KADIR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
DEPARTMENT OF MANAGEMENT INFORMATION SYSTEMS

SMART METHODS IN ELECTRICAL DISTRIBUTION SYSTEMS: MINIMIZATION OF VOLTAGE DEVIATIONS

İSMAİL ALPEREN ÖZLÜ
ASSIST. PROF. DR. OĞUZHAN CEYLAN

MASTER’S DEGREE THESIS

ISTANBUL, MAY, 2021

İsmail Alperen Özlü

Master's Degree Thesis

2021

SMART METHODS IN ELECTRICAL DISTRIBUTION SYSTEMS: MINIMIZATION OF VOLTAGE DEVIATIONS

İSMAİL ALPEREN ÖZLÜ
ASSIST. PROF. DR. OĞUZHAN CEYLAN

MASTER’S DEGREE THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES
WITH THE AIM TO MEET THE PARTIAL REQUIREMENTS REQUIRED TO RECEIVE A MASTER’S DEGREE IN THE DEPARTMENT OF MANAGEMENT INFORMATION SYSTEMS

İSTANBUL, MAY, 2021

NOTICE ON RESEARCH ETHICS AND
PUBLISHING METHODS

I, İSMAİL ALPEREN ÖZLÜ;

- hereby acknowledge, agree and undertake that this Master's Degree Thesis that I have prepared is entirely my own work and I have declared the citations from other studies in the bibliography in accordance with the rules;
- that this Master's Degree Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes of Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with the university legislation.

İSMAİL ALPEREN ÖZLÜ

____

DATE AND SIGNATURE

ACCEPTANCE AND APPROVAL

This study, titled **SMART METHODS IN ELECTRICAL DISTRIBUTION SYSTEMS: MINIMIZATION OF VOLTAGE DEVIATIONS**, prepared by the **İSMAİL ALPEREN ÖZLÜ**, was deemed successful with the **UNANIMOUS/MAJORITY VOTING** as a result of the thesis defense examination held on the _______________ and approved as a **MASTER'S DEGREE THESIS** by our jury.

JURY: SIGNATURE:

(Assist. Prof. Dr. Oğuzhan Ceylan) (Advisor) (Kadir Has University) _______________

(Prof. Dr. Hasan Dağ) (Jury Member) (Kadir Has University) _______________

(Assist. Prof. Dr. Şahin Aydın) (Jury Member) (Işık University) _______________

I confirm that the signatures above belong to the aforementioned faculty members.

____

(Title, Name and Surname)

Director of the School of Graduate Studies

APPROVAL DATE:

TABLE of CONTENTS

LIST of TABLES i

LIST of FIGURES ii

ABSTRACT v

ÖZET vi

ACKNOWLEDGEMENT vii

1. INTRODUCTION 1

2. POWER DISTRIBUTION SYSTEMS 4

2.1. Distribution Systems 4

2.2. Open Networks 5

2.3. Closed Networks 5

3. OPTIMIZATION 7

3.1. Meta-heuristic Optimization Algorithms 7

3.1.1. Genetic algorithm 8

3.1.2. Differential evolution algorithm 10

3.1.3. Particle swarm optimization algorithm 10

3.1.4. Grey wolf optimization 11

3.1.4.1. Mathematical model 11

3.1.4.2. Pseudocode of the algorithm 13

3.1.4.3. Algorithm steps and flowchart 13

3.1.5. Whale optimization algorithm 14

3.1.5.1. Mathematical model 14

3.1.5.2. Pseudocode of the algorithm 16

3.1.5.3. Algorithm steps and flowchart 17

3.1.5.4. Ant lion algorithm 19

3.1.5.5. Mathematical model 19

3.1.5.6. Pseudocode of the algorithm 21

3.1.5.7. Algorithm steps and flowchart 21

4. IMPLEMENTATION 23

4.1. Benchmark Functions 23
4.2. Distributed Energy Resources In Energy Systems 25
4.3. Use of MATPOWER Software 26
4.4. Data Collection 27
4.5. Minimize Objective Function with Random Walk Algorithm 30
4.6. Implementation of the Tap Changer on Solving Voltage Deviation Problem 31
4.7. Implementation of the Battery on Solving Voltage Deviation Problem 33
4.8. Visualization of the Test Systems 34
4.9. Application with Meta-Heuristic Algorithms 35
5. GRAPHICAL USER INTERFACE 36
5.1. Structure and General Information 36
5.2. Settings Panel 37
5.3. Preview Panel 39
5.4. Results Panel 39
5.5. Test Results 40
6. CONCLUSION 43
REFERENCES 45
CURRICULUM VITAE 51

LIST of TABLES

Table 4.1 Constant values of test functions. 24
Table 4.2 The result of calculating test functions with meta-heuristic algorithms. 24
Table 4.3 Hourly load values for test case. 27
Table 4.4 Hourly load values for PV. 28

LIST of FIGURES

Figure 2.1 Radial distribution network (*Elektrik Dağıtım Şebekeleri - İTÜ EMK*, 2015).5
Figure 2.2 Ring main distribution network (Kiran, 2018). 6
Figure 2.3 Interconnected system (*What is Electrical Grid? Definition & Types of an Interconnection - Circuit Globe*, 2018)....................6
Figure 3.1 Procedure of a GA based on (Höschel and Lakshminarayanan, 2019). 9
Figure 3.2 Pseudocode of the GWO algorithm based on (Mirjalili, 2014)....................13
Figure 3.3 Flowchart of the GWO algorithm based on (Mirjalili, 2014)....................13
Figure 3.4 Bubble-net feeding of humpback whales (Mirjalili and Lewis, 2016).........14
Figure 3.5 Spiral motion (Mirjalili and Lewis, 2016)....................15
Figure 3.6 Pseudocode of the WOA algorithm based on (Mirjalili and Lewis, 2016). ..17
Figure 3.7 Flowchart of the WOA algorithm based on (Soliman *et al.*, 2009)..............18
Figure 3.8 Cone-shaped traps (Mirjalili, 2015)....................19
Figure 3.9 Pseudocode of the ALO algorithm based on (Mirjalili, 2015). 21
Figure 3.10 Flowchart of the ALO algorithm based on (Pradhan *et al.*, 2019). 22
Figure 4.1 Example code for load case, update values and run case. 26
Figure 4.2 Example code for disabling print out of runpf method....................26
Figure 4.3 Scaled hourly load values. 28
Figure 4.4 Hourly PV active power outputs in kW....................29
Figure 4.5 Code fragment in which static values are defined. 30
Figure 4.6 The graph of the change of the pu values of the base case and optimized case 33 bus test system. 31
Figure 4.7 The graph of the change of the pu values of the base case and optimized case with tap changer at 33 bus test system. 32
Figure 4.8 A sample graph showing the hourly power change of two batteries. 34
Figure 4.9 Vector drawing of the 33 bus test system. 34
Figure 4.10 Flowchart of the custom software....................35
Figure 5.1 Placement of 3 panels on the application. 36
Figure 5.2 Settings panel of the software. 37
Figure 5.3 Screenshot of the preview panel. 39
Figure 5.4 Test results for GWO. 40
Figure 5.5 Test results for WOA. 40
Figure 5.6 Test results for ALO. 41
Figure 5.7 Comparison of the 24-hour voltage magnitude of 33 bus systems between base case (a) and optimized case (b). 41
Figure 5.8 Change of battery power state for 24 hours. 42
Figure 5.9 Tap changer values during 24 hours simulation. 42

LIST of ABBREVIATIONS

ALO	Ant Lion Optimization
AVG	Average
BESS	Battery Energy Storage System
CSV	Comma-separated values
DE	Differential Evolution
DG	Distribution Generation
ELD	Economic Load Dispatching
EV	Electic Vehicle
FBS	Forward-Backward Sweep
GA	Genetic Algorithm
GUI	Graphical User Interface
GWO	Grey Wolf Optimization
HTML	Hypertext Markup Language
kVA	Kilovolt-ampere
kW	Kilowatt
kWh	Kilowatt-hour
MW	Megawatt
OPF	Optimal Power Flow
PSO	Particle Swarm Optimization
PU	Per-unit
PV	Photovoltaic
SoC	State of Charge

STD	Standard Deviation
SVG	Scalable Vector Graphics
TXT	TeXT
WOA	Whale Optimization Algorithm

SMART METHODS IN ELECTRICAL DISTRIBUTION SYSTEMS:
MINIMIZATION OF VOLTAGE DEVIATIONS

ABSTRACT

The concept of smart distribution systems has emerged to make the current electricity distribution systems more efficient in line with the increasing energy need.

In this thesis, studies that try to minimize the voltage deviation in electrical distribution systems using heuristic algorithms are examined. Load flow tests were run in the 33 bus, 69 bus, and 141 bus test systems using the Grey Wolf Optimization algorithm (GWO), Whale Optimization algorithm (WOA), and Ant Lion Optimization (ALO) algorithms, which are also used in the literature, and improvements in voltage values were observed.

The aim of this thesis is to develop an open-source software tool that uses meta-heuristic algorithms for the problem of minimizing voltage deviations in electrical distribution systems. The software tool user has two options to minimize voltage deviations, the first of which is distributed generation sources and tap changers, and the second is batteries and tap changers. Before optimization, the user can choose one of the 33 bus, 69 bus, and 141 bus test systems and one of the GWO, WOA, and ALO algorithms. The user will be able to load his/her own load profile. It can adjust the number of DGs, batteries, and tap changers in the system through the program. The number of iterations and run the optimization takes can also be adjusted. As a result of the optimization, the user is shown graphs of the change between the base case and the optimized case, the 24-hour power state change in the batteries, the 24-hour tap changer values change.

Keywords: optimization, electrical distribution systems, meta-heuristic algorithms

ELEKTRİK DAĞITIM SİSTEMLERİNDE AKILLI YÖNTEMLER: GERİLİM SAPMALARININ EN KÜÇÜKLEŞTİRİLMESİ

ÖZET

Mevcut elektrik dağıtım sistemlerini, günümüzde artan enerji ihtiyacı doğrultusunda daha verimli hale getirmek amacıyla akıllı dağıtım sistemleri kavramı ortaya çıkmıştır.

Bu tezde, elektrik dağıtım sistemlerinde voltage sapmasını sezgisel algoritmalar kullanarak en küçükleştirmeye çalışan çalışmalar incelenmiştir. Literatürde de kullanılan, Gri Kurt Optimizasyon algoritması (GWO), Balina Optimizasyon algoritması (WOA) ve Karınca Aslanı Optimizasyon (ALO) algoritmaları kullanılarak yük akış testleri 33 baralı, 69 baralı ve 141 baralı test sistemlerinde çalıştırılmış ve gerilim değerlerinde iyileşmeler görülmüştür.

Bu tezin amacı, elektrik dağıtım sistemlerindeki gerilim sapmalarını en aza indirme problemi için meta-sezgisel algoritmalar kullanan açık kaynaklı bir yazılım aracı geliştirmektir. Yazılımın kullanıcısı, gerilim sapmalarını en küçükleştirmek için iki seçeneğe sahiptir, bunlardan ilkinde dağıtık üretim kaynakları ve kademe değiştiriciler kullanılır, ikincisinde ise bataryalar ve kademe değiştiriciler kullanılır. Kullanıcı optimizasyon öncesinde 33 bus, 69 bus ve 141 bus test sistemlerinden birini ve GWO, WOA ve ALO algoritmalarından birini seçebilir. Kullanıcı kendi yük profilini yükleyebilir. Sistemdeki dağıtık üretim kaynaklarının, bataryaların ve kademe değiştiricilerin adetleri program üzerinden ayarlanabilir. Optimizasyonun kaç iterasyon ve tur süreceği de ayarlanabilmektedir. Optimizasyon sonucunda kullanıcıya ana durum ve optimize edilmiş durum arasındaki değişim, bataryalardaki 24 saatlik güç durum değişimi, 24 saatlik kademe değiştirici değerleri değişimi grafikleri gösterilir.

Anahtar Sözcükler: en küçükleştirme, elektrik dağıtım sistemleri, meta-sezgisel algoritmalar

ACKNOWLEDGEMENT

I would like to express my deep thanks to my thesis advisor, Assist. Prof. Dr. Oğuzhan Ceylan. This thesis was formed thanks to his valuable knowledge, guidance, and motivation.

I would like to thank Prof. Dr. Hasan Dağ, who helped me in all circumstances and kept his doors open for all kinds of problems.

I would like to thank Aykut Çayır, who guided and encouraged me in every subject during my master's degree. I am grateful to İlyas Osmanoğulları for his help in this process.

Last, but not least, I would like to thank my family members, close friends, and colleagues who were with me in this process.

1. INTRODUCTION

Global energy consumption has increased rapidly with the increasing population and increasing welfare in the world (Terreson *et al.*, 2020). For the first time in human history since 2007, more than half of the population lived in cities (Madlener and Sunak, 2011). As poor households rose to the middle class with the developing world, they began to buy new assets that use significant amounts of energy (Wolfram, Shelef and Gertler, 2012). Thanks to urbanization, increasing living standards, extending life expectancy, industrialization, and technological developments, the energy needs of consumers, have increased significantly.

In power systems, electrical energy is generated in power plants, and then it is transmitted to the city centers, and lastly, to the end-users (customers). It is known that with increasing quality of life and industrialization, there are changes in the amount of energy that consumers need at different hours of a single day. According to this need, the load in the system may change from time to time. This varying load demand must be met by a properly designed power system. To meet the increasing need, equipment such as tap changer transformers, batteries, and distributed generation sources are installed in the networks to improve the voltage level, thus supplying the system requirement in case of overload.

By performing load flow analysis, all the bus voltages and thus, the bus with the highest voltage drop in a distribution system can be determined. Some installation can be made to the distribution system in order to maintain this voltage level along the distribution line to the required values. This installation can be a distributed generation source (such as renewable energy), battery, electric vehicle (EV), or a tap changer transformer. The quality of energy offered to customers should be improved by reducing lost power values in distribution systems. For example, at a bus where wind energy is used, changes in wind speed during the day can suddenly change the voltage values. This change will also affect the quality of energy the consumer has. Minimization of these voltage deviations in distribution networks is an optimization problem.

In the literature, numerical and heuristic optimization algorithms are used to solve optimization problems. Numerical algorithms guarantee us optimism, while heuristic algorithms guarantee the closest to the optimal result. However, as the optimization problem's size increases, numerical algorithms become insufficient. Heuristic algorithms are more successful than numerical algorithms in solving complex problems (Rodríguez *et al.*, 2018).

Our primary goal in this study is to optimize the voltage deviations in electrical distribution systems using meta-heuristic optimization algorithms. We used Grey Wolf Optimization algorithm (GWO), Whale Optimization algorithm (WOA), and Ant Lion Optimization (ALO) algorithm, and proposed an educational free interface to solve optimization problems in distribution systems. In the software we developed, we run load flow analysis using GWO, WOA, and ALO algorithms on 33 bus (Baran and Wu, 1989a), 69 bus (Baran and Wu, 1989b), and 141 (Khodr *et al.*, 2008) bus systems.

There are many studies in the literature that solve optimization problems using meta-heuristic algorithms. In 1962, the optimal power flow (OPF) was introduced by Carpainter (Carpentier, 1962). OPF often represents the problem of determining the most optimal operating levels in power systems to reduce operating costs and meet demand throughout the distribution system (Bukhsh *et al.*, 2013). In 2004, the optimal power flow solution about the economic load dispatching (ELD) was made by Osman et al. using a genetic algorithm (Osman, Abo-Sinna and Mousa, 2004). Various researches have been done using artificial bee colony algorithm (Ayan and Kiliç, 2012), ant-lion optimizer (Trivedi, Jangir and Parmar, 2016), grey wolf algorithm (Ladumor *et al.*, 2017), and whale optimization algorithm (Bentouati, Chaib and Chettih, 2017) for optimal power flow solution. In 2019, studies on the optimal placement of renewable energy sources and sizing were carried out using the grey wolf algorithm (Ahmadi, Ceylan and Ozdemir, 2019).

Various studies have also been carried out to evaluate the effects of adding a tap changer to the distribution system. In these studies, it has been observed that the addition of a tap changer for voltage profile adjustment has a positive effect on achieving the desired voltage values. (Gao and Redfern, 2010; Liu *et al.*, 2012; Daylak, 2016; Sarimuthu *et al.*, 2016; Ceylan, Liu and Tomsovic, 2018).

There are different studies on the use of batteries in distributed electricity distribution systems. Some of these focus on optimizing the location of energy systems (Awad, El-Fouly and Salama, 2015; Zhang *et al.*, 2016), others on optimizing the charging and discharge times of batteries (Ahmadi, Ceylan and Ozdemir, 2020). Apart from these, there are also studies focusing on BESS capacity optimization and operating costs (Bahramirad, Reder and Khodaei, 2012).

In this study, two different methods are applied to minimize the voltage deviations. In the first method, distributed generation (DG) sources and tap changers were installed to the system, and in the second method, battery and tap changers were installed to the system. While the first method is run on an hourly basis, the second method is operated on a 24-hourly (daily) basis. Photovoltaics were used as DG in the first method. In both methods, 33 bus, 69 bus, and 141 bus test systems are used. The minimization process was performed using meta-heuristic algorithms such as GWO, WOA, and ALO, which are widely used in the literature. Thanks to the developed software tool, according to the chosen method, the optimization result, the power state changes of the 24-hour batteries, the 24-hour tap changer values, and the 24-hour voltage magnitude graphs of the optimized system with battery and voltage regulators are drawn.

2. POWER DISTRIBUTION SYSTEMS

Electrical power systems are composed of several parts. They consist of large power plants which produce electricity, transmission lines for carrying the generated electricity and lastly, distribution systems deliver produced electricity to the consumer. Large power plants are established in areas where products used as primary energy sources are concentrated. The electricity produced is transmitted to the consumers with high voltage transmission lines to reduce losses. Today, the use of electronic devices has increased considerably for personal and commercial use. With the increasing population and the number of devices consuming electricity, the energy need has increased. In the previous years, electricity distribution was done only next to power generation networks. It is costly to transmit energy over long distances (Daylak, 2016).

2.1. Distribution Systems

The distribution network must provide the same electrical energy to consumers in residential units along the line. The consumer can be at the beginning, middle, or end of the line. The position of the consumer should not affect energy efficiency. The high or low voltage sent to the consumer can reduce the efficiency of electrical devices. Various network systems have been developed to meet these conditions. Distribution networks are examined in two groups as open networks and closed networks (Atalay, no date).

The most suitable and used networks according to the distribution types are generally as follows (Prakash *et al.*, 2016).

- Radial distribution networks
- Ring main distribution networks
- Interconnected distribution networks

2.2. Open Networks

It is called the type of network whose distribution shape resembles tree branches. It is generally preferred in city centers, villages, or industrial areas, where there is a single source of energy supply. Radial networks are also open networks. In a radial grid, there are a group of consumers for one energy source. Central points to be distributed are determined. Transformers are placed at these points. There are consumers around the transformer such as lamps, houses, and workplaces to be transmitted. In the first diagram (Figure 2.1), the line near the transformer appears to be thicker, these are the main lines. Lines to other consumers are thinner. These thin lines are called branch lines (Atalay, no date).

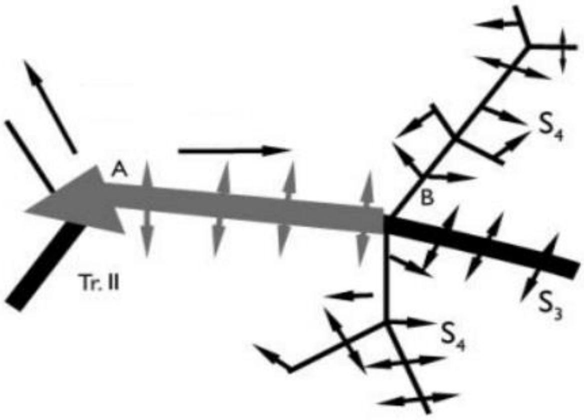


Figure 2.1 Radial distribution network (*Elektrik Dağıtım Şebekeleri* - *İTÜ EMK*, 2015).

Branched networks can be preferred because of their low cost, easy maintenance, and operation.

2.3. Closed Networks

Ring networks (Figure 2.2) can be given as examples of closed networks. Ring networks, the system feeding is carried out by a parallel connection of more than one transformer. Since the supply is made with more than one transformer in ring networks, in case of a fault in the ring, only the part with the fault will be disabled (Atalay, no date).

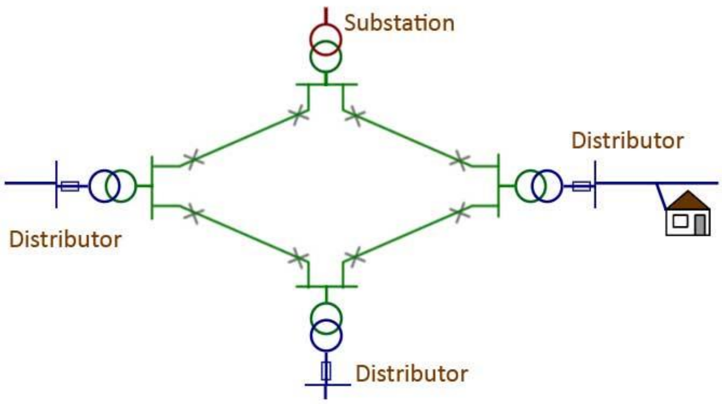


Figure 2.2 Ring main distribution network (Kiran, 2018).

Ring networks are more expensive than radial networks. However, the resulting system reliability and continuity are better.

Transmission between power generation plants and consumption centers, which are generally at distances from each other, is provided by interconnected networks (Figure 2.3). When a fault occurs in the interconnected system, only the defective part is deactivated, and the continuity of the energy flow is ensured.

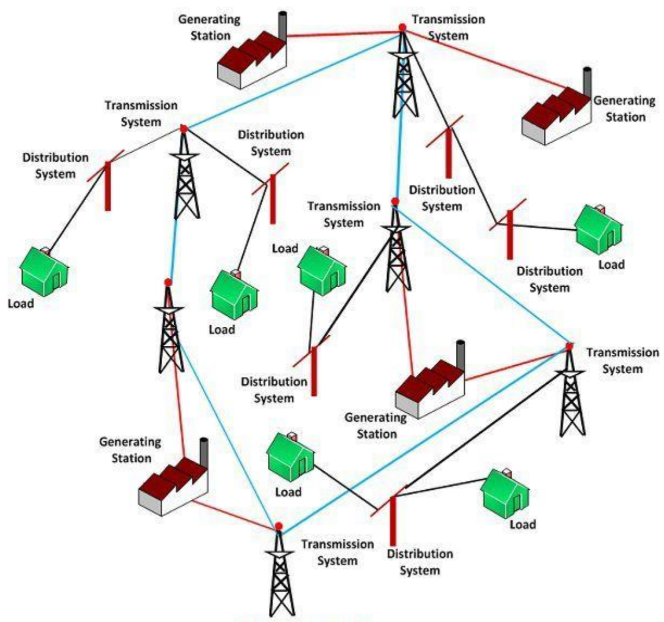


Figure 2.3 Interconnected system (*What is Electrical Grid? Definition & Types of an Interconnection - Circuit Globe*, 2018).

3. OPTIMIZATION

Optimization is a method that enables one to reach certain goals by using the resources in a system in the most efficient way. These goals can be cost minimization, profit maximization, capacity utilization maximization, and efficiency maximization. Optimization consists of two parts as modeling and analysis.

Mathematical optimization techniques have been used in many engineering fields for years. Mathematical algorithms are used in the solution of planning, operation, and control problems in energy systems. Energy systems are large, complex and geographically spread over a wide area. Due to this complex structure, it is difficult to make an optimization with mathematical assumptions.

Solution methods of optimization problems can be examined in two branches as analytical methods and numerical methods. Numerical methods are also divided into two as derivative-based methods and non-derivative-based methods (Antoniou and Lu, 2007).

In this study, non-derivative-based heuristic algorithms will be used. Some of these heuristic algorithms are genetic algorithm (GA), particle swarm optimization (PSO), differential evolution algorithm (DE) (Beheshti and Shamsuddin, 2013).

3.1. Meta-heuristic Optimization Algorithms

In the past years, many optimization algorithms such as exact and approximate algorithms have been proposed to solve optimization problems. These algorithms have good performance in many problems, but they are not efficient in solving large-scale optimization problems. The increase in the search area in optimization problems has also made these algorithms inefficient. Therefore, more flexible algorithms inspired by nature have begun to be proposed in the literature to overcome these limitations (Beheshti and Shamsuddin, 2013).

Meta-heuristic methods are problem-solving techniques that do not care if the result is provable or not. Meta-heuristic algorithms do not guarantee the best results, but they do guarantee that they will deliver a solution within the most reasonable time. They usually find the closest solution to the best, quickly and easily (Beheshti and Shamsuddin, 2013).

Meta-heuristic algorithms are becoming increasingly popular. Because heuristic algorithms can be used in different business lines belonging to many different disciplines, are based on very simple concepts, and finally can bypass local optimism (Mirjalili and Lewis, 2016).

3.1.1. Genetic algorithm

John Holland used the term genetic algorithm in 1975 for the first time. Some evolutionary ideas were put forward by Ingo Rechenberg and Hans-Paul Schwefel in Germany in the 1960s. All these ideas were familiar to the concepts of mutation and selection in Darwin's theory of evolution. However, these techniques could not be used effectively until the 80s due to the lack of sufficient computing power (Goldberg, 1994).

Genetic Algorithms (GAs) were developed as random search algorithms to mimic natural selection and genetic mechanics. GAs work on sequence structures, such as biological structures, that evolve over time according to the survival rule of the fittest, using a randomized but structured exchange of information. In this way, a new sequence is created in each generation using parts from the most suitable members of the old set (Roetzel, Luo and Chen, 2020). In general, a simple genetic algorithm process is selection, crossover, and mutation. Instead of producing a single solution to the problem, GA creates a solution set that includes the solution. Thus, more than one point in the search space can be evaluated at the same time, and the probability of reaching the best solution increases (Beasley, Bull and Martin, 1993). Solutions that contain the best results in the solution set are independent of each other. The processes of the genetic algorithm are shown in Figure 3.1.

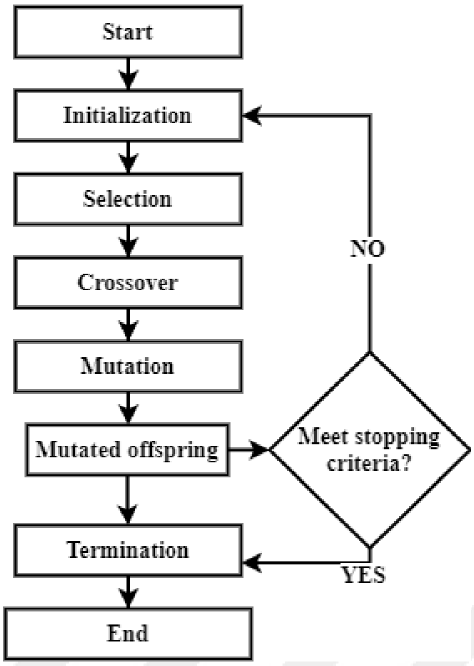


Figure 3.1 Procedure of a GA based on (Höschel and Lakshminarayanan, 2019).

GA's parameters can be thought of as genes in biology. These parameter sets produce the chromosome. Every potential solution is used as a chromosome in GA. The population is the name given to the solution set in which chromosomes are created. Under pre-determined rules, the suitability of the population is minimized or maximized.

The first step in developing a genetic algorithm for solving a problem, it is the representation of all solutions in the form of a string of bits with the same dimensions. Each series represents a random point in the space of possible solutions to the problem (Yeniay, 2001). A solution group is created in which possible solutions are coded. The solution group is called the population, and the codes of the solutions are also called chromosomes. The first step after a belt has been created is the step of calculating the fitness value of each member in the population (Taşkın and Emel, 2002). For example, for a maximization problem, the fitness value of the i-th member $f(i)$ is usually the value of the objective function at that point (Jyh-Shing Roger Jang, 1977). Selection, crossover, and mutation processes continue until the stop criteria are met. If desired, the maximum number of iterations can be specified and the loop can be stopped when the algorithm reaches this number. The stopping criterion can also be a fit-to-target value (Fung, Tang and Wang, 2002).

The working steps of the GA are as follows:

1. A set of solutions from all possible solutions in the search space is encoded as a string. Usually, a random process is done, and the starting population is formed.
2. The fit value is calculated for each sequence. The fit values were found to indicate the solution quality of the arrays.
3. A set of sequences is randomly selected based on a particular probability value. Selected sequences are subjected to crossover and mutation.
4. The new population formed is replaced by the old population.
5. The previous steps are repeated until the stopping criteria are met. The most suitable array is chosen as a solution.

3.1.2. Differential evolution algorithm

Differential evolution algorithm is a population-based heuristic method developed by Storn and Price in 1995 to solve optimization problems (Storn and Price, 1995). In the algorithm, many calculations are made at the same time. During the iterations, it is aimed to obtain better results for the solution of the problem with the help of operators.

Storn and Price have tried to improve the solution performance of problems using coding with real values with some changes in genetic operators. Selection, crossover, and mutation used in GA are also used in DE. But this application is not done sequentially. It is made on individual chromosomes. Using chromosomes, new individuals are obtained, and these are passed on to the next population (Keskintürk, 2006). An important advantage of DE to other heuristics is that it can be easily coded. While there are codes consisting of thousands of lines for other algorithms, about 20 lines of code are sufficient for DE (Mayer, Kinghorn and Archer, 2005).

3.1.3. Particle swarm optimization algorithm

Particle Swarm Optimization is an optimization method inspired by fish and insects moving as swarms (Kennedy and Eberhart, 1995). When the swarm of fish and birds looking for food were examined, it was seen that these animals interacted with each other. It has been noticed that if an animal in the herd finds food, the others turn their positions in the direction of the food without breaking away from the swarm. Although PSO is an evolutionary algorithm, it is simpler as it does not have operators such as crossover and mutation. It is also faster to converge to the optimum solution. Each bird that makes up the swarm is called a particle. In PSO, every particle is a candidate solution (Erdoğmuş and Yalçın, 2015).

3.1.4. Grey wolf optimization

The grey wolf optimization algorithm (GWO) is a nature-inspired optimization algorithm that mimics the strategy and leadership that grey wolves use when hunting. GWO was developed by Mirjalili in 2014. Grey wolves prefer to live in groups ranging in size from 5 to 12 wolves. Grey wolves are hierarchically divided into four groups as alpha, beta, delta, and omega wolves. Alpha wolves lead the pack. Like all living things, wolves need nourishment. Since they move in groups, they have developed a hunting technique. In the province, the place of prey is found and surrounded by the alpha wolf leadership. The algorithm aims to reach the optimal result using alpha, beta, and delta wolves. Omega wolf is not used in the GWO algorithm (Mirjalili, 2014).

3.1.4.1. Mathematical model

In the GWO algorithm, alpha wolves represent the best solution. Beta and delta wolves represent the second and third best solutions, respectively. Finally, omega wolves also represent candidate solutions (Mirjalili, 2014).

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{3.1}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D} \tag{3.2}$$

Equations 3.1 and 3.2 are the equations used to express the enclosure of the prey. t holds the current number of iterations, $\overrightarrow{X_p}$ the position of the prey, $\vec{X}$ the position vector of a grey wolf. $\vec{A}$ and $\vec{C}$ express the vector coefficients. These values are calculated as shown in equation 3.3 and equation 3.4 (Mirjalili, 2014).

$$\vec{A} = \vec{a} * (2 \cdot \overrightarrow{r_1} - 1) \tag{3.3}$$

$$\vec{C} = 2 \cdot \overrightarrow{r_2} \tag{3.4}$$

In the equations, $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ [0,1] refers to the random number, and $\vec{a}$ refers to the coefficient that decreases linearly as the iteration progresses from 2 to 0 (Mirjalili, 2014).

In GWO, the search process starts randomly, and then the fitness value of each wolf is calculated according to the cost function. The three most available positions are stored as alpha, beta, and delta. The alpha wolf manages the hunting process, but if necessary, beta and delta wolves can also participate. The position calculations of alpha, beta, and delta wolves are given in Equations 3.5a, 3.5b, and 3.5c (Mirjalili, 2014).

$$\overrightarrow{D_\alpha} = \left|\vec{C}_1 * \overrightarrow{X_\alpha} - \overrightarrow{X(t)}\right| \tag{3.5a}$$

$$\overrightarrow{D_\beta} = \left|\vec{C}_2 * \overrightarrow{X_\beta} - \overrightarrow{X(t)}\right| \tag{3.5b}$$

$$\overrightarrow{D_\delta} = \left|\vec{C}_3 * \overrightarrow{X_\delta} - \overrightarrow{X(t)}\right| \tag{3.5c}$$

$\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, and $\overrightarrow{X_\delta}$ represent the positions of alpha, beta, and delta wolves, respectively.

$$\overrightarrow{X_1} = \left|\overrightarrow{X_\alpha} - \overrightarrow{a_1}\overrightarrow{D_\alpha}\right| \tag{3.6a}$$

$$\overrightarrow{X_2} = \left|\overrightarrow{X_\beta} - \overrightarrow{a_2}\overrightarrow{D_\beta}\right| \tag{3.6b}$$

$$\overrightarrow{X_3} = \left|\overrightarrow{X_\delta} - \overrightarrow{a_3}\overrightarrow{D_\delta}\right| \tag{3.6c}$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{3.7}$$

Equation 3.7 represents the new position of the prey. Grey wolves attack the prey after the prey gets tired and stops its movement. The attack process starts according to the value of $\vec{A}$ in equation 3.3. $\vec{A}$ value decreases randomly from 2 to 0 depending on the value of $\overrightarrow{r_1}$. So, variable $\vec{A}$ takes a value between [-2,2]. If the $\vec{A}$ value is greater than 1, the wolves will move away from the prey and seek more suitable prey. If the value is less than 1, they will start attacking the hunt. In the GWO, the hunting process is continued until the stop criterion is met or the specified number of iterations is reached (ŞENEL *et al.*, 2018).

3.1.4.2. Pseudocode of the algorithm

The pseudocode for the GWO is shown in (Figure 3.2).

Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent
X_β=the second best search agent
X_δ=the third best search agent
while *(t < Max number of iterations)*
    for *each search agent*
        Update the position of the current search agent by equation (3.7)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t=t+1
end while
return X_α

Figure 3.2 Pseudocode of the GWO algorithm based on (Mirjalili, 2014).

3.1.4.3. Algorithm steps and flowchart

The flowchart for the GWO algorithm is shown in Figure 3.3.

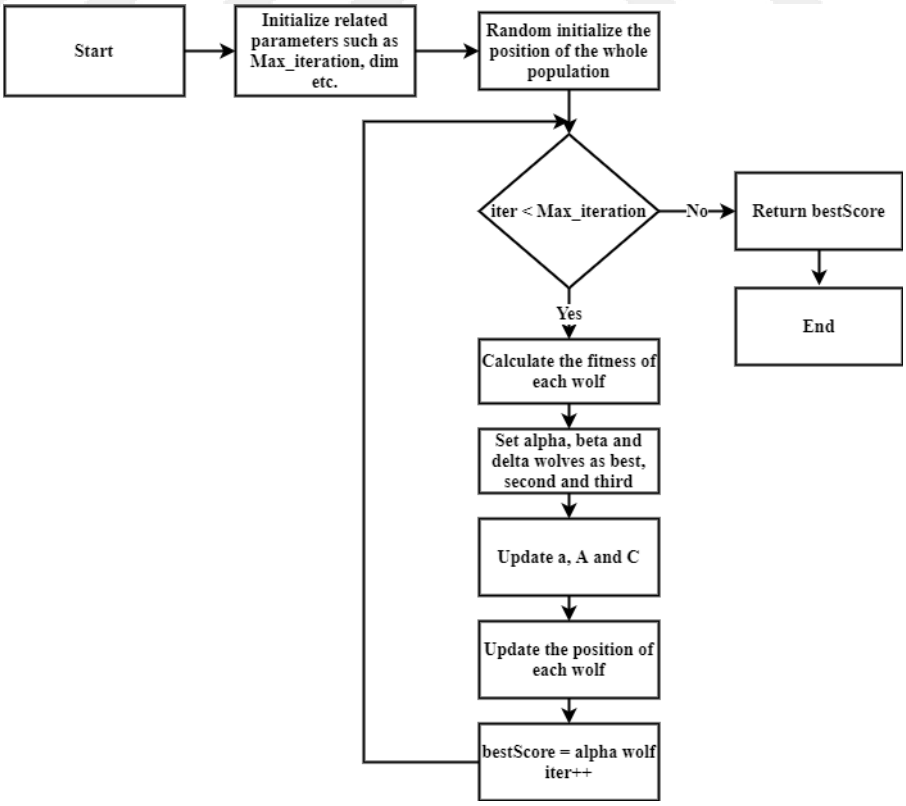


Figure 3.3 Flowchart of the GWO algorithm based on (Mirjalili, 2014).

3.1.5. Whale optimization algorithm

Whales are known as the largest mammals in the world. An adult whale can reach a length of 30 meters and a weight of 180 tons. There are very similar cells between whales' brains and human brains. These cells, which are related to judgment and emotions in social behavior, distinguish people from other living things. The number of these cells is two times higher in whales than in humans. It has been proven that whales can learn, think, and communicate. Whales usually live in groups, but there are also solitary whales (Hof and Van Der Gucht, 2007).

There are many types of whales, but WOA is modeled after humpback whales. Humpback whales hunt small fish. Humpback whales have a unique feeding behavior called an air bubble-net. They create clouds of air bubbles by breathing underwater. This large cluster of interconnected air bubbles gathers prey together. Afterward, the whale rises to the surface in the water bubbles. As it rises, it continues to bubble. This action creates a bubble circle, and the target gets squeezed inside it (Goldbogen *et al.*, 2013). Figure 3.4 shows the feeding of humpback whales.

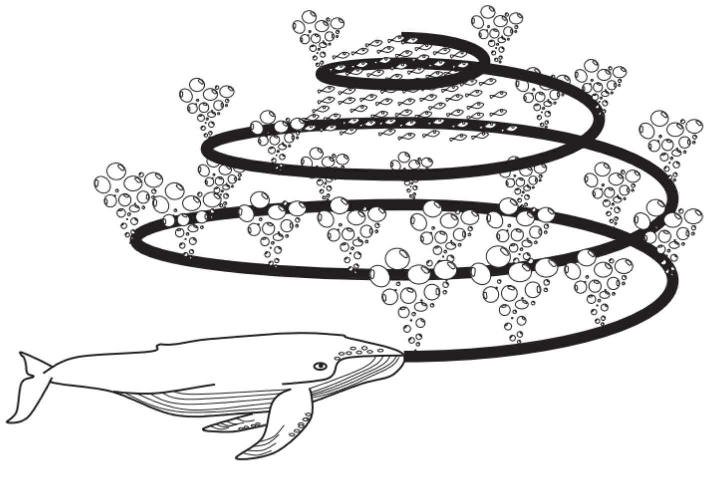


Figure 3.4 Bubble-net feeding of humpback whales (Mirjalili and Lewis, 2016).

3.1.5.1. Mathematical model

Humpback whales can predict the location of their prey. Therefore, they can surround their prey with air bubbles. In WOA, the whale's hunt is considered the optimum point to reach. Once the best search agent is determined, the locations of other search agents are updated using the best search agent. The mathematical model of the behavior of wrapping around the prey is shown in Equation 3.8 and Equation 3.9 (Mirjalili and Lewis, 2016).

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{3.8}$$

$$\vec{X}(t+1) = \left|\overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D}\right| \tag{3.9}$$

t holds the current number of iterations, $\overrightarrow{X^*}$ the best solution vector obtained. $\vec{A}$ and $\vec{C}$ express the vector coefficients. Calculation of $\vec{A}$ and $\vec{C}$ is shown in Equation 3.10 and Equation 3.11 (Mirjalili and Lewis, 2016).

$$\vec{A} = \vec{a} * (2 \cdot \vec{r} - 1) \tag{3.10}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{3.11}$$

In the equations, $\vec{r}$ [0,1] refers to the random number, and $\vec{a}$ refers to the coefficient that decreases linearly as the iteration progresses from 2 to 0 (Mirjalili and Lewis, 2016).

The spiral motion that humpback whales use while hunting is mathematically possible by decreasing the value of $\vec{a}$ in Equation 3.10. The spiral motion of the search agent and the position of the best agent are shown in Figure 3.5. Equation 3.12 is constructed by calculating the distance between the best agent and the search agent for spiral motion (Mirjalili and Lewis, 2016).

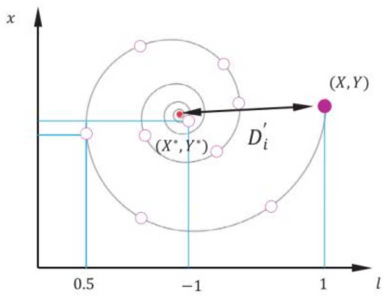


Figure 3.5 Spiral motion (Mirjalili and Lewis, 2016).

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot cos\,(2\pi l) + \overrightarrow{X^*}(t) \tag{3.12}$$

$$\overrightarrow{D'} = \overrightarrow{X^*}(t) - \vec{X}(t) \tag{3.13}$$

Equation 3.13 gives the distance between the search agent and the best-known point. b is the logarithmic spiral constant, and l is the random number between [-1,1].

Humpback whales swim around their prey simultaneously according to both the narrowing siege mechanism and the spiral movement. In WOA, it is assumed that the tapering enclosure mechanism or spiral motion can be selected with a 50% probability ratio to model this simultaneous motion, and its mathematical model is as in Equation 3.14 (Mirjalili and Lewis, 2016).

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} p < 0.5 \\ \vec{D} e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) p \geq 0.5 \end{cases} \quad (3.14)$$

The p-value in Equation 3.14 is a random number in the range [0-1]. Humpback whales can hunt their prey randomly, apart from the bubble-net method. In search of random prey, the values of the $\vec{A}$ vector randomly greater than 1 or less than -1 are used to move away from the reference whale. Unlike the bubble-net mechanism in the Prey Search mechanism, when updating the location of the search agent, a randomly selected search agent is used instead of the best search agent ever found. $\vec{A}$ > 1 and the use of a random search agent will cause the WOA to search globally. The mathematical model used here is shown in Equation 3.15 and Equation 3.16 (Mirjalili and Lewis, 2016).

$$\overrightarrow{D'} = \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \quad (3.15)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} + \vec{A} \cdot \vec{D} \quad (3.16)$$

3.1.5.2. Pseudocode of the algorithm

The pseudocode for the WOA is shown in (Figure 3.6).

```
Initialize the whales population Xi (i = 1, 2, ..., n)
Calculate the fitness of each search agent
X*=the best search agent
while (t < Max number of iterations)
        for each search agent
        Update a, A, C, l, and p
                if1 (p<0.5)
                        if2 (|A| < 1)
                                Update the position of the current search agent by the Eq. (3.8)
                        else if2 (|A| ≥ 1)
                                Select a random search agent (Xrand)
                                Update the position of the current search agent by the Eq. (3.16)
                        end if2
                else if1 (p ≥ 0.5)
                        Update the position of the current search by the Eq. (3.12)
                end if1
        end for
        Check if any search agent goes beyond the search space and amend it
        Calculate the fitness of each search agent
        Update X* if there is a better solution
        t=t+1
end while
return X*
```

Figure 3.6 Pseudocode of the WOA algorithm based on (Mirjalili and Lewis, 2016).

3.1.5.3. Algorithm steps and flowchart

The flowchart for the WOA algorithm is shown in Figure 3.7.

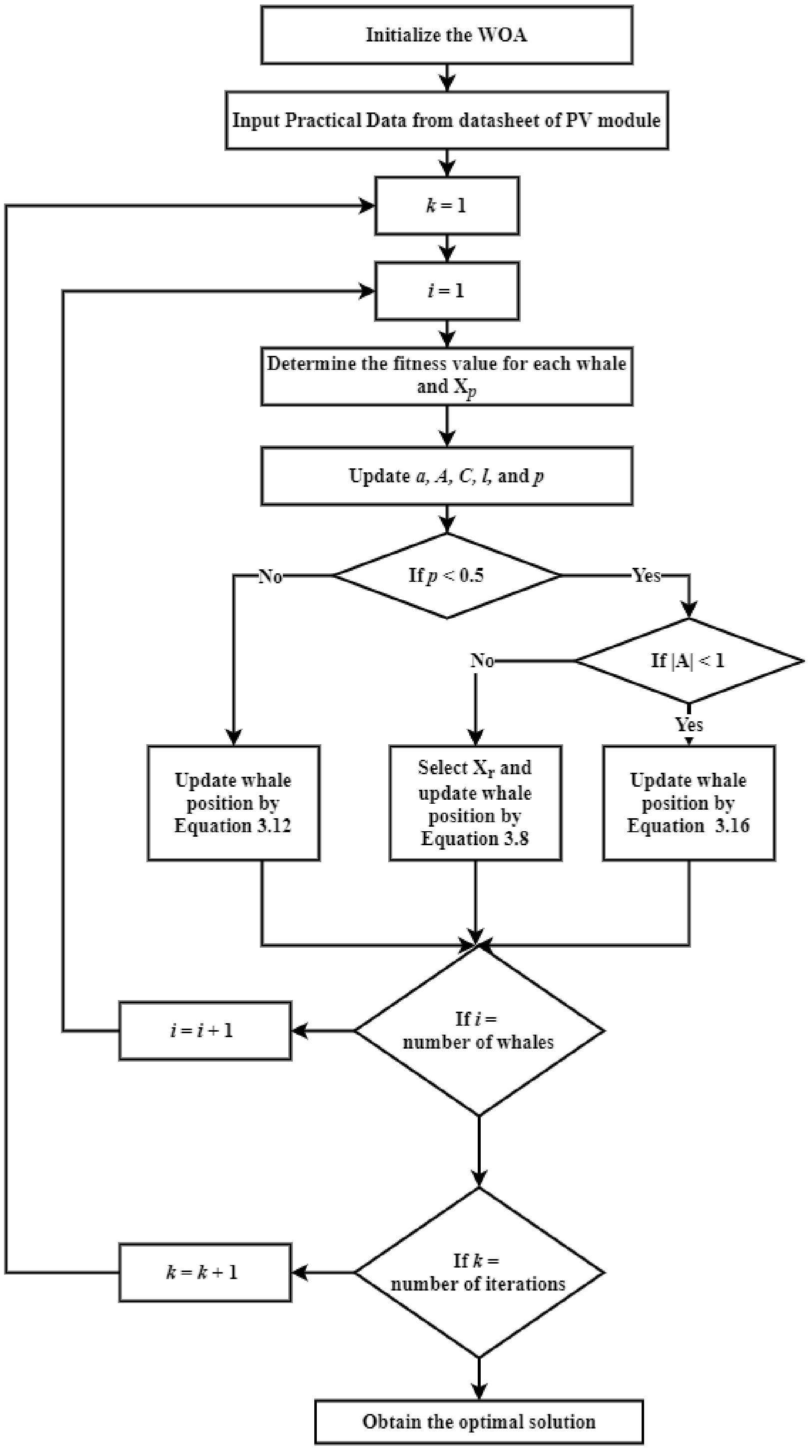


Figure 3.7 Flowchart of the WOA algorithm based on (Soliman *et al.*, 2009).

3.1.5.4. Ant lion algorithm

The ant lion algorithm imitates the movements of ant lions while hunting. Here the predator is the ant lion, the prey is the ant. ALO is a meta-heuristic algorithm introduced by Mirjalili (Mirjalili, 2015).

Ant lions create their traps in the shape of a cone by drawing a circular path to the areas where the ants are located. They then bury themselves at the bottom of this cone. When the ants enter this cone, they start throwing sand to prevent them from coming out of the cone (Kilic and Yüzgeç, 2018). The drawing of the trap in the form of the cone made by the ant lions to catch the ants is shown in Figure 3.8.

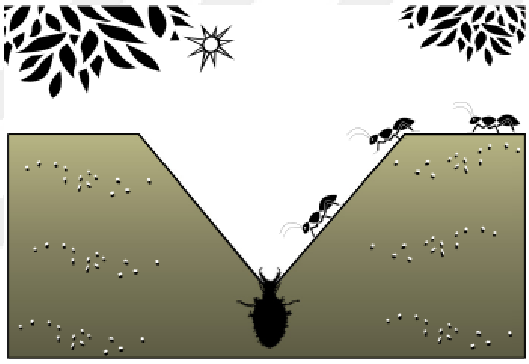


Figure 3.8 Cone-shaped traps (Mirjalili, 2015)

3.1.5.5. Mathematical model

The mathematical model of the hunting mechanism first begins with a random walk.

$$X(t) = \begin{bmatrix} 0 \\ cumsum\ (2r(t_1) - 1) \\ cumsum\ (2r(t_2) - 1) \\ \vdots \\ cumsum\ (2r(t_n) - 1) \end{bmatrix} \tag{3.17}$$

n means the maximum number of iterations, r means random walking steps, and $cumsum$ means cumulative total. Below is the definition of $r(t)$ (Mirjalili, 2015).

$$r(t) = \begin{cases} 1, & if\, rand > 0.5 \\ 0, & if\, rand \leq 0.5 \end{cases} \tag{3.18}$$

There are five main stages of ant lion hunting. These are the random walk of the ants, the ants falling into the ant lion's trap, the building of the trap, the sliding of the ants towards antlion, the catching and rebuilding of prey, and lastly, elitism (Mirjalili, 2015).

During optimization, the ants' positions are updated by a random walk. During this search process, normalization is done to ensure that the locations remain within the boundaries. The mathematical model of this operation is expressed as in Equation 3.19 (Mirjalili, 2015).

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{b_i - a_i} + c_i^t \tag{3.19}$$

Here i shows the number of variables, the number of t iterations, a and b refer to the minimum and maximum random walk, respectively. $c,$ and d represent the minimum and maximum value of the ant lion positions changing in every iteration, respectively.

The ants' walks are also affected by the ant lions. When the ant enters the trap, the ant lion starts throwing sand at it. The mathematical model of this operation is expressed as in Equation 3.20 and Equation 3.21 (Mirjalili, 2015).

$$c_i^t = Antlion_j^t + c^t \tag{3.20}$$

$$d_i^t = Antlion_j^t + d^t \tag{3.21}$$

$$c^t = c^t \cdot I^{-1} \tag{3.22}$$

$$\mathrm{d^t = d^t \cdot I^{-1}} \tag{3.23}$$

Where I represent a shift ratio, c^t represents the minimum of all variables in the t-th iteration, and d^t represents the vector containing the maximum of all variables in the t-th iteration (Mirjalili, 2015).

The mathematical equation for capturing the ants by their hunters and rebuilding the pit is shown in Equation 3.24 (Mirjalili, 2015).

$$Antlion_j^t = Ant_i^t, \qquad f(Ant_i^t) > f\big(\, Antlion_j^t\big) \tag{3.24}$$

In Equation 3.24, Antlion $_j^t$ represents the position of the j-th ant lion in the t-th iteration, and Ant_i^t represents the position of the i-th ant in the t-th iteration.

Elitism is one of the most important features of evolutionary algorithms. In the ALO algorithm, the best ant lion obtained in any iteration is recorded as elite. As the elite is the most suitable ant lion, it can direct the movements of the remaining ants through iterations. In Equation 3.25, elitism mechanism is given as a mathematical model (Mirjalili, 2015).

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{3.25}$$

In Equation 3.25, R_A^t shows the random walk throughout the ant lion selected by the roulette wheel method in the t-th iteration. R_E^t shows the random walk throughout the elite ant lion in the t-th iteration, and Ant_i^t represents the position of the i-th ant in the t-th iteration (Mirjalili, 2015).

3.1.5.6. Pseudocode of the algorithm

The pseudocode for the ALO is shown in (Figure 3.9).

```
Initialize the first population of ants and antlions randomly
Calculate the fitness of ants and antlions
Find the best antlions and assume it as the elite (determined optimum)
while the end criterion is not satisfied
        for every ant
                Select an antlion using Roulette wheel
                Update c and d using equations Eqs. (3.22) and (3.23)
                Create a random walk and normalize it using Eqs. (3.17) and (3.19)
                Update the position of ant using (3.25)
        end for
        Calculate the fitness of all ants
        Replace an antlion with its corresponding ant it if becomes fitter (Eq. (3.24))
        Update elite if an antlion becomes fitter than the elite

end while
Return elite
```

Figure 3.9 Pseudocode of the ALO algorithm based on (Mirjalili, 2015).

3.1.5.7. Algorithm steps and flowchart

The flowchart for the ALO algorithm is shown in Figure 3.10.

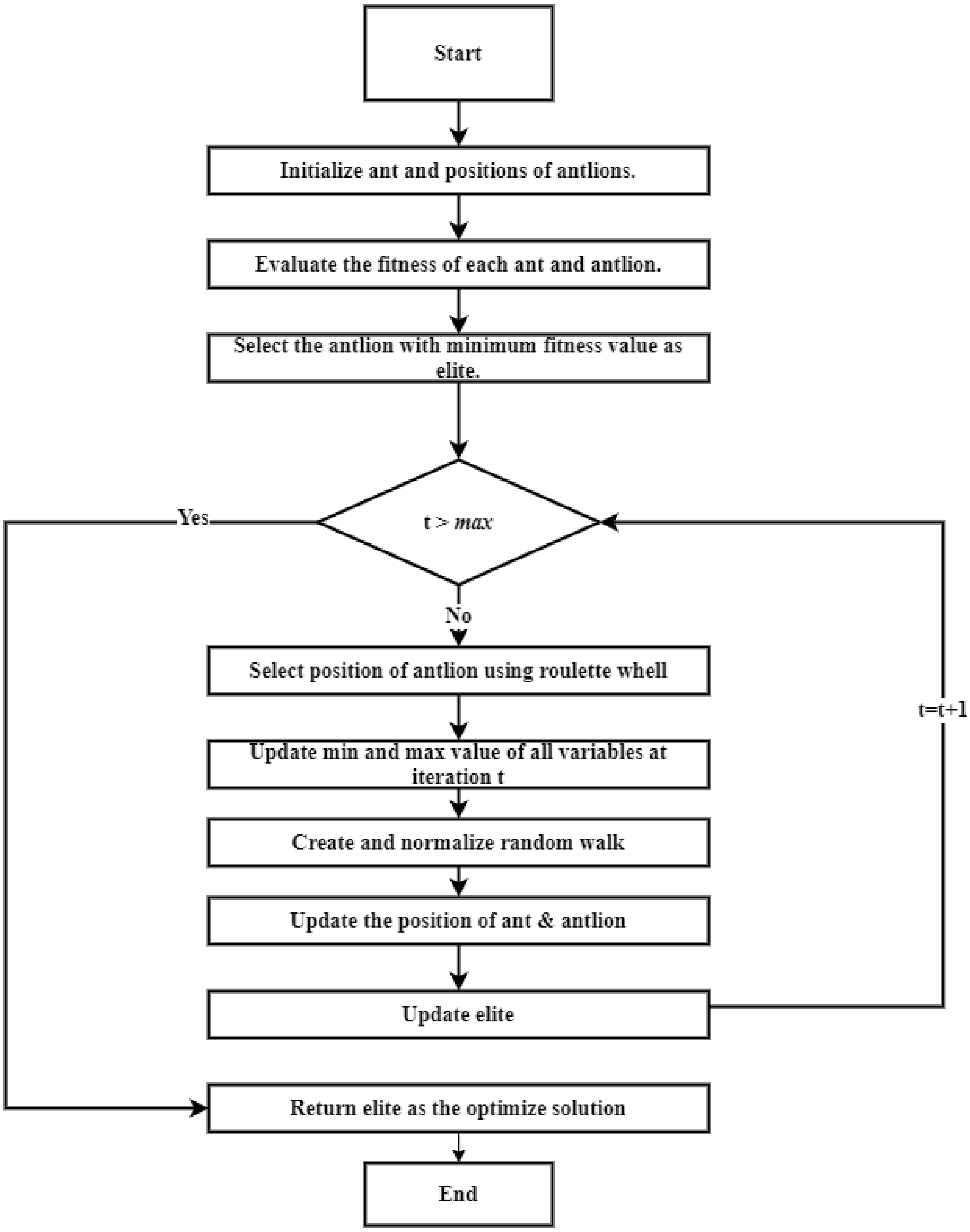


Figure 3.10 Flowchart of the ALO algorithm based on (Pradhan *et al.*, 2019).

4. IMPLEMENTATION

The aim of this thesis to demonstrate the usability of meta-heuristic algorithms to optimize voltage deviations in electrical distribution systems and to present it to researchers who will work on optimization in energy systems by making it an educational, open-source program. The software application is developed using MATLAB programming language. In order to run load flow in test systems, MATPOWER package with open-source code was preferred (Zimmerman, Murillo-Sánchez and Thomas, 2011).

MATPOWER uses the Newton-Raphson method when running power flow. Because of the high R/X in distribution systems, we also used the Forward / Backward Sweep (FBS) method in this study. When applying forward sweep, the node voltage calculation is made from the sending end to the far end and laterals of the feeder. The backward sweep step is used to calculate the branch current and power from the far end of the feeder to the sending end and laterally (Eminoglu and Hakan Hocaoglu, 2009).

4.1. Benchmark Functions

When starting our research, we tested the success of the meta-heuristic algorithms mentioned in the thesis with various benchmark functions. In this study, we computed Sphere (Equation 4.1), Ackley (Equation 4.2), Rosenbrock (Equation 4.3), and Rastrigin (Equation 4.4) benchmark functions with GWO and WOA algorithms using MATLAB.

$$f_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 \tag{4.1}$$

$$f_2(\boldsymbol{x}) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos\ 2\pi x_i\right) + 20 + e \tag{4.2}$$

$$f_3(\boldsymbol{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{4.3}$$

$$f_4(\boldsymbol{x}) = \sum_{i=1}^{n} [x_i^2 - 10cos\,(2\pi x_i) + 10)] \quad (4.4)$$

Table 4.1 describes the values at which the test functions operate, where the value x will be between the specified lower bound and upper bound values.

Table 4.1 Constant values of test functions.

Benchmark Function	Iterations	Number of Runs	Lower bound	Upper bound
f_1	1000	30	-100	100
f_2	1000	30	-30	30
f_3	1000	30	-5.12	5.12
f_4	1000	30	-32	32

The specified benchmark functions were calculated in 2, 5, 10, and 20 dimensions using GWO and WOA algorithms. The best, worst, mean and standard deviation values are given in Table 4.2.

Table 4.2 The result of calculating test functions with meta-heuristic algorithms.

f	n	GWO				WOA			
		AVG	**STD**	**Best**	**Worst**	**AVG**	**STD**	**Best**	**Worst**
f_1	2	0	0	0	0	3.37E-220	0	8.68E-254	3.37E-219
	5	5E-223	0	1E-236	4E-222	7.56E-167	0	2.80E-190	7.45E-166
	10	2E-133	6E-133	2E-139	2E-132	1.60E-152	4.80E-152	1.34E-174	1.60E-151
	20	4.4E-84	1.1E-83	3E-87	3.8E-83	5.08E-158	0	1.30E-164	3.78E-157
f_2	2	2.9E-07	2.2E-07	3.1E-08	8.2E-07	1.99E-06	4.06E-06	9.38E-09	1.38E-05
	5	1.67635	1.04095	0.48455	4.4718	1.13294267	0.9694327	0.42961575	3.9332377
	10	6.63122	0.94966	5.24294	8.83984	30.0997561	72.216665	5.22280388	246.746818
	20	16.1694	0.7313	15.1783	17.1756	16.6910744	0.5162604	16.1804178	17.9582115
f_3	2	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0
	10	0.20188	0.60564	0	2.01881	0	0	0	0
	20	3.68046	2.8332	0	8.75532	0	0	0	0

f_4	2	4.4E-16	9.9E-32	4.4E-16	4.4E-16	1.15E-15	1.42E-15	4.44E-16	4.00E-15
	5	1.9E-15	1.7E-15	4.4E-16	4E-15	2.22E-15	1.78E-15	4.44E-16	4.00E-15
	10	5.1E-15	1.6E-15	4E-15	7.5E-15	2.93E-15	1.63E-15	4.44E-16	4.00E-15
	20	7.9E-15	1.1E-15	7.5E-15	1.1E-14	2.93E-15	1.63E-15	4.44E-16	4.00E-15

As a result of the tests, it was seen that using meta-heuristic algorithms, values of benchmark functions near to optimum can be calculated. As the number of runs and iterations increases in meta-heuristic algorithms, the probability of convergence to the optimum value also decreases. These test results are calculated for 1000 iterations and 30 runs. Another variable that affects the optimum value is n. The value of n is also called a dimension.

4.2. Distributed Energy Resources In Energy Systems

Voltage regulation in electrical distribution systems is generally performed by on-load tap changers, capacitor banks, and voltage regulators. Even though these parts are cost effective, the voltage regulation they provide is not continuous (Rizy *et al.*, 2011).

The unit of active power is watts (W). In the equation, active power is expressed with the letter (P). Reactive power is indicated by (Q). Its unit is volt-amperes reactive (var). Apparent power refers to the total power drawn from the grid. It is expressed as apparent power (S) and calculated as in Equation 4.5 (Kuphaldt, 2007).

$$|S| = \sqrt{P^2 + Q^2} \tag{4.5}$$

Distributed energy resources can also be used to regulate distribution system voltage magnitudes. Inverter-based distributed energy systems can absorb/inject reactive power. As shown in Equation 4.5, a small change in active power output will provide a fairly wide range of reactive power capacity. Based on this, distributed energy systems can be used to control voltage in electrical distribution systems (Ceylan *et al.*, 2014).

4.3. Use of MATPOWER Software

MATPOWER is an open-source MATLAB programming package that provides power flow and optimum power flow solutions. This package contains pre-defined functions and predefined test cases for use in power systems. One of the test systems in the package is the 33 bus system (Baran and Wu, 1989a).

First, runpf method is run to solve the load flow problems. In order for this method to work, the first parameter to be passed is the name of the test case to be used. The name of the 33 bus test system is defined as 'case33bw'. To run the load flow, runpf ('case33bw') is run in the MATLAB program.

In some cases, it may be necessary to update the values defined in the test system. The loadcase method is used to define a system for value assignment before running it. Case can be run after the necessary changes are made. This piece of code can be seen in Figure 4.1 (Buayai *et al.*, 2014).

```
foo=loadcase('case33bw'); % Load 33 bus test case
foo.bus(1)=70; % Update real power demand at bus 5 to 70 MW
runpf(foo);
```

Figure 4.1 Example code for load case, update values and run case.

After a power flow simulation, the results are shown on the screen. The result values of the power flow simulation can be reached by accessing the used variable. Every time the runpf method is called, it prints the output to the screen. The printouts on the screen both cause the program to run slowly and make it difficult to follow. The runpf method takes the options variable as the second parameter. The type of this variable is mpoptions. This piece of code can be seen in Figure 4.2 (Buayai *et al.*, 2014).

```
mpopt = mpoption('verbose', 0, 'out.all', 0);
foo=loadcase('case33bw');
runpf(foo,mpopt);
```

Figure 4.2 Example code for disabling print out of runpf method.

4.4. Data Collection

This study, it is aimed to minimize the voltage deviations by adding distributed energy resources, batteries, and tap changers to 33 bus, 69 bus, and 141 bus test systems. In order to run load flow of the test system with real data, the hourly load profile of a randomly selected consumer from the commercial and residential hourly load profiles in the United States was selected on May 31, 2013 (Office of Energy Efficiency & Renewable Energy (EERE), 2014). The load profiles of the distributed energy resources (photovoltaics) to be added to the test system were obtained from renewables.ninja (Pfenninger and Staffell, 2016; Staffell and Pfenninger, 2016).

In the hourly load profile, we consider the value at 12 noon as the base case value and proportion the values in the other hours according to the base case. The values we obtained as a result of this proportioning are as in Table 4.3.

Table 4.3 Hourly load values for test case.

Hour	Load (MW)
0:00	1.12389
1:00	1.12831
2:00	1.14268
3:00	1.16257
4:00	1.18474
5:00	1.23023
6:00	1.30551
7:00	1.37532
8:00	1.37234
9:00	1.35075
10:00	1.27216
11:00	1.25226
12:00	1
13:00	0.85029
14:00	0.74154
15:00	0.67476
16:00	0.65657
17:00	0.70156
18:00	0.90661
19:00	0.85619
20:00	0.93823
21:00	1.01495

22:00	1.08295
23:00	1.07826

Figure 4.3 shows the hourly variation of load values.

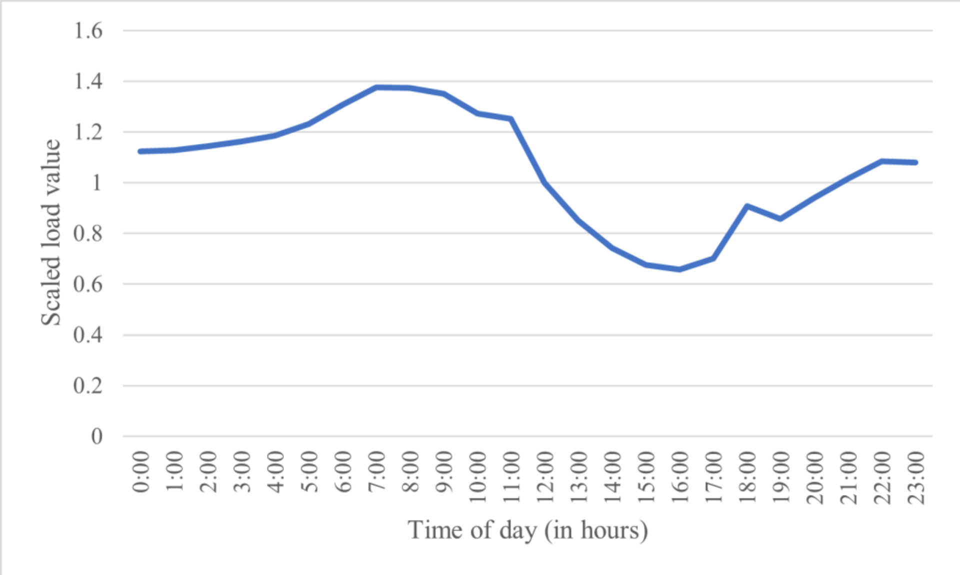


Figure 4.3 Scaled hourly load values.

On renewables.ninja website, where we obtained the solar PV data, the capacity and system loss of the PV can be adjusted. In this study, we have determined the solar PV capacity as 10 kW and system loss as 0.1. The 24-hour solar energy load records selected from the middle of August 2019 can be seen in Table 4.4.

Table 4.4 Hourly load values for PV.

Hour	**Load (kW)**
0:00	0
1:00	0
2:00	0
3:00	0
4:00	0
5:00	0
6:00	0.057
7:00	0.729
8:00	1.564
9:00	2.741
10:00	4.246
11:00	5.871
12:00	6.261
13:00	6.352

14:00	6.097
15:00	5.39
16:00	3.872
17:00	2.514
18:00	1.107
19:00	0.139
20:00	0
21:00	0
22:00	0
23:00	0

Figure 4.4 shows the hourly variation of PV ouputs.

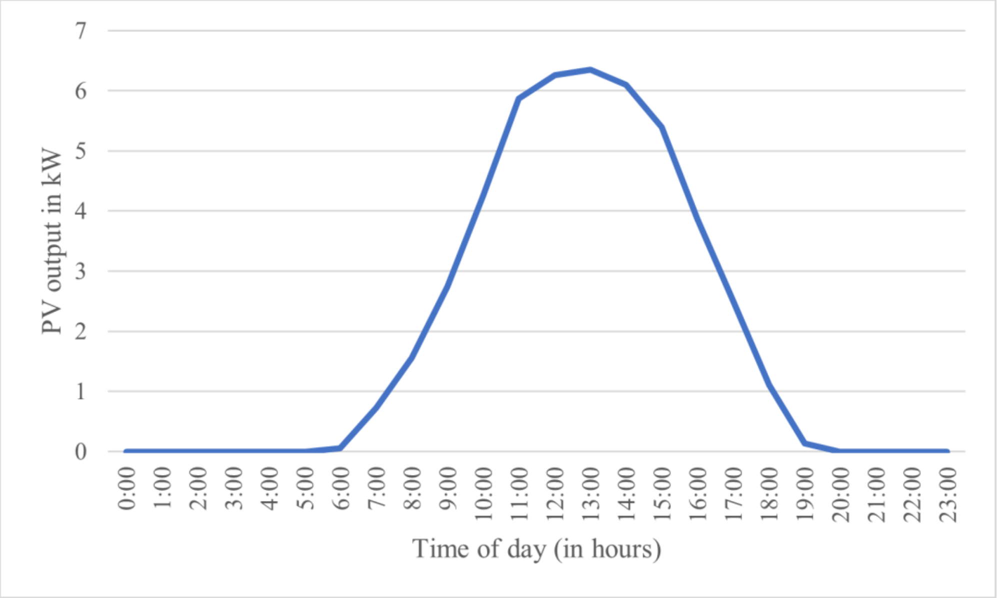


Figure 4.4 Hourly PV active power outputs in kW.

In this case, the S (constant) at time t is 10 kVA, and the P is the active power output of PV specified in Table 4.4. Using Equation 4.5, reactive power can be calculated. The reactive power was calculated in Equations 4.6a and 4.6b.

$$|10| = \sqrt{(6.261)^2 + Q^2} \tag{4.6a}$$

$$Q = -\sqrt{60.799879} \;\; or\, Q = \sqrt{60.799879}\,, \sim \pm 7.7974 \tag{4.6b}$$

4.5. Minimize Objective Function with Random Walk Algorithm

First, we implemented static variable definitions. These variables are the hourly load profile, the positions where the PV's are, the PV number on each bus, the hourly load values of the PV's, and the apparent load power for the PV's. We assumed that we place 8 PVs per bus at bus points 15, 18, 20, 25, and 30. We divide the apparent power and PV load values by 100 to convert them to MW. The purpose of the objective function that we will use in this study is to bring the voltage magnitudes as close to 1 pu as possible and to prevent losses. The mathematical definition of the objective function can appear in Equation 4.7.

$$\begin{aligned} &\underset{X}{minimize} && \sum_{i=1}^{N} \mid V_i - 1 \mid^2 \\ &subject\ to && 0.95 \leq V_i \leq 1.05 \\ & && P_{DER_i}^2 + Q_{DER_i}^2 \leq S_{DER_i}^2 \end{aligned} \tag{4.7}$$

Where, V_i represents the voltage magnitude at bus i. P_{DER} and Q_{DER} refer to the active and reactive power output of DER, respectively. S_{DER} represents the apparent power of DER. Before adding a metaheuristic algorithm to our code, we tested it with the random walk algorithm we wrote. Using the apparent power equation (Equation 4.5), we determined the upper and lower boundaries of the positions that the algorithm will use. Upper and lower bound values are calculated as in Figure 4.5.

```
ub = sqrt((S^2) - (pvoutputs(n)^2 )); % calculate upper bound
lb = -1 * ub; % calculate lower bound
```

Figure 4.5 Code fragment in which static values are defined.

The objective function has been calculated using the random walk algorithm. First of all, two values named bestFitness and bestFitnessVector are defined to store temporary values. In general, the fitness values to be calculated in each round will be compared with the temporary bestFitness value, the lower value will be set as bestFitness. The loop will run until it reaches maximum iteration. In each round, a vector containing random positions is defined. The random values that the vector can take must be between upper bound and lower bound values. Random values are calculated for each PV position. In each cycle, columns 3 and 4 of the test system are updated with the determined hourly load values. These columns contain the values of P_d and Q_d, respectively. P_d means active load, and Q_d means reactive load. The PV number is multiplied by the hourly output values of the PVs, and the P_d value is updated. To update the value of Q_d, the relevant value is taken from the randomly calculated vector, and if it is less than 0, it is added to the Q_d value. If it is greater than 0, it is subtracted from the Q_d value.

The test case with updated values is run. The V_m value vector obtained from this system is sent to the objective function as a parameter. The result of the Objective function is assigned as the fitness value. If the fitness value is better than the temporary fitness value (bestFitness), the bestFitness value and the bestFitnessVector values are updated.

Two variables are defined as defaultCase and optimCase. DefaultCase variable refers to the test system that was run without any optimization. OptimCase variable, on the other hand, runs a power flow using the calculated optimal solution using the bestFitnessVector value calculated in the previous step. MATLAB plot command is used to plot the graph of both running test systems. Plot commands are written between hold on and hold off commands in order to display both data on the same graph.

In Figure 4.6, the values of the random walk optimized system are shown with a continuous orange line. The base case system is shown with a dashed blue line.

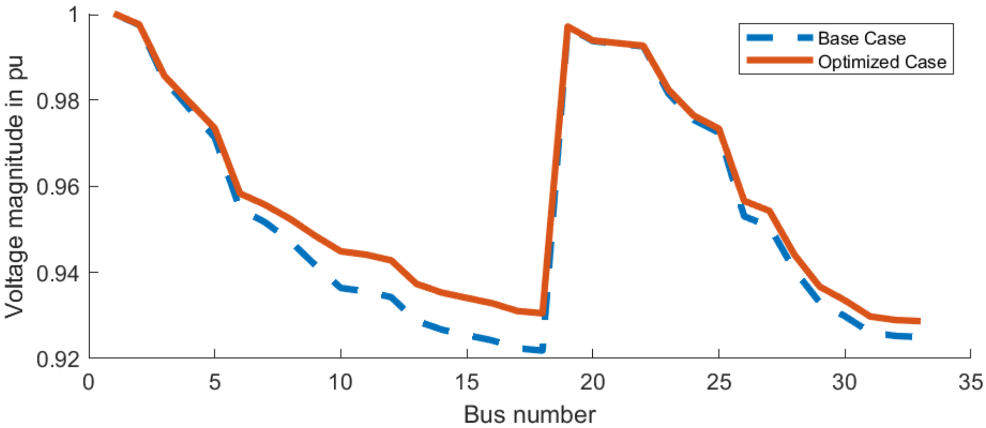


Figure 4.6 The graph of the change of the pu values of the base case and optimized case 33 bus test system.

4.6. Implementation of the Tap Changer on Solving Voltage Deviation Problem

Since the FBS method is not compatible with the case files defined in MATPOWER, we have created a function for the active and reactive power values of the 33, 69, and 141 bus test systems. We wrote a code that runs the FBS algorithm by taking the determined active, reactive power, and branch values. The mathematical definition of the target function using the tap changer is made in Equation 4.8.

$$\begin{aligned} \underset{X}{minimize} \quad & \sum_{i=1}^{N} | V_i - 1 |^2 \\ subject\ to \quad & 0.95 \leq V_i \leq 1.05 \\ & T^{min} \leq T_i \leq T^{max} \end{aligned} \tag{4.8}$$

Where, V_i represents the voltage magnitude at bus i. T_i, T^{min}, and T^{max} represent the regulator's actual tap position, minimum tap position, and maximum tap position, respectively. Voltage regulation can be made by using tap changers. Tap changers can take different values to bring the voltage value closer to 1 pu. Each step-change changes the voltage value between 0.00625 pu, in other words, 5% - 8%. The maximum and minimum tap changer values are assumed to be ± 16 (Daylak, 2016). The values of tap changers were found by using GWO, WOA, and ALO algorithms.

Figure 4.7 change of the pu values of the base case and optimized case with tap changer at 33 bus test system. The tap changers are placed in the 6th and 26th buses. Optimization was made using the GWO algorithm at 17:00 according to the load values given in Table 4.3.

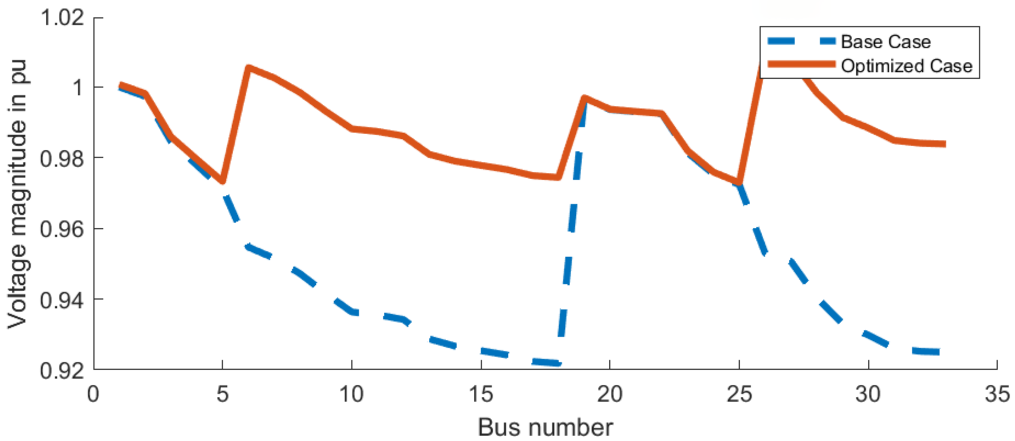


Figure 4.7 The graph of the change of the pu values of the base case and optimized case with tap changer at 33 bus test system.

4.7. Implementation of the Battery on Solving Voltage Deviation Problem

Energy storage is the storage of energy obtained by various techniques for later use. In our daily life, there are batteries in our smartphones and portable electronic devices.

The most common technologies on the market today are lithium-based, lead-based, nickel-based, and sodium-based batteries. Lithium-based batteries have a wide range of applications, but their full utilization potential has not yet been reached (EASE and EERA, 2014).

The system developed for electrical charge storage using specially developed batteries is called the Battery Energy Storage System (BESS). BESS has a wide range of uses nowadays, and these techniques will develop in the future. Today, BESS from 2 kW to 50 MW can be used in a distribution network. The capacity of BESSs can range from 5 kWh to MWhs (*Battery Energy Storage for Smart Grid Applications*, 2013). They can be used for renewable energy or as stabilizers in high, medium, or low voltage substations.

Batteries can energize or draw power from the system by the operating logic. A battery can be considered as energy source when powering the system and as a load when it absorbs power from the system. Remarkably, batteries can be helpful at night when the solar panels used in the system are not generating power. The charge level of a battery according to its capacity is called as state of charge (SoC). An SoC value of 0% indicates that the battery is empty, and 100% means fully charged.

To ensure that the batteries can be used for a long time, SoC values were determined as 20% to 80% in this study (*Battery Charging and Discharging Parameters | PVEducation*, no date). The capacity of the BESS used in the study is 100 kW. The BESS output power started on 30 kW and was decreased or increased according to the needs of the system with 5 kW at each hour. We conducted a 24-hour analysis to examine the charging and discharging status of the batteries. We calculated the charging and discharging conditions of the batteries using meta-heuristic algorithms. We ran a load flow analysis for the 24 values generated by the algorithms and tried to maximize the objective function.

Figure 4.8 shows the power change over time in kW of two batteries initiated at 30 kW.

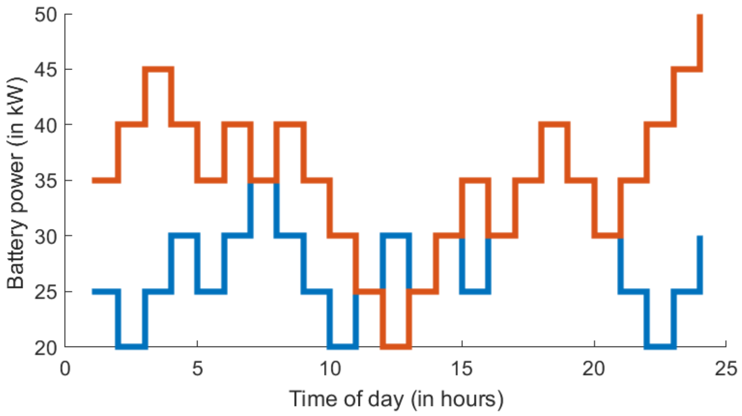


Figure 4.8 A sample graph showing the hourly power change of two batteries.

4.8. Visualization of the Test Systems

We have developed a dynamic structure so that users can see the test system they choose and the DG, tap changer, and batteries they place on the system more efficiently. We made the drawings of the test systems on the draw.io site. We saved the drawings we made here in SVG format. It is an image format that dynamically draws elements contained in SVG.

Since it is difficult to use the SVG format directly in MATLAB, we have placed an HTML view component in our application. This component loads the corresponding HTML file with SVG files. Thanks to the component, the javascript codes in the HTML file and the MATLAB interface can communicate. In this way, DG, tap changer, and battery positions selected in the MATLAB interface can be sent to javascript. The javascript file colors the buses and lines in the SVG files according to the incoming data. In Figure 4.9, there is an example drawing of the 33-bus system.

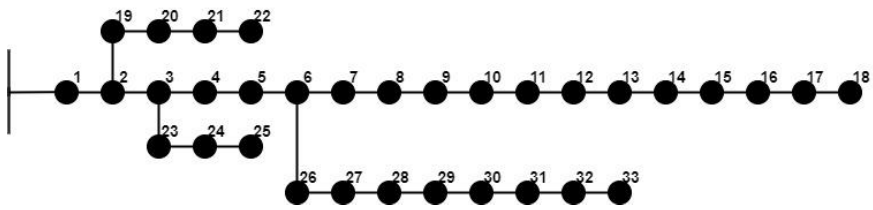


Figure 4.9 Vector drawing of the 33 bus test system.

4.9. Application with Meta-Heuristic Algorithms

The source codes of the GWO, WOA, and ALO optimization algorithms used in the study were obtained from Seyedali Mirjalili's website (Mirjalili, no date). There are some parameters necessary for all three algorithms to be executed. These parameters are the objective function, the number of search agents, the number of iterations, dimension, upper bound, and lower bound.

Figure 4.10 shows the integration of meta-heuristic algorithms with our software. At the beginning of the program, the values to be used as parameters are defined, then the codes of the relevant algorithm are executed according to the user selection. Each algorithm calculates the optimal value of the objective function defined in our source code. The optimal solution calculated is returned.

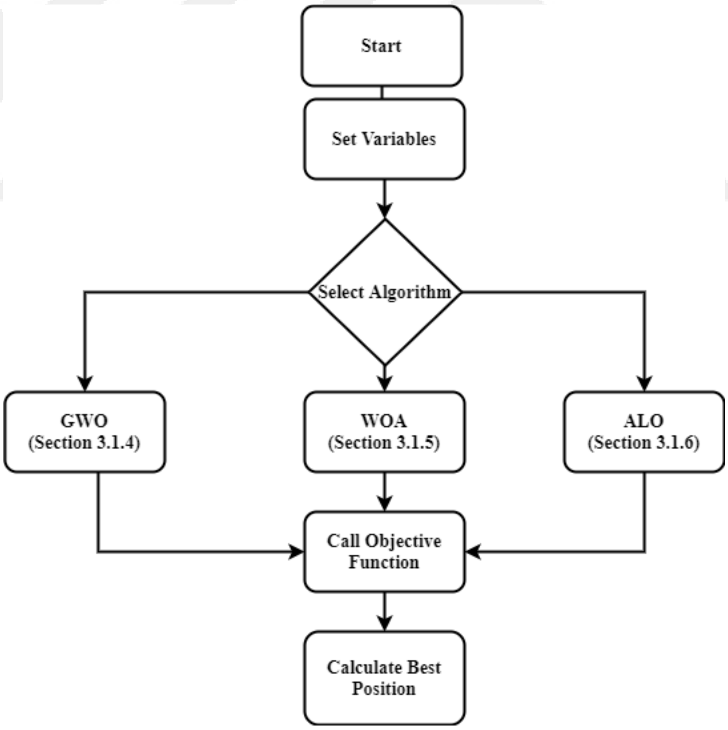


Figure 4.10 Flowchart of the custom software.

We have defined an objective function that the algorithm will calculate. The params variable passes the variables that need the function. These variables are values such as the name of the test system, hourly load and PV load values, PV numbers, and PV positions. These values will be set from the GUI.

5. GRAPHICAL USER INTERFACE

The interface of the application is programmed using the MATLAB App Designer tool. There are two different types of simulation in the application. In the first mode, voltage deviation can be minimized by DG and tap changer; In the second mode, the voltage deviation can be minimized by battery and tap changer. Various information is obtained from the users according to the selected mode. Analyses can be performed on 33 bus, 69 bus, and 141 bus test systems in the program. During these analyzes, GWO, WOA, and ALO algorithms are used.

5.1. Structure and General Information

The interface consists of 3 different panels. These are the settings panel, preview panel, and results panel. There may be some changes in the panels depending on the simulation mode selected. DG and tap changer mode works for hourly analysis, while battery and tap changer mode works daily.

The compiled application extension is .mlapp. This extension can be installed via MATLAB and accessed from the apps tab in the MATLAB interface. The positioning of the panels mentioned in the Figure 5.1 is shown on the application.

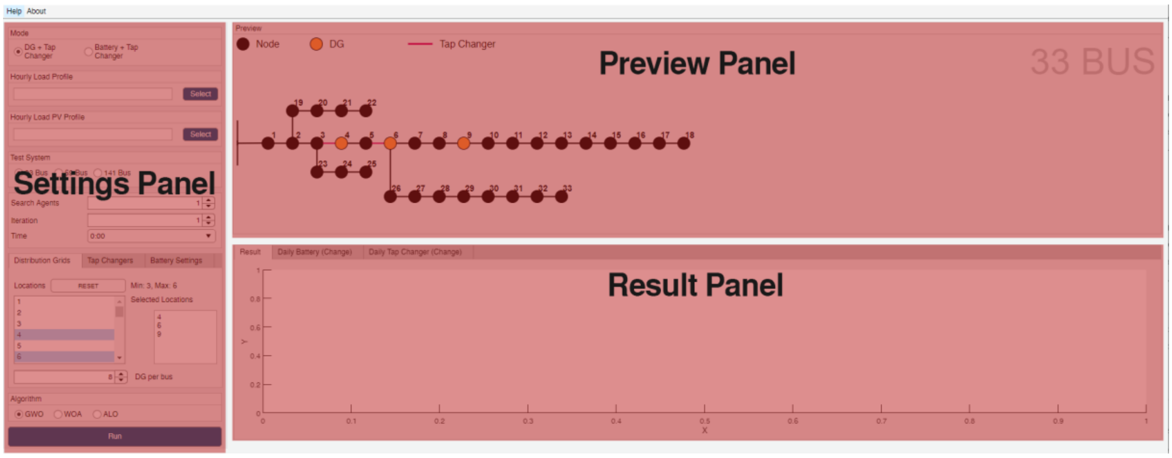


Figure 5.1 Placement of 3 panels on the application.

5.2. Settings Panel

The values that need to be adjusted to minimize the voltage change are controlled from this panel. A screenshot of the panel is shown in Figure 5.2. The fields on the figure are numbered from 1 to 13. The use and rules of these areas are specified items by item.

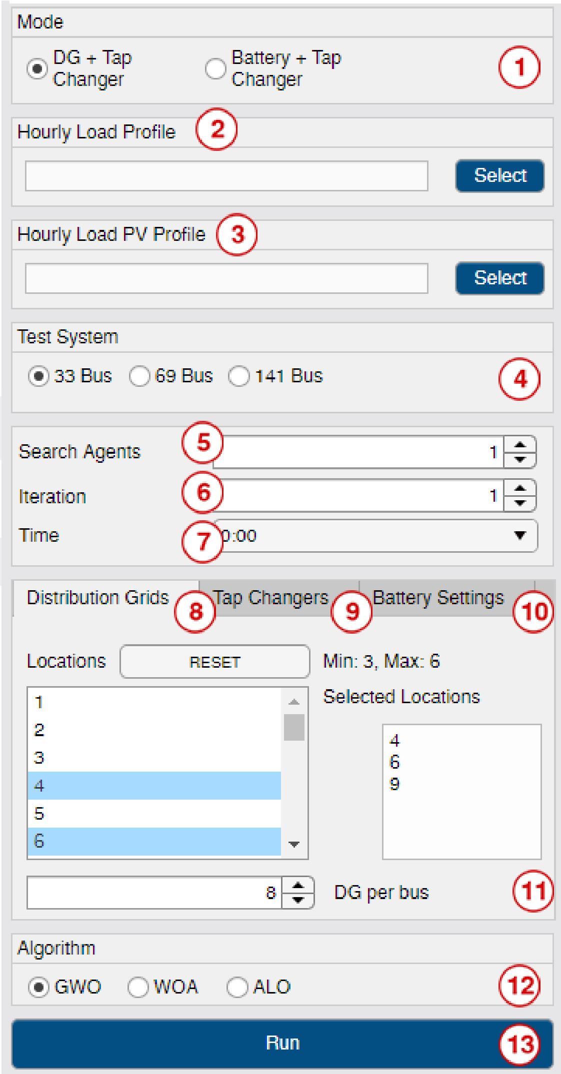


Figure 5.2 Settings panel of the software.

The meanings and usage rules of these fields are as follows:

1. This field shows the mode to be used during voltage deviation minimization. DG and tap changer mode is operated hourly, battery and tap changer mode is operated daily.

2. It is a TXT or CSV file containing the hourly load profile. When the Select button is pressed, the file can be selected from the opened file selection tool. The file should only contain 24-hour power output data in a comma-separated format.
3. It is a TXT or CSV file containing the per-hour PV load profile. When the Select button is pressed, the file can be selected from the opened file selection tool. The file should only contain 24-hour power output data in a comma-separated format. This setting works only if DG and tap changer mode is selected.
4. The user can choose which test system he wants to use from this field. By default, 33 bus system is selected.
5. The number of search agents is set in this field. The larger the number, the higher the probability of reaching near-optimum results. Numerically small numbers can be used for faster results.
6. The number of iterations during optimization is determined by this field. The situations that are valid in the search agent field are also valid in this field.
7. It specifies which value will be taken from the 24-hour load data loaded. This setting works only if DG and tap changer mode is selected.
8. The points where DG will be placed on the selected buses are selected in this field. The minimum and maximum number of DGs that can be selected for each test system are defined. There are numbers in the range starting from 1 in the selection box to the number of buses of the selected test system. More than one bus selection can be made by holding down the CTRL-key. The chosen locations appear on the side and are updated on the preview screen. All selections can be reset by clicking the Reset button. If battery and tap changer mode is active, this field is updated as the battery location.
9. The operating rules of this area are the same as the previous one. The only difference is that the selection boxes start from 2, as the tap changers are placed on the lines between the buses, not on the buses.
10. The initial output value of the battery, the power value it will give to or receive from the system per hour, and the lower and upper SoC limits of the battery can be determined here. This field works only if the battery and tap changer mode is active.

11. How many DGs with the same feature will be added to the selected bus can be set here. If the battery and tap changer mode is active, this field is updated as the number of batteries.
12. The algorithm to be used is selected from this area.
13. The optimization is run by pressing the button. If an error has occurred with the selected values, an error message is displayed after pressing this button. If there is no error, the selected algorithm will be run.

5.3. Preview Panel

DG, tap changer, and battery positions selected from the settings panel are dynamically displayed on the scheme of the chosen test system. Each bus is shown in black, DG and batteries are shown as an orange round, and tap changers as a red line. The screenshot of this panel is shown in Figure 5.3.

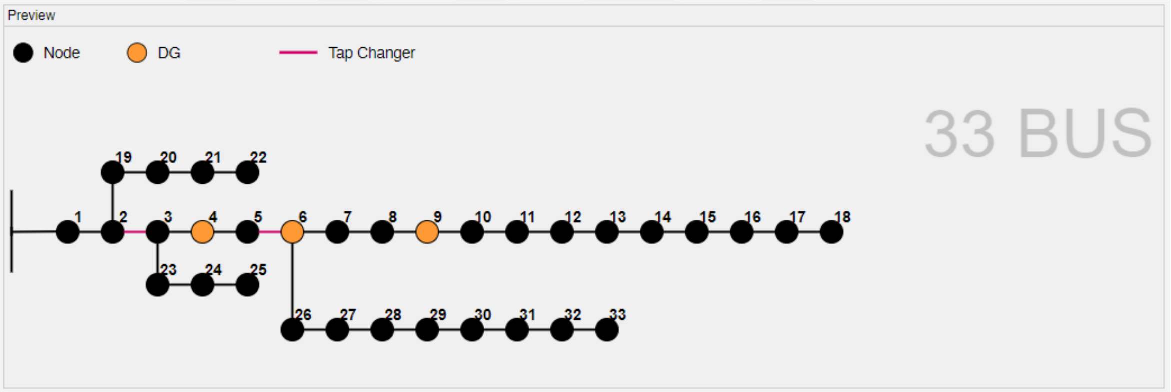


Figure 5.3 Screenshot of the preview panel.

5.4. Results Panel

The results panel has a 3-tab structure. The titles of these tabs are the result, daily battery (change), and daily tap changer (change), respectively. The contents of these tabs vary according to the active mode. If the DG and tap changer mode is active, only the result tab will work, and a graph of the optimized and base case load flow analysis will appear on this tab.

When battery and tap changer mode is active, all tabs can be used. The results tab contains graphs for 24-hour optimized and base case load flow results. The daily battery tab includes the change of the installed batteries during the day, and the daily tap changer tab contains the change of the installed tap changers during the day.

5.5. Test Results

Various tests have been carried out using the interface. Using the load values given in Table 4.3 and Table 4.4, load flow was run in a 33 bus test system. The number of search agents is set to 50 and the number of iterations to 1000. 8 PVs are placed on buses 15, 16, 19, 20, 25, and 33. GWO, WOA, and ALO algorithms were run for 17:00. The results obtained are shown in Figure 5.4, Figure 5.5, and Figure 5.6. In all results, it was seen that the pu values were between 0.95 and 1.05.

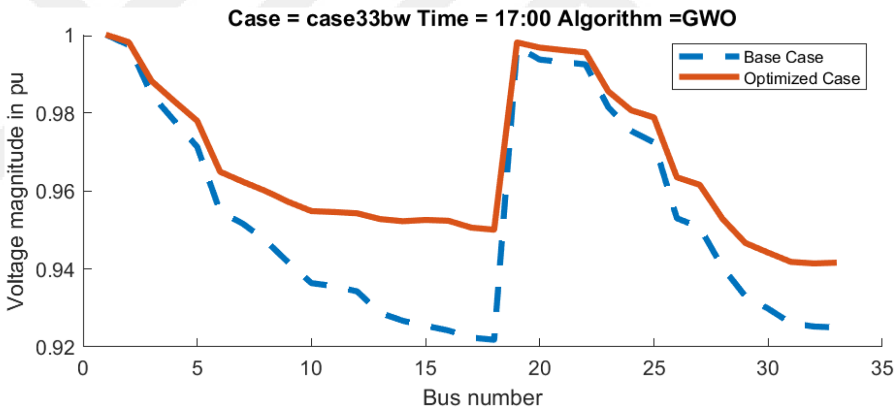


Figure 5.4 Test results for GWO.

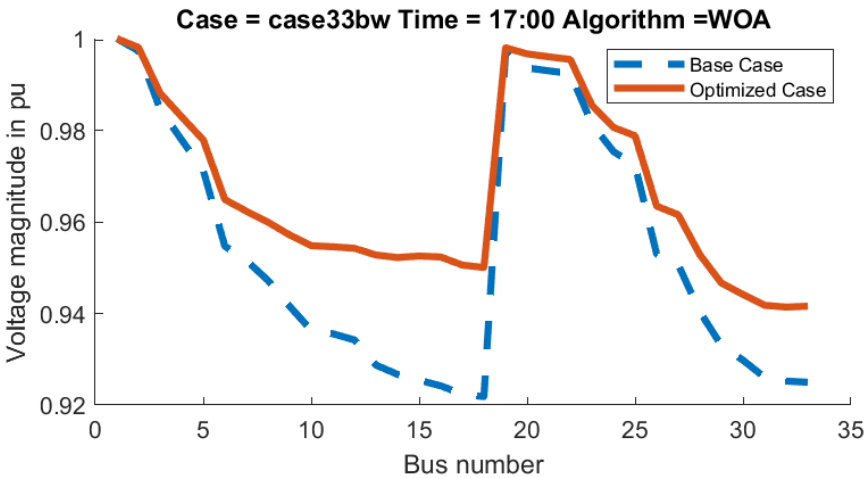


Figure 5.5 Test results for WOA.

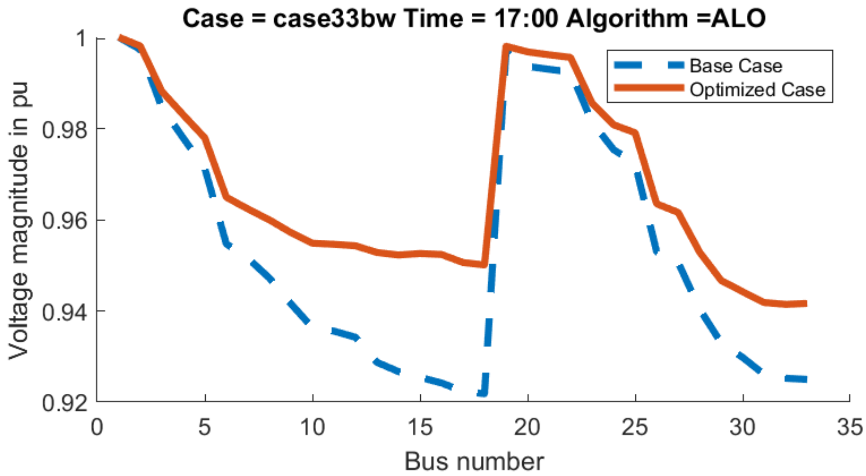


Figure 5.6 Test results for ALO.

Various tests have also been carried out in the battery and tap changer mode. By using the load values given in Table 4.3, load analysis was run in a test system with 33 bus by GWO algorithm. The iteration number is set to 100 and the search agent number to 50. The initial output values of the batteries are set as 30 kW, the step value is 5 kWh, the SoC lower limit value is 20 kW, and the SoC upper limit value is 80 kW. Tap changers has been added to the 6th and 25th buses. 8 batteries have been added to the 5th, and 13th buses. The results obtained are shown in Figure 5.7, Figure 5.8, and Figure 5.9.

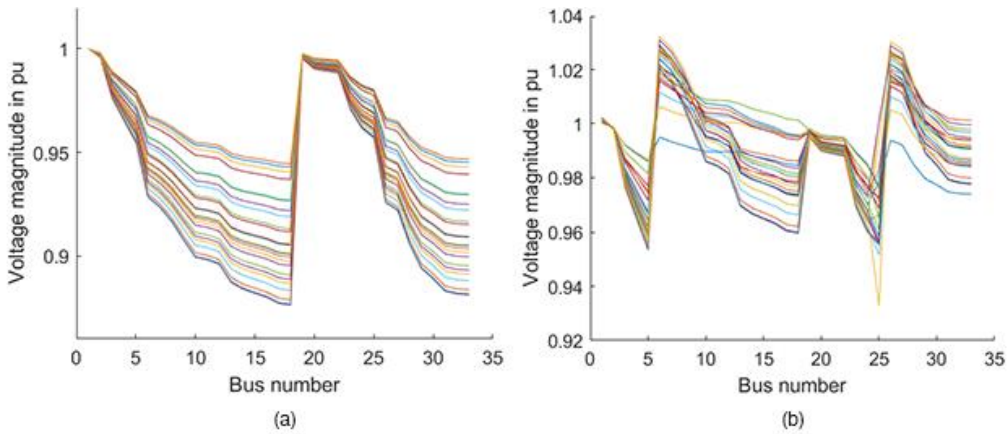
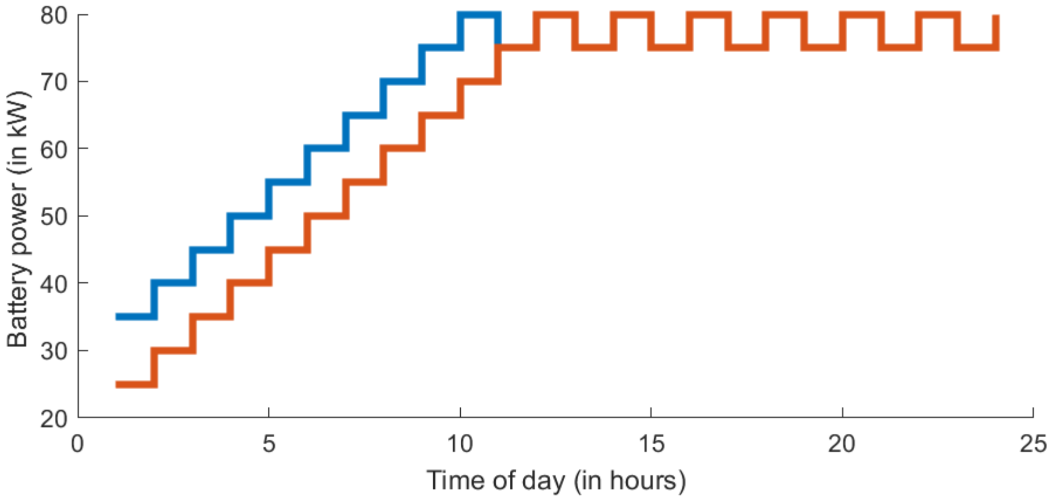


Figure 5.7 Comparison of the 24-hour voltage magnitude of 33 bus systems between base case (a) and optimized case (b).



m

Figure 5.8 Change of battery power state for 24 hours.

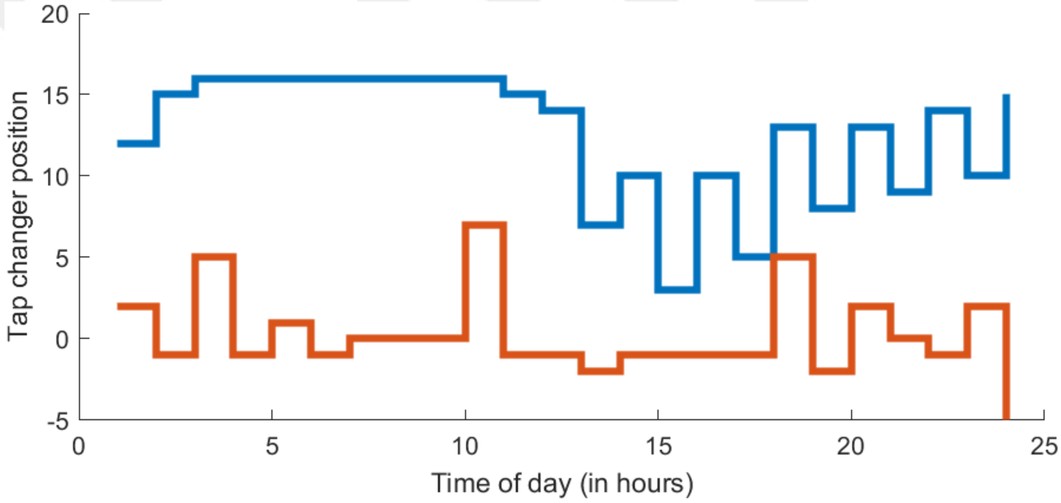


Figure 5.9 Tap changer values during 24 hours simulation.

6. CONCLUSION

In this study, the need for smart grids was mentioned due to the increasing demand for electricity. We analyzed DGs, tap changers, and BESSs as tools to minimize voltage deviations in electrical distribution systems. The closest optimum values for the tap changer and BESS system were calculated using meta-heuristic algorithms in the study.

The methods used in the literature were examined, and three meta-heuristic algorithms were selected for the graphical interface to be developed. The three algorithms to be used in the graphical interface are explained with their parts affected by nature, mathematical models, and flow charts. By using the 24-hour electricity consumption data of the consumers and the 24-hour power outputs of the PVs, the load conditions were examined. Studies have been conducted on 33 bus, 69 bus, and 141 bus systems used in the literature. We performed our analyzes using the MATLAB software language through the MATPOWER library.

To add a tap changer to the system, FBS is used instead of the Newton-Raphson method used by the MATPOWER library. Both algorithms have been tested with the same values and have been accepted as successful since they have a margin of deviation less than 0.001. It is possible to meet the energy needs of the system through batteries. Therefore, batteries are also included in the system. As a result of all these improvements, it has been observed that the voltage values in the buses are close to the desired 1 pu value.

In future studies on electricity distribution systems, an open-source tool has been developed in order to facilitate the work of researchers and students. Since the program is open source, people who want to work with different test systems or different algorithms can easily add features to the program. All source files and usage documents of the program are published on GitHub (Ozlu, 2021).

Thanks to this program, the test system, load values, algorithm to be used, DG, battery, and tap changer positions and numbers can be easily adjusted with a graphical interface. A drawing of the selected test system is displayed on the interface, and the positions chosen by the user are dynamically colored on this interface. Thus, the user can clearly see the system she/he wants to test.

Future studies aim to add electric vehicles (EVs), PVs integrated with batteries, more test systems, and algorithms to the system.

REFERENCES

Ahmadi, B., Ceylan, O. and Ozdemir, A. (2019) 'Grey Wolf optimizer for allocation and sizing of distributed renewable generation', in *2019 54th International Universities Power Engineering Conference, UPEC 2019 - Proceedings*. doi: 10.1109/UPEC.2019.8893584.

Ahmadi, B., Ceylan, O. and Ozdemir, A. (2020) 'Voltage Profile Improving and Peak Shaving Using Multi-type Distributed Generators and Battery Energy Storage Systems in Distribution Networks', in *UPEC 2020 - 2020 55th International Universities Power Engineering Conference, Proceedings*. doi: 10.1109/UPEC49904.2020.9209880.

Antoniou, A. and Lu, W. S. (2007) *Practical optimization: Algorithms and engineering applications, Practical Optimization: Algorithms and Engineering Applications*. doi: 10.1007/978-0-387-71107-2.

Atalay, T. (no date) *ELEKTRİK ENERJİSİ DAĞITIMI*. Available at: https://avys.omu.edu.tr/storage/app/public/atalayt/105449/II. SUNU ELEKTRİK ENERJİ DAĞITIMI (ŞEBEKE TİPLERİ)TA.pdf (Accessed: 30 November 2020).

Awad, A. S. A., El-Fouly, T. H. M. and Salama, M. M. A. (2015) 'Optimal ESS allocation for load management application', *IEEE Transactions on Power Systems*. doi: 10.1109/TPWRS.2014.2326044.

Ayan, K. and Kiliç, U. (2012) 'Artificial bee colony algorithm solution for optimal reactive power flow', *Applied Soft Computing Journal*, 12(5), pp. 1477–1482. doi: 10.1016/j.asoc.2012.01.006.

Bahramirad, S., Reder, W. and Khodaei, A. (2012) 'Reliability-constrained optimal sizing of energy storage system in a microgrid', *IEEE Transactions on Smart Grid*. doi: 10.1109/TSG.2012.2217991.

Baran, M. E. and Wu, F. F. (1989a) 'Network reconfiguration in distribution systems for loss reduction and load balancing', *IEEE Transactions on Power Delivery*, 4(2), pp. 1401–1407. doi: 10.1109/61.25627.

Baran, M. E. and Wu, F. F. (1989b) 'Optimal capacitor placement on radial distribution systems', *IEEE Transactions on Power Delivery*, 4(1), pp. 725–734. doi: 10.1109/61.19265.

Battery Charging and Discharging Parameters | PVEducation (no date). Available at: https://www.pveducation.org/pvcdrom/battery-characteristics/battery-charging-and-discharging-parameters (Accessed: 20 April 2021).

Battery Energy Storage for Smart Grid Applications (2013).

Beasley, D., Bull, D. R. and Martin, R. R. (1993) 'An overview of genetic algorithms : Part 1, fundamentals', *University Computing*, 2(15).

Beheshti, Z. and Shamsuddin, S. M. H. (2013) 'A review of population-based meta-heuristic algorithm', *International Journal of Advances in Soft Computing and its Applications*.

Bentouati, B., Chaib, L. and Chettih, S. (2017) 'A hybrid whale algorithm and pattern search technique for optimal power flow problem', in *Proceedings of 2016 8th International Conference on Modelling, Identification and Control, ICMIC 2016*. doi: 10.1109/ICMIC.2016.7804267.

Buayai, K. *et al.* (2014) 'Applied MATPOWER for power system optimization research', *Energy Procedia*, 56(C), pp. 505–509. doi: 10.1016/j.egypro.2014.07.185.

Bukhsh, W. A. *et al.* (2013) 'Local solutions of the optimal power flow problem', *IEEE Transactions on Power Systems*. doi: 10.1109/TPWRS.2013.2274577.

Carpentier, J. (1962) 'Contribution a l'etude du dispatching economique.', *Bulletin de la Societe Francaise des Electriciens*, 3(32), pp. 431–447.

Ceylan, O. *et al.* (2014) 'Distribution system voltage regulation by distributed energy resources', in *2014 North American Power Symposium, NAPS 2014*. doi: 10.1109/NAPS.2014.6965466.

Ceylan, O., Liu, G. and Tomsovic, K. (2018) 'Coordinated distribution network control of tap changer transformers, capacitors and PV inverters', *Electrical Engineering*. doi: 10.1007/s00202-017-0563-x.

Daylak, F. (2016) *Elektrik dağıtım sistemlerinde gerilim profilinin, güç kayıpları ve maliyet minimizasyonunun dağıtık üretim kaynakları ve kademe değiştiriciler ile iyileştirilmesi*. Istanbul Technical University.

EASE and EERA (2014) *Energy Storage Technology Development Roadmap towards 2030*. Available at: https://ease-storage.eu/publication/easeeera-energy-storage-technology-development-roadmap-towards-2030/ (Accessed: 24 April 2021).

Elektrik Dağıtım Şebekeleri - İTÜ EMK (2015). Available at: https://emhk.itu.edu.tr/elektrik-dagitim-sebekeleri/ (Accessed: 10 October 2020).

Eminoglu, U. and Hakan Hocaoglu, M. (2009) 'Three-phase transformer and voltage regulator modelling for forward/backward sweep-based distribution system power flow algorithms', *International Journal of Electrical Engineering and Education*. doi:

10.7227/IJEEE.46.1.3.

Erdoğmuş, P. and Yalçın, E. (2015) ‘Parçacık Sürü Optimizasyonu ile Kısıtsız Optimizasyon Test Problemlerinin Çözümü’, *Journal of Advanced Technology Sciences*.

Fung, R. Y. K., Tang, J. and Wang, D. (2002) ‘Extension of a hybrid genetic algorithm for nonlinear programming problems with equality and inequality constraints’, *Computers and Operations Research*. doi: 10.1016/S0305-0548(00)00068-X.

Gao, C. and Redfern, M. A. (2010) ‘A review of voltage control techniques of networks with distributed generations using on-load tap changer transformers’, in *Proceedings of the Universities Power Engineering Conference*.

Goldberg, D. E. (1994) ‘Genetic and Evolutionary Algorithms Come of Age’, *Communications of the ACM*, 37(3), pp. 113–119. doi: 10.1145/175247.175259.

Goldbogen, J. A. *et al.* (2013) ‘Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology’, *BioScience*. doi: 10.1525/bio.2013.63.2.5.

Hof, P. R. and Van Der Gucht, E. (2007) ‘Structure of the cerebral cortex of the humpback whale, Megaptera novaeangliae (Cetacea, Mysticeti, Balaenopteridae)’, *Anatomical Record*, 290(1), pp. 1–31. doi: 10.1002/ar.20407.

Höschel, K. and Lakshminarayanan, V. (2019) ‘Genetic algorithms for lens design: a review’, *Journal of Optics (India)*. doi: 10.1007/s12596-018-0497-3.

Jyh-Shing Roger Jang (1977) ‘Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence’, in *Neuro-Fuzzy and Soft Computing: A Computational Approach To Learning and Machine Intelligence*. Prentice-Hall, pp. 173–196.

Kennedy, J. and Eberhart, R. (1995) ‘Particle swarm optimization’, in *IEEE International Conference on Neural Networks - Conference Proceedings*. doi: 10.4018/ijmfmp.2015010104.

Keskintürk, T. (2006) ‘Diferansiyel gelişim algoritması’.

Khodr, H. M. *et al.* (2008) ‘Maximum savings approach for location and sizing of capacitors in distribution systems’, *Electric Power Systems Research*, 78(7), pp. 1192–1203. doi: https://doi.org/10.1016/j.epsr.2007.10.002.

Kilic, H. and Yüzgeç, U. (2018) ‘Kutulama Problemi için Geliştirilmiş Karınca Aslanı Optimizasyonu Algoritması Improved Antlion Optimization Algorithm for Bin Packing Problem’.

Kiran, D. (2018) *Radial, Parallel, Ring main and Interconnected Distribution Systems | electricaleasy.com*. Available at: https://www.electricaleasy.com/2018/02/radial-parallel-ring-main-interconneted-distribution.html (Accessed: 6 January 2021).

Kuphaldt, T. (2007) 'Lessons In Electric Circuits, Volume II--AC'.

Ladumor, D. P. *et al.* (2017) 'A grey wolf optimizer algorithm for Voltage Stability Enhancement', in *Proceedings of the 3rd IEEE International Conference on Advances in Electrical and Electronics, Information, Communication and Bio-Informatics, AEEICB 2017*. doi: 10.1109/AEEICB.2017.7972429.

Liu, X. *et al.* (2012) 'Coordinated control of distributed energy storage system with tap changer transformers for voltage rise mitigation under high photovoltaic penetration', *IEEE Transactions on Smart Grid*. doi: 10.1109/TSG.2011.2177501.

Madlener, R. and Sunak, Y. (2011) 'Impacts of urbanization on urban structures and energy demand: What can we learn for urban energy planning and urbanization management?', *Sustainable Cities and Society*. doi: 10.1016/j.scs.2010.08.006.

Mayer, D. G., Kinghorn, B. P. and Archer, A. A. (2005) 'Differential evolution - An easy and efficient evolutionary algorithm for model optimisation', *Agricultural Systems*. doi: 10.1016/j.agsy.2004.05.002.

Mirjalili, S. (2014) 'Grey Wolf Optimizer', *Advances in Engineering Software*. doi: 10.1016/j.advengsoft.2013.12.007.

Mirjalili, S. (2015) 'The Ant Lion Optimizer'. doi: 10.1016/j.advengsoft.2015.01.010.

Mirjalili, S. (no date) 'Seyedali Mirjalili'. Available at: https://seyedalimirjalili.com.

Mirjalili, S. and Lewis, A. (2016) 'The Whale Optimization Algorithm', *Advances in Engineering Software*, 95, pp. 51–67. doi: 10.1016/j.advengsoft.2016.01.008.

Office of Energy Efficiency & Renewable Energy (EERE) (2014) *Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States - Datasets - OpenEI DOE Open Data*. Available at: https://openei.org/doe-opendata/dataset/commercial-and-residential-hourly-load-profiles-for-all-tmy3-locations-in-the-united-states (Accessed: 20 October 2020).

Osman, M. S., Abo-Sinna, M. A. and Mousa, A. A. (2004) 'A solution to the optimal power flow using genetic algorithm', *Applied Mathematics and Computation*. doi: 10.1016/S0096-3003(03)00785-9.

Ozlu, I. A. (2021) *Electrical Distribution Systems Optimization GUI*. Available at: https://github.com/alperenozlu/electrical-distribution-systems-optimization-gui

(Accessed: 26 April 2021).

Pfenninger, S. and Staffell, I. (2016) ‘Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data’, *Energy*, 114, pp. 1251–1265. doi: 10.1016/j.energy.2016.08.060.

Pradhan, R. *et al.* (2019) ‘Optimal fractional order PID controller design using Ant Lion Optimizer’, *Ain Shams Engineering Journal*, 11. doi: 10.1016/j.asej.2019.10.005.

Prakash, K. *et al.* (2016) ‘Review of Power System Distribution Network Architecture’, in, pp. 124–130. doi: 10.1109/APWC-on-CSE.2016.030.

Rizy, D. T. *et al.* (2011) ‘Volt/var control using inverter-based distributed energy resources’, in *IEEE Power and Energy Society General Meeting*. doi: 10.1109/PES.2011.6039020.

Rodríguez, N. *et al.* (2018) ‘Optimization algorithms combining (meta)heuristics and mathematical programming and its application in engineering’, *Mathematical Problems in Engineering*. doi: 10.1155/2018/3967457.

Roetzel, W., Luo, X. and Chen, D. (2020) ‘Design and Operation of Heat Exchangers and their Networks’, in Roetzel, W., Luo, X., and Chen, D. B. T.-D. and O. of H. E. and their N. (eds). Academic Press, pp. 231–317. doi: https://doi.org/10.1016/B978-0-12-817894-2.00006-6.

Sarimuthu, C. R. *et al.* (2016) ‘A review on voltage control methods using on-load tap changer transformers for networks with renewable energy sources’, *Renewable and Sustainable Energy Reviews*. doi: 10.1016/j.rser.2016.05.016.

ŞENEL, F. A. *et al.* (2018) ‘Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi’, *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. doi: 10.25092/baunfbed.433321.

Soliman, O. *et al.* (2009) ‘Whale optimisation algorithm for photovoltaic model identification’, *The Journal of Engineering*, 1. doi: 10.1049/joe.2017.0662.

Staffell, I. and Pfenninger, S. (2016) ‘Using bias-corrected reanalysis to simulate current and future wind power output’, *Energy*, 114, pp. 1224–1239. doi: 10.1016/j.energy.2016.08.068.

Storn, R. and Price, K. (1995) *Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report*.

Taşkın, Ç. and Emel, G. (2002) ‘Genetik Algoritmalar ve Uygulama Alanları’, *Uludağ Üniversitesi İİBF Dergisi*, 21, pp. 129–152.

Terreson, D. *et al.* (2020) *Global Energy Outlook 2020 Energy Transition or Energy Addition, RESOURCES for the FUTURE*.

Trivedi, I. N., Jangir, P. and Parmar, S. A. (2016) 'Optimal power flow with enhancement of voltage stability and reduction of power loss using ant-lion optimizer', *Cogent Engineering*. doi: 10.1080/23311916.2016.1208942.

What is Electrical Grid? Definition & Types of an Interconnection - Circuit Globe (2018). Available at: https://circuitglobe.com/electrical-grid.html (Accessed: 6 January 2021).

Wolfram, C., Shelef, O. and Gertler, P. (2012) 'How will energy demand develop in the developing world?', in *Journal of Economic Perspectives*. doi: 10.1257/jep.26.1.119.

Yeniay, Ö. (2001) 'An overview of genetic algorithms'.

Zhang, Y. *et al.* (2016) 'Optimal allocation of battery energy storage systems in distribution networks with high wind power penetration', *IET Renewable Power Generation*. doi: 10.1049/iet-rpg.2015.0542.

Zimmerman, R. D., Murillo-Sánchez, C. E. and Thomas, R. J. (2011) 'MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education', *IEEE Transactions on Power Systems*, 26(1), pp. 12–19. doi: 10.1109/TPWRS.2010.2051168.

CURRICULUM VITAE

Personal Information

Name Surname	: İsmail Alperen Özlü
Place	: Istanbul

Education

Undergraduate Education	: Computer Engineering, Kadir Has University
Graduate Education	: Management Information Systems, Kadir Has University
Foreign Language Skills	: English

Work Experience

Name of Employer and Dates of Employment:

Alafortanfoni	09/2014 - 08/2015
Developtica	09/2015 - 2021
Microsoft (Consultant)	03/2019 - 2021