_1 x_2^4 + 9.38293 \cdot 10^{-13} x_2^5 - 0.00485126 \text{Cos}[x_1] + 0.0168508 \text{Cos}[x_1]^2 - 0.00410729 \text{Cos}[x_2] - 0.00216205 \text{Cos}[x_1] \text{Cos}[x_2] + 0.0265735 \text{Cos}[x_2]^2 + 0.0270676 \text{Sin}[x_1] + 0.0579445 \text{Sin}[x_1]^2 + 0.00134839 \text{Sin}[x_2] + 0.00433191 \text{Sin}[x_1] \text{Sin}[x_2] + 0.0251471 \text{Sin}[x_2]^2$
29-SOTNR+5thONR	$(-1.03785 \cdot 10^{11} + 9.2379 \cdot 10^8 x_1 - 1.95606 \cdot 10^6 x_1^2 - 25154.9 x_1^3 - 1836.71 x_1^4 - 26.0788 x_1^5 + 2.83175 \cdot 10^9 x_2 - 7.69131 \cdot 10^6 x_1 x_2 - 471580. x_1^2 x_2 + 3.31497 x_1^3 x_2 + 56.4355 x_1^4 x_2 + 2.76335 \cdot 10^6 x_2^2 + 242677. x_1 x_2^2 - 4100.68 x_1^2 x_2^2 + 68.3922 x_1^3 x_2^2 - 197728. x_2^3 + 1723.64 x_1 x_2^3 - 77.9789 x_1^2 x_2^3 + 24.5249 x_2^4 + 31.45 x_1 x_2^4 - 4.81977 x_2^5 - 3.34911 \cdot 10^{10} \text{Cos}[x_1] + 1.29286 \cdot 10^{11} \text{Cos}[x_1]^2 - 1.14047 \cdot 10^{10} \text{Cos}[x_2] - 4.19282 \cdot 10^{10} \text{Cos}[x_1] \text{Cos}[x_2] - 6.50039 \cdot 10^{10} \text{Cos}[x_2]^2 - 1.85813 \cdot 10^{11} \text{Sin}[x_1] + 2.7656 \cdot 10^{11} \text{Sin}[x_1]^2 + 4.56726 \cdot 10^9 \text{Sin}[x_2] - 9.45237 \cdot 10^9 \text{Sin}[x_1] \text{Sin}[x_2] - 4.61544 \cdot 10^{10} \text{Sin}[x_2]^2) / (-3.30084 \cdot 10^{10} - 9.16298 \cdot 10^8 x_1 - 6.37993 \cdot 10^6 x_1^2 - 132335. x_1^3 + 1586.72 x_1^4 - 32.657 x_1^5 - 2.95405 \cdot 10^8 x_2 + 4.19145 \cdot 10^7 x_1 x_2 + 321145. x_1^2 x_2 - 10631.4 x_1^3 x_2 + 69.9187 x_1^4 x_2 + 7.62392 \cdot 10^7 x_2^2 + 1.58839 \cdot 10^6 x_1 x_2^2 - 1953.08 x_1^2 x_2^2 + 157.939 x_1^3 x_2^2 + 486145. x_2^3 + 5244.76 x_1 x_2^3 + 56.8884 x_1^2 x_2^3 - 10781.7 x_2^4 - 146.16 x_1 x_2^4 - 73.3219 x_2^5 + 4.29823 \cdot 10^{11} \text{Cos}[x_1] - 1.04524 \cdot 10^{11} \text{Cos}[x_1]^2 - 8.38238 \cdot 10^{10} \text{Cos}[x_2] - 5.678 \cdot 10^{11} \text{Cos}[x_1] \text{Cos}[x_2] - 3.04898 \cdot 10^{11} \text{Cos}[x_2]^2 + 1.8199 \cdot 10^{10} \text{Sin}[x_1] - 1.16872 \cdot 10^{11} \text{Sin}[x_1]^2 + 6.66474 \cdot 10^{10} \text{Sin}[x_2] - 1.63002 \cdot 10^{11} \text{Sin}[x_1] \text{Sin}[x_2] - 5.33468 \cdot 10^{10} \text{Sin}[x_2]^2$
30-TOTN+TON	$0.0231692 + 0.000288138 x_1 + 3.44854 \cdot 10^{-6} x_1^2 + 3.99499 \cdot 10^{-8} x_1^3 - 0.0041154 x_2 + 0.0000462323 x_1 x_2 - 2.66919 \cdot 10^{-7} x_1^2 x_2 + 0.0000105009 x_2^2 + 7.85453 \cdot 10^{-10} x_1 x_2^2 - 1.64354 \cdot 10^{-8} x_2^3 + 0.0062655 \text{Cos}[x_1] + 0.0246745 \text{Cos}[x_1]^2 + 0.0121298 \text{Cos}[x_1]^3 + 0.0037673 \text{Cos}[x_2] - 0.00551644 \text{Cos}[x_1] \text{Cos}[x_2] - 0.0000432365 \text{Cos}[x_1]^2 \text{Cos}[x_2] + 0.027316 \text{Cos}[x_2]^2 - 0.00854449 \text{Cos}[x_1] \text{Cos}[x_2]^2 - 0.0145653 \text{Cos}[x_2]^3 + 0.00992135 \text{Sin}[x_1] + 0.053064 \text{Sin}[x_1]^2 + 0.0445347 \text{Sin}[x_1]^3 + 0.00407067 \text{Sin}[x_2] + 0.0236697 \text{Sin}[x_1] \text{Sin}[x_2] - 0.0343388 \text{Sin}[x_1]^2 \text{Sin}[x_2] + 0.0352573 \text{Sin}[x_2]^2 - 0.00297197 \text{Sin}[x_1] \text{Sin}[x_2]^2 + 0.00349978 \text{Sin}[x_2]^3$
31-TOTNR+TONR	$(-3.24149 \cdot 10^6 - 68116.4 x_1 - 712.916 x_1^2 - 1.61768 x_1^3 - 49929.1 x_2 + 1642.83 x_1 x_2 - 10.6507 x_1^2 x_2 - 100.745 x_2^2 - 0.680877 x_1 x_2^2 + 0.569521 x_2^3 + 7.34784 \cdot 10^6 \text{Cos}[x_1] + 2.33948 \cdot 10^7 \text{Cos}[x_1]^2 - 1.20664 \cdot 10^6 \text{Cos}[x_1]^3 + 82229.1 \text{Cos}[x_2] - 33225.2 \text{Cos}[x_1] \text{Cos}[x_2] - 18967.8 \text{Cos}[x_1]^2 \text{Cos}[x_2] + 6.11027 \cdot 10^6 \text{Cos}[x_2]^2 + 328062. \text{Cos}[x_1] \text{Cos}[x_2]^2 - 35600.6 \text{Cos}[x_2]^3 + 3.62624 \cdot 10^7 \text{Sin}[x_1] - 3.56356 \cdot 10^7 \text{Sin}[x_1]^2 - 6.90859 \cdot 10^6 \text{Sin}[x_1]^3 + 303112. \text{Sin}[x_2] + 182152. \text{Sin}[x_1] \text{Sin}[x_2] - 380116. \text{Sin}[x_1]^2 \text{Sin}[x_2] + 5.42983 \cdot 10^6 \text{Sin}[x_2]^2 - 101146. \text{Sin}[x_1] \text{Sin}[x_2]^2 - 357317. \text{Sin}[x_2]^3) / (2.51257 \cdot 10^7 + 76394.6 x_1 - 1696.78 x_1^2 - 3.86605 x_1^3 - 3.56524 \cdot 10^6 x_2 + 119213. x_1 x_2 - 791.76 x_1^2 x_2 - 8604.48 x_2^2 - 49.4245 x_1 x_2^2 + 48.4076 x_2^3 - 4.53858 \cdot 10^6 \text{Cos}[x_1] - 3.93759 \cdot 10^7 \text{Cos}[x_1]^2 - 9.6657 \cdot 10^6 \text{Cos}[x_1]^3 + 6.68298 \cdot 10^6 \text{Cos}[x_2] - 1.50593 \cdot 10^6 \text{Cos}[x_1] \text{Cos}[x_2] - 781698. \text{Cos}[x_1]^2 \text{Cos}[x_2] + 6.31412 \cdot 10^6 \text{Cos}[x_2]^2 + 6.63026 \cdot 10^6 \text{Cos}[x_1] \text{Cos}[x_2]^2 - 3.67269 \cdot 10^6 \text{Cos}[x_2]^3 - 5.8536 \cdot 10^6 \text{Sin}[x_1] + 4.00963 \cdot 10^7 \text{Sin}[x_1]^2 - 7.44009 \cdot 10^7 \text{Sin}[x_1]^3 + 2.94271 \cdot 10^7 \text{Sin}[x_2] + 8.28418 \cdot 10^6 \text{Sin}[x_1] \text{Sin}[x_2] - 1.30833 \cdot 10^7 \text{Sin}[x_1]^2 \text{Sin}[x_2] - 2.40233 \cdot 10^7 \text{Sin}[x_2]^2 + 601430. \text{Sin}[x_1] \text{Sin}[x_2]^2 - 3.84775 \cdot 10^7 \text{Sin}[x_2]^3)$
32-TOTN+4thON	$0.0209952 + 0.000262927 x_1 + 3.17656 \cdot 10^{-6} x_1^2 + 3.72425 \cdot 10^{-8} x_1^3 + 4.26648 \cdot 10^{-10} x_1^4 - 0.0116208 x_2 + 0.000140702 x_1 x_2 + 1.6647 \cdot 10^{-6} x_1^2 x_2 - 2.5136 \cdot 10^{-8} x_1^3 x_2 + 9.08167 \cdot 10^{-9} x_2^3 - 7.98576 \cdot 10^{-10} x_1 x_2^3 + 5.61846 \cdot 10^{-11} x_2^4 + 0.000386313 \text{Cos}[x_1] + 0.0221986 \text{Cos}[x_1]^2 + 0.00487097 \text{Cos}[x_1]^3 + 0.00847042 \text{Cos}[x_2] - 0.00448216 \text{Cos}[x_1] \text{Cos}[x_2] - 0.00119968 \text{Cos}[x_1]^2 \text{Cos}[x_2] + 0.0255634 \text{Cos}[x_2]^2 - 0.00408857 \text{Cos}[x_1] \text{Cos}[x_2]^2 - 0.0191194 \text{Cos}[x_2]^3 + 0.0125526 \text{Sin}[x_1] + 0.0487018 \text{Sin}[x_1]^2 + 0.0444176 \text{Sin}[x_1]^3 - 0.000441289 \text{Sin}[x_2] + 0.0150495 \text{Sin}[x_1] \text{Sin}[x_2] - 0.0183494 \text{Sin}[x_1]^2 \text{Sin}[x_2] + 0.031097 \text{Sin}[x_2]^2 - 0.00576173 \text{Sin}[x_1] \text{Sin}[x_2]^2 +$

	$0.00680698 \sin[x_2]^3$
33-TOTNR+4thONR	$(1.00032 + 1.01274 x_1 + 1.38719 x_1^2 + 6.40088 x_1^3 + 0.522508 x_1^4 + 1.00991 x_2 + 1.34335 x_1 x_2 + 8.34394 x_1^2 x_2 - 1.75491 x_1^3 x_2 + 1.27284 x_2^2 + 7.43434 x_1 x_2^2 + 1.40948 x_1^2 x_2^2 + 8.17719 x_2^3 - 0.655218 x_1 x_2^3 + 0.138076 x_2^4 + 0.995985 \cos[x_1] + 0.999352 \cos[x_1]^2 + 0.997066 \cos[x_1]^3 + 0.999435 \cos[x_2] + 1.00027 \cos[x_1] \cos[x_2] + 0.999701 \cos[x_1]^2 \cos[x_2] + 1.00033 \cos[x_2]^2 + 0.99737 \cos[x_1] \cos[x_2]^2 + 0.999562 \cos[x_2]^3 + 1.00235 \sin[x_1] + 1.00097 \sin[x_1]^2 + 1.00121 \sin[x_1]^3 + 1.00055 \sin[x_2] + 2.00626 \sin[x_1] \sin[x_2] + 1.00095 \sin[x_1]^2 \sin[x_2] + 0.999987 \sin[x_2]^2 + 1.00088 \sin[x_1] \sin[x_2]^2 + 1.00042 \sin[x_2]^3)/(0.999929 + 0.996453 x_1 + 0.844088 x_1^2 - 4.07737 x_1^3 + 4.36982 x_1^4 + 0.997255 x_2 + 0.873944 x_1 x_2 - 3.38736 x_1^2 x_2 - 8.82901 x_1^3 x_2 + 0.905885 x_2^2 - 2.19621 x_1 x_2^2 + 8.93325 x_1^2 x_2^2 - 1.72324 x_2^3 - 10.8257 x_1 x_2^3 + 8.25796 x_2^4 + 1.00031 \cos[x_1] + 1.00005 \cos[x_1]^2 + 1.0002 \cos[x_1]^3 + 1.00006 \cos[x_2] + 1.00002 \cos[x_1] \cos[x_2] + 1.00004 \cos[x_1]^2 \cos[x_2] + 0.999964 \cos[x_2]^2 + 1.00019 \cos[x_1] \cos[x_2]^2 + 1.00006 \cos[x_2]^3 + 0.999765 \sin[x_1] + 0.999881 \sin[x_1]^2 + 0.999871 \sin[x_1]^3 + 0.999882 \sin[x_2] + 1.99932 \sin[x_1] \sin[x_2] + 0.999884 \sin[x_1]^2 \sin[x_2] + 0.999965 \sin[x_2]^2 + 0.999905 \sin[x_1] \sin[x_2]^2 + 0.999917 \sin[x_2]^3)$
34-FOLNR+LR	$(211.339 + 0.503878 x_1 - 0.0194479 x_2 + 0.535497 \log[x_2] - 55.6631 \log[x_1])/(-4679.47 - 21.0836 x_1 + 1.69883 x_2 + 15.6621 \log[x_2] + 1464.27 \log[x_1])$
35-FOLN+SON	$0.301894 - 0.00405219 x_1 + 0.0000160018 x_1^2 - 0.00163856 x_2 + 5.6541 \cdot 10^{-6} x_1 x_2 + 2.90552 \cdot 10^{-6} x_2^2 - 0.00330706 \log[x_2] + 0.00916418 \log[x_1]$
36-FOLNR+SONR	$(2.26714 \cdot 10^7 - 116308. x_1 + 979.434 x_1^2 - 8113.61 x_2 + 63.9685 x_1 x_2 + 28.2714 x_2^2 - 33799.5 \log[x_2] - 4.42467 \cdot 10^6 \log[x_1])/(3.09691 \cdot 10^7 - 806338. x_1 + 3837.83 x_1^2 - 46345.5 x_2 - 205.478 x_1 x_2 + 1919.68 x_2^2 - 520826. \log[x_2] + 2.58932 \cdot 10^6 \log[x_1])$
37-FOLN+TON	$0.561752 - 0.00736437 x_1 - 0.000090157 x_1^2 + 1.10707 \cdot 10^{-6} x_1^3 - 0.00673525 x_2 + 0.000121724 x_1 x_2 - 7.79189 \cdot 10^{-7} x_1^2 x_2 + 9.68359 \cdot 10^{-6} x_2^2 + 4.68174 \cdot 10^{-9} x_1 x_2^2 - 1.58846 \cdot 10^{-8} x_2^3 - 0.00371593 \log[x_2] + 0.0385104 \log[x_1]$
38-FOLNR+TONR	$(151482. - 2339.14 x_1 - 12.7875 x_1^2 + 0.284316 x_1^3 - 3282.52 x_2 + 91.3465 x_1 x_2 - 0.662545 x_1^2 x_2 + 0.0527751 x_2^2 + 0.0231602 x_1 x_2^2 + 0.000976266 x_2^3 + 263.509 \log[x_2] - 4821.87 \log[x_1])/(668968. - 20305.4 x_1 + 17.4808 x_1^2 + 1.19429 x_1^3 + 11323.4 x_2 - 267.232 x_1 x_2 + 1.49467 x_1^2 x_2 - 95.6669 x_2^2 + 0.843192 x_1 x_2^2 + 0.565529 x_2^3 + 6312.67 \log[x_2] + 63917.2 \log[x_1])$
39-FOLN+4thON	$0.592372 - 0.00551175 x_1 - 0.000116077 x_1^2 - 5.88615 \cdot 10^{-7} x_1^3 + 1.81939 \cdot 10^{-8} x_1^4 - 0.0151752 x_2 + 0.000194591 x_1 x_2 + 2.28577 \cdot 10^{-6} x_1^2 x_2 - 3.36367 \cdot 10^{-8} x_1^3 x_2 + 0.0000745524 x_2^2 - 2.00149 \cdot 10^{-6} x_1 x_2^2 + 1.56284 \cdot 10^{-8} x_1^2 x_2^2 + 2.45576 \cdot 10^{-8} x_2^3 - 7.58509 \cdot 10^{-10} x_1 x_2^3 + 2.5168 \cdot 10^{-11} x_2^4 - 0.00400813 \log[x_2] + 0.0628062 \log[x_1]$
40-FOLNR+4thONR	$(9.63872 \cdot 10^6 - 76665.9 x_1 - 1696.74 x_1^2 - 35.3172 x_1^3 + 0.614054 x_1^4 - 403192. x_2 + 5413.89 x_1 x_2 + 95.7471 x_1^2 x_2 - 1.337 x_1^3 x_2 + 2097.12 x_2^2 - 66.9089 x_1 x_2^2 + 0.587559 x_1^2 x_2^2 + 0.022218 x_2^3 - 0.0203753 x_1 x_2^3 + 0.00153669 x_2^4 + 21242.7 \cos[x_2] + 360674. \log[x_1])/(2.91749 \cdot 10^6 - 755722. x_1 - 18905. x_1^2 - 44.633 x_1^3 + 2.87401 x_1^4 - 307113. x_2 + 8192.93 x_1 x_2 + 47.3433 x_1^2 x_2 - 1.73298 x_1^3 x_2 - 17417.2 x_2^2 + 371.967 x_1 x_2^2 - 1.72806 x_1^2 x_2^2 - 99.4177 x_2^3 + 1.90221 x_1 x_2^3 - 0.147357 x_2^4 + 276417. \cos[x_2] + 2.12239 \cdot 10^7 \log[x_1])$

Appendix D

Results of the Neuro-regression models boundedness

	ENERGY (kWh)		MOISTURE (%)	
	Min	Max	Min	Max
1-L	-3.09002	51.2355	-0.0192	0.0904
2-LR	-0.08642	53.5356	-0.0001	0.1532
3-SON	-1.3751	54.0085	-0.0007	0.1424
4-SONR	-0.5055	54.4104	0.0111	0.1473
5-TON	-0.7882	54.5859	0.0058	0.1640
6-TONR	-0.6734	55.1037	0.0088	0.1546
7-4thON	-0.6496	55.1424	-0.0042	0.1649
8-4thONR	0.1636	1.3*10 ¹³	0.0111	0.1506
9-5thON	-0.6505	54.3461	0.0053	0.1587
10-5thONR	-0.2997	55.0832	-5.1*10 ¹¹	5.7*10 ⁹
11-FOTN	-2.2222	51.7891	-0.0198	0.0938
12-FOTNR	-2.1*10 ¹⁶	40.1728	0.0015	0.4168
13-SOTN	12.8151	41.8999	-0.0064	0.0786
14-SOTNR	-2.4*10 ¹⁵	7.6*10 ¹²	-2.6*10 ¹⁰	4.4*10 ¹³
15-TOTN	13.1550	49.8808	-0.0024	0.0864
16-TOTNR	-2.9*10 ¹²	8.2*10 ¹⁵	-2.2*10 ⁹	2.8*10 ¹¹
17-FOLN	18.0854	30.4703	0.0197	0.0508
18-FOLNR	19.6278	26.7954	-2.6*10 ⁸	4.6*10 ¹⁰
19-SOLN	10.0695	35.0202	0.0105	0.0730

20-SOLNR	-3.0*10 ¹²	6.1*10 ¹³	-6.5*10 ¹⁰	0.0312
21-SOTNR+LR	-5.4*10 ¹⁴	3.2*10 ¹⁴	-2.2101	2.5*10 ¹²
22-TOTNR+LR	-1.4*10 ¹⁴	4.8*10 ¹⁴	-1.6*10 ¹²	5.0*10 ¹¹
23-SOLNR+LR	-0.3982	54.0859	0.0038	0.3769
24-SOTN+TON	-0.5119	54.3952	-0.0426	0.1030
25-SOTNR+TONR	-1.8*10 ¹⁵	1.4*10 ¹⁶	-3.9*10 ¹³	7.8*10 ¹¹
26-SOTN+4thON	0.1518	54.9578	-0.0293	0.1145
27-SOTNR+4thONR	-1.2*10 ¹⁵	55.4625	0.0083	3.3*10 ¹³
28-SOTN+5thON	-0.5750	54.3692	-0.0239	0.1642
29-SOTNR+5thONR	-1.1*10 ¹²	1.6*10 ¹⁵	-3.0*10 ¹¹	5.1*10 ¹¹
30-TOTN+TON	-0.9122	54.2703	-0.0952	0.1740
31-TOTNR+TONR	-5.6*10 ¹³	1.8855	-1.1*10 ¹⁵	1.0*10 ¹²
32-TOTN+4thON	-0.1967	55.6535	0.0115	0.1123
33-TOTNR+4thONR	-0.3925	53.6216	-0.0622	0.1442
34-FOLNR+LR	-0.3954	54.1126	0.0010	0.2137
35-FOLN+SON	-1.2837	54.0115	-0.0048	0.1410
36-FOLNR+SONR	-1.9140	55.0628	-2.6*10 ⁸	4.6*10 ¹⁰
37-FOLN+TON	-0.7415	54.6143	0.0083	0.1632
38-FOLNR+TONR	-0.6601	55.1064	0.0100	0.2248
39-FOLN+4thON	-0.5748	55.1514	-0.0070	0.1649
40-FOLNR+4thONR	-0.6628	3.6*10 ¹²	-13728.6	0.1707

Curriculum Vitae

Name Surname : Mustafa Dinç

Education:

2014 - 2020 B.Sc. / Izmir Katip Celebi University, Dept. of Mechanical Eng.

2020 - 2023 MSc. / Izmir Katip Celebi University, Dept. of Mechanical Eng.

Work Experience:

2021 – ERMETAL A.Ş – Design and Analysis Engineer - Bursa